Efficient solution of population balance equations with discontinuities by finite elements

Alan W. Mahoney, Doraiswami Ramkrishna *

School of Chemical Engineering, Purdue University, West Lafayette, IN 47907-1283, USA

Received 4 May 2001; accepted 30 August 2001

Abstract

Two refinements of Galerkin’s method on finite elements were evaluated for the solution of population balance equations for precipitation systems. The traditional drawbacks of this approach have been the time required for computation of the two-dimensional integrals arising from the aggregation integrals and the difficulty in handling discontinuities that often arise in simulations of seeded reactors. The careful arrangement of invariant integrals for separable aggregation models allows for a thousandfold reduction in the computational costs. Discontinuities that may be present due to the hyperbolic nature of the system may be specifically tracked by the method of characteristics. These discontinuities will arise only from the initial distribution or nucleation and are readily identified. A combination of these techniques can be used that is intermediate in computational cost while still allowing discontinuous number densities. In a case study of calcium carbonate precipitation, it is found that the accuracy improvement gained by tracking the slope discontinuity may not be significant and that the computation speed may be sufficient for dynamic online optimization. © 2002 Published by Elsevier Science Ltd.

Keywords: Population balance; Dynamic simulation; Numerical analysis; Precipitation; Crystallization

1. Introduction

Particulate processes are among the most common in chemical processing. Techniques for the efficient accurate generation of solutions to the governing equations are required for design and control of these systems. The varying demands of solution quality and speed among applications, and the diversity specific systems require careful consideration in choosing appropriate numerical techniques. In this paper we consider new developments in the numerical application of the method of weighted residuals, specifically Galerkin’s method on finite elements, for the solution of population balance models that promise an expanded range of applications, as well as efficient solutions required by real-time control applications.

Population balance models are the natural framework for systems where a discrete population is distributed over one or more variables including particle size distributions in crystallization, precipitation, and emulsion polymerization or size and internal structure of biological cells. The resulting equations are often partial integro-differential equations with integral boundary conditions and rarely admit analytic solutions, necessitating the use of numerical techniques.

The primary application considered here is modeling particle size distribution in crystallization and precipitation. While the framework and results are applicable to a wide variety of systems, the formulation details favor these areas.

The particle size distribution is described by the number density, \( n(l,t) \), where length, \( l \), has been chosen for the characteristic particle dimension and may represent either radius or diameter. The population balance equation defines the relationship between this distribution and models governing individual particle behavior. A formulation considering the mechanisms of particle growth, aggregation, and nucleation can be written as

\[
\frac{\partial n(l,t)}{\partial t} + \frac{\partial}{\partial l} \left[ G(l,y)n(l,t) \right] + Dn(l,t) = h(l,n,y),
\]

where \( G(l,y) = \dot{L}(l,y) \) is the specific growth rate of a particle of size \( l \). The vector \( y(t) \) is the collection of continuous phase variables, that may include concentrations or supersaturation, temperature, and shear rate or power dissipation. These may be either inputs to the system or governed by algebraic or differential equations. The dilution rate, \( D \), is defined as the rate of volumetric addition to the system divided...
by the current system volume and may vary with time. This allows the same equation to be used at all times, whether the system is undergoing batch, semi-batch, or continuous operation.

Candidate models for growth rate include first principles models such as diffusion limited growth, spiral dislocation growth, and surface poly-nucleation (House, 1981; Nielsen, 1984; Tai, Cheng, & Huang, 1992); empirical models (Abegg, Stevens, & Larson, 1967; Rojkowski, 1978); or can be obtained through inverse problem modeling (Mahoney, Doyle, & Ramkrishna, 2001).

Nucleation can be considered to occur in a distributed manner across a range of particle sizes or, as considered here, at a single particle size which may change with system conditions. For simplicity, we will consider a constant, zero, here, at a single particle size which may change with system manner across a range of particle sizes or, as considered later.

Di;UVculty in choosing appropriate integration stability reduce the appeal of;HHnite di;ˆQerence techniques f or these problems. The complexity of these systems precludes the existence of analytic solutions in most cases, resulting in the development or adaptation of a number of numerical techniques. Three most common approaches are finite difference methods, discretization techniques, and the method of weighted residuals.

Finite differences are a popular general approach to the solution of partial differential equations. An analysis of performance with population balance equations, including comparison of a number of finite difference schema has been presented by Wójcik and Jones (1998). Nucleation, growth, aggregation and breakage are included in a system distributed both in size and degree of aggregation. Careful selection of numerical scheme was required for accurate solution. Muhr et al. apply;HHnite di;ˆQerences to a system undergoing growth and nucleation only (Muhr, David, Villemaux, & Jezequel, 1996). One drawback of the finite difference method is that number and mass conservation are only guaranteed in the limit of infinite resolution (Patankar, 1980). As the moments of the distribution often relate directly to measurable quantities, their accurate prediction is often required. A second dif;uciency arises due to the hyperbolic nature of the differential equation. As batch systems are often seeded to provide for secondary nucleation, a discontinuity can arise along the separatrix, the curve that divides states deriving from initial conditions from those arising from boundary conditions. In a seeded system, the initial number density of particles of nuclei size is unlikely to be the same as that initially created by nucleation. Even in the event that the values match, the first derivative of the distribution is unlikely to match. This can create a sharp discontinuity that quickly broadens by numerical diffusion in simulation. These drawbacks, along with the complexity of choosing appropriate integration stability reduce the appeal of finite difference techniques for these problems.
A second class of methods used to solve population balance equations has been called discretized population balances (Hounslow, Ryall, & Marshall, 1988; Kumar & Ramkrishna, 1997), the method of classes (Marchal, David, Klein, & Villermaux, 1988) or sectional methods (Gelbard, Tambour, & Seinfeld, 1980). These techniques divide the domain into finite regions on which the number density is integrated to provide balances between the number of particles in each “bin”. This formulation is equivalent to the control volume formulations or integration method applied in the areas of fluid dynamics (Versteeg & Malalasekera, 1995; Varga, 1965) and reaction kinetics. One advantage of this method is that desired moments of the distribution can be explicitly conserved in aggregation processes by the appropriate distribution of newly formed particles in existing bins (Kumar & Ramkrishna, 1996). Number is also explicitly conserved under growth. Sharp discontinuities are captured by allowing the discretization to move or deform with particle growth (Kumar & Ramkrishna, 1997). This also preserves higher moments under growth conditions. While chosen moments of the distribution are captured, convergence of the shape of the distribution is not guaranteed. When using moving bins, the number of states and their physical interpretation change through time. This change in structure which can cause difficulties for control applications.

The method of weighted residuals is a general technique for the solution of partial differential equations, of which finite element methods are one subset. Rather than holding at every point in the domain, only selected weighted residuals of the equation are considered to hold. These residuals form a set of algebraic (or ordinary differential) equations in coefficients that form a basis function expansion of the solution. Formulations are differentiated on the use of global or local basis functions. Global functions are often used when approximate or limiting forms of the solution are available. Global expansion has been used for steady state solutions with aggregation and growth by Bhatia et al. using limiting solutions for basis functions and moment equations for weighting (Bhatia & Chakraborty, 1992). Finite element techniques use local basis functions and allow more general solutions. Possible choices of the weighting function include Dirac delta functions, resulting in collocation methods, or the basis functions themselves (possibly weighted), resulting in Galerkin’s techniques. Finite element techniques have been used in dynamic collocation (Gelbard & Seinfeld, 1978; Eyre, Wright, & Reuter, 1988) and steady-state Galerkin’s formulations (Nicmanis & Hounslow, 1996, 1998). Advantages are convergence properties on the entire distribution and the availability of adaptive mesh techniques. Major drawbacks include the inability to capture discontinuities, as may arise along the separatrix, and the computational overhead arising from double integral evaluations in Galerkin’s formulations.

This paper addresses these two drawbacks of the application of Galerkin’s method on finite elements for the solution of population balance equations. A major computational cost is the evaluation of double integrals arising from aggregation. This can be reduced by retaining as much of the previous computations as possible. For separable aggregation models, proper subdivision of the required integrals allows most or all of the computation products to be retained from one time step to the next, greatly reducing the cost of further evaluations. Other questions are under what conditions this improved efficiency is possible. The presence of discontinuities in the solution can be problematic because they cannot be captured using the general basis functions used for the solution expansion. Using characteristics in a locally similar way to the method of moving classes allows these phenomena to be found and the number distribution captured.

2. Method of weighted residuals

All numerical solutions of partial differential equations are generated by an approximation. In finite difference methods, the derivatives are approximated by truncated series expansion. In the method of weighted residuals, the equations are considered to hold in an approximate sense: rather than holding everywhere, only selected weighted residuals of the equations hold. Applying weighting functions \( \psi_j(l) \) to the population balance equation (1) and forming residuals yields

\[
R_i = \int_0^\infty dl \, \psi_i(l) \frac{\partial n(l,t)}{\partial t} + \int_0^\infty dl \, \psi_i(l) \frac{\partial (G(l,t)n(l,t))}{\partial l} + D \int_0^\infty dl \, \psi_i(l)n(l,t) - \int_0^\infty dl \, \psi_i(l)h(l,n,t).
\]  

(6)

The approximate solution is defined as

\[
n(l,t) = \sum_{j=1}^N n_j(t) \phi_j(l),
\]

where \( n_j \) are the coefficients of expansion, and \( \phi_j \) are the still unspecified basis functions. If this expansion is applied to the first term of the formula for the residual, and the order of summation and integration reversed,

\[
\int_0^\infty dl \, \psi_i(l) \frac{\partial n(l,t)}{\partial t} = \sum_{j=1}^N \frac{dn_j}{dt} \int_0^\infty dl \, \psi_i(l) \phi_j(l) = \sum_{j=1}^N \frac{dn_j}{dt} M_{ij},
\]  

(7)

where \( M_{ij} \) are elements of the mass matrix \( \mathbf{M} \), with definition given in Table 1. The remaining terms are similarly treated,
giving rise to the convection matrix $P$ and the aggregation tensor $H$. These terms are also found in Table 1. If the growth law contains step discontinuities, this can be accommodated by applying integration by parts and results in a slightly modified formulation.

The substitutions allow the residuals to be written in a matrix equation:

$$
r = Mn + P(y)n + D Mn - n^T H(y)n. 
$$

If the time dependence can be factored from aggregation or growth models, the computational time can be dramatically reduced. Separable models will be considered those which can be expressed as

$$
a(l, l', y) = \sum_m a_{m, l}(y) a_{l, m}(l, l')
$$

or similarly for $G(l, y)$. In precipitation, separable laws are frequently used for growth, nucleation, and aggregation. If aggregation and growth are separable and have only one term, the residual equation becomes

$$
r = Mn + G_i(y) P n + D Mn - a_i(y) n^T H n,
$$

where $M$, $P$ and $H$ are all independent of time. Additional terms in the laws generate additional terms in the equation.

The boundary condition arises from nucleation. Secondary nucleation usually dominates when there are already particles in the system and is dependent on the particles currently in the system. It is commonly written as an functional of the number density:

$$
n(0, t)G(0, y(t)) = \int_0^\infty dl' \beta_3(l', y)n(l', t) = B^T(y)n. 
$$

The formulation to this point is general for weighted residual methods, holding for any sets of weighting and basis functions, global or local, and includes collocation methods. The remainder of the paper will focus on the finite element formulation, but several of the issues dealt with are common to all weighted residual formulations.

## 3. Finite element formulation

The intended application of the solution plays a large role in the choice of solution technique. One of the goals here is to capture discontinuities moving through the particle size distribution. As this is a local phenomenon, local basis functions are chosen. Global basis functions require a full set of coefficients of expansion on each side of the discontinuity whereas local formulation requires only those at the interface be duplicated. The banded Jacobian often seen in finite element problems is not produced in this formulation, as aggregation produces a coupling of all expansion coefficients, resulting in a non-sparse Jacobian.

Linear basis functions were chosen as they are the simplest that will capture convection/growth. Example of local linear basis functions used for expanding the solution are shown in Fig. 2. Unfortunately, the collocation formulation is unstable on these basis functions, and so its performance is not included in the results. In Galerkin’s method, weighting functions are identical to the basis functions (with a possible weighting factor). Here, $\psi(l) = \phi(l)$.

### 3.1. Evaluation of integrals

The integrals shown in Table 1 are computed using Gaussian quadrature. As the accuracy of numerical quadrature is highly dependent on the smoothness of the integrand, care needs to be taken to avoid slope discontinuities (corners) in the integrand. Most of the integrals are over a single dimension, requiring splitting only at the node points, but the two-dimensional integrals in $H$ require more careful treatment. Fig. 1 shows the domain of integration along with the lattice defined by the basis functions. The region integrated for any individual element is small, but complex in shape. Smoothness constraints require subdividing the area both along nodes of the basis functions (solid lines)
turbation is introduced to eliminate the singular behavior. The domain is not sampled, therefore a change of variables causes problems with numerical accuracy even if the end of the integral to avoid singularity in the number density. This singularity is a remnant of choosing length as the characteristic dimension for notational simplicity. The singularity in the integrand is

\[ h = 1 \]

without corners, as shown by the dashed lines. However, as shown in Fig. 1. Partitioning of plane of integration for calculation of aggregation of the singularity in the integrand. The values were of unity order.

As \( u \) approaches 0, \( 3l^2u - 3lu^2 + u^3 \) approaches \( 3l^2u \) showing the singularity to be order \( u^{-3/2} \), so the integral converges for bounded \( a \) and \( n \). The removal of such singularities can be accomplished through the transformation \( u = w^{1/(1-\gamma)} \), where \( \gamma \) is the order of singularity. Here, \( \gamma = \frac{2}{3} \) and \( u = w^3 \). Note that this cubic dependence is unrelated to the transformation from volume to length coordinates and does not return the integral to volume coordinates. Applying the transformation and simplifying,

\[ h^+_i = \frac{3}{2l} \int_0^\infty dl \int_0^l n(l') \]

\[ \times (\sqrt{l^3} - l^3) a(l', \sqrt{l^3} - l^3) \frac{l^2}{(l^3 - l^3)^{2/3}}, \]

where the time dependence of \( n \) and \( a \) have been suppressed for notational simplicity. The singularity in the integrand is a remnant of choosing length as the characteristic dimension to avoid singularity in the number density. This singularity causes problems with numerical accuracy even if the end of the domain is not sampled, therefore a change of variables is introduced to eliminate the singular behavior.

The first concern is the convergence of the integral, which depends on the order of the singularity. Considering a perturbation \( u = l - l' \) to replace \( l' \) gives

\[ h^+_i = \frac{3}{2} \int_0^\infty dl \int_0^l du n(l-u) \]

\[ \times a(l-u, \sqrt{3l^2u - 3lu^2 + u^3}) \]

\[ \times l^2 \]

\[ \times \frac{l^2}{(3l^2 - 3lu^2 + u^3)^{2/3}}. \]

Fig. 1. Partitioning of plane of integration for calculation of aggregation integrals.

Table 2

<table>
<thead>
<tr>
<th>Quadrature domain and formula</th>
<th>[ \sqrt{\sum \sigma^2} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original, 3 point Gaussian</td>
<td>0.4136</td>
</tr>
<tr>
<td>Original, 7 point Gaussian</td>
<td>0.2435</td>
</tr>
<tr>
<td>Transformed, midpoint</td>
<td>0.1006</td>
</tr>
<tr>
<td>Transformed, 3 point Gaussian</td>
<td>1.4119e-4</td>
</tr>
<tr>
<td>Transformed, 7 point Gaussian</td>
<td>5.8521e-5</td>
</tr>
</tbody>
</table>

*The error is dramatically reduced by variable transformation to eliminate the singularity in the integrand. The values were of unity order.

Errors in the computed aggregation tensor

As \( u \) approaches 0, \( 3l^2u - 3lu^2 + u^3 \) approaches \( 3l^2u \) showing the singularity to be order \( u^{-3/2} \), so the integral converges for bounded \( a \) and \( n \). The removal of such singularities can be accomplished through the transformation \( u = w^{1/(1-\gamma)} \), where \( \gamma \) is the order of singularity. Here, \( \gamma = \frac{2}{3} \) and \( u = w^3 \). Note that this cubic dependence is unrelated to the transformation from volume to length coordinates and does not return the integral to volume coordinates. Applying the transformation and simplifying,

\[ h^+_i = \frac{3}{2l} \int_0^\infty dl \int_0^l n(l') \]

\[ \times (\sqrt{l^3} - l^3) a(l', \sqrt{l^3} - l^3) \frac{l^2}{(l^3 - l^3)^{2/3}}, \]

Not only is the integrand now bounded over the entire domain of integration, the weighting term is finite at both ends, effectively reducing the polynomial degree of the integrand and improving the quadrature accuracy.

Certain forms of the aggregation kernel may change the order of singularity and require individual treatment. The constant, sum, turbulent inertia, and Thompson kernels are all similarly behaved, while the product kernel gives \( \gamma = \frac{1}{3} \). The Brownian kernel is more difficult as the integrand is singular in \( 1/u \) for finite \( n(0,t) \), and the behavior of \( n(l,t) \) near \( l = 0 \) must also be considered. The use of a small, finite nucleation size does not completely eliminate the concerns, as the singular behavior affects quadrature accuracy over a wide range of particle sizes.

To quantify the integration error and choose a quadrature scheme, \( \alpha^H \) \( H_n \) for constant \( a \) and \( n \) were calculated to high precision using a symbolic software package on a domain of \( l = [0, 1, \ldots, 10] \) and compared with the numerical results. The errors for various integration techniques in Table 2 show a strong penalty for performing quadrature in the original coordinates. From the results, a three point Gaussian quadrature on the transformed domain was chosen for the remainder of the paper.
4. Discontinuity tracking

The framework presented so far has difficulty capturing sharp changes or fronts in the distribution. If the sharp change occurs over a finite width, locally or globally fine grids can be used to capture the solution. However, in precipitation and crystallization systems true step changes can arise. At start-up of a seeded system the number density at zero size has two values, one corresponding to the seeded distribution and another arising from the initial nucleation rate. This discontinuity proceeds through the system along the separatrix, the line separating states dependent on the initial condition from those dependent on the boundary condition. Other discontinuities may be present in the initial distribution or arise from sharp changes in the nucleation rate.

One way to deal with these effects is to choose a discretization that moves with the growth paths of the system. As discontinuities enter the system, they can be captured with double nodes that are carried along with the solution. This requires continual re-evaluation of aggregation terms even if aggregation is separable, at a high computational cost. This idea can be used in a hybrid approach where only the nodes corresponding to discontinuities are moved and the remainder of the domain is solved using the previously computed integrals, retaining the advantage of time invariant calculations.

Of initial concern is the number and source of discontinuities in the solution. Those present in the initial seed distribution, entering through time discontinuities in the nucleation rate (boundary condition) and the separatrix, are easily identified before or during simulation. Any arising in the midst of the distribution would be substantially more difficult to identify. Two possible sources of new discontinuities are shock wave formation and the reproduction of existing discontinuities through aggregation. It is shown that neither of these occurs in the system under consideration.

First order partial differential equations display shock wave formation when the characteristics arising from different sources in the initial or boundary conditions cross. This crossing can only occur if the growth rate \( G \) has multiple values at the point of crossing. Thus the reasonable condition that the growth rate is independent of the number density \( n(l,t) \) prevents shock wave formation. This leaves only aggregation as a method for creating discontinuities in the midst of the particle size distribution. It is shown in Appendix A that \( h(l,n,v) \) is discontinuous only where discontinuities already exist and creates no new ones. Often in batch systems the separatrix will be the sole discontinuity, as seed distributions and the nucleation rate are likely to be smooth. In continuous systems after startup, there may be none.

The location of the added node point will be referred to as an interface since it separates the domain into two parts that do not interact through growth. This may track a step, slope or higher order discontinuity. Additional state variables required for each interface are the current particle size and the limiting number densities from the positive and negative directions. The particle size corresponding to the interface is described by particle growth,

\[
\frac{dl_I}{dt} = G(l_I, t). \tag{14}
\]

This relationship is applied by directly making use of the hyperbolic nature of the system to find the characteristics. The left and right limits of the number density \( n(l, t) \), designated by \( n_+(l_I, t) \) and \( n_-(l_I, t) \), are obtained by solving the population balance equation along characteristics:

\[
\frac{dn_{\pm}(l_I)}{dt} + n_{\pm}(l_I) \frac{\partial G(l_I, y)}{\partial l} + Dn_{\pm}(l_I, t) = h(l_I, n, y) = n^T h_I(y) n. \tag{15}
\]

The aggregation coefficient matrix \( h_I \) given in Table 1 is recalculated as \( l_I \) changes. Formulations that use derivative information at the node points (Hermite cubics, for example) will require additional states to be stored and calculated.

As the interface moves, the node point nearest to the interface is relocated to the location of the interface. The domains of each residual are restricted to remain on the same side of the interface at both the last and the current times, ensuring the time derivatives in the residuals do not cross the interface. Original and modified local linear basis functions are sketched in Fig. 2.
with new calculations involving the basis functions corresponding to \( n_3(l, t) \). Values adjacent to these must also be recalculated as the weighting and basis functions will have changed. In the matrices, the values intersecting either node must be recalculated. The recalculation for the vectors involved 3 integrals, and the matrices up to 6 integrals is involved. The recalculation for the vectors involved 9 calculations, though sparsity reduces this.

5. Accuracy and speed comparison

Application requirements will determine the desired characteristics for simulation techniques, and these must be used in evaluation. The potential choices are limited as few analytical solutions are available for population dynamics. These include solution by characteristics for systems undergoing time-varying nucleation and growth; similarity solutions for purely aggregating systems for certain initial conditions and kernels (which may be time varying if separable) (Scott, 1968); and aggregation dominated growth without nucleation (Ramabhadran, Peterson, & Seinfeld, 1976). While these systems do not span the behavior of interest, they can provide some measure of the comparable accuracy of various solution techniques.

The system of weighted differential equations, coupled with an algebraic integral equation for nucleation, were approximated in time using a second order Adams–Bashforth–Moulton predictor–corrector scheme with a fully implicit corrector step. Little additional computation is required to provide a full analytic Jacobian as the integrals are all approximated in time using a second order Adams–Bashforth–Moulton predictor–corrector scheme. The computations were performed with and without interface tracking, even if no discontinuity was present, for method validation. In most cases the simulations converged at the expected rate (\( N_p^{-2} \)) to the analytic distribution, with exceptions for some growth systems when the separatrix was not tracked (see Fig. 3). This is expected, as this is the motivation for performing interface tracking. We will now examine the computational and accuracy issues arising from two typical systems.

The first system investigated is constant aggregation. The conditions used for simulation are in Table 3. The domain \( l = [0, 10] \) was divided into 100 linearly spaced panels unless otherwise noted. Note that linear spacing in length is equivalent to cubic spacing in volume, the more common characteristic dimension in the literature. Solutions are compared at \( t = 50 \) when the number of particles has been reduced to \( \sim 3.5\% \) of the original.

![Fig. 3. Solutions showing dramatically improved solution with interface tracking for a seeded system.](image)

Table 3

<table>
<thead>
<tr>
<th>Conditions, parameters, and analytic solution for pure aggregation system</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a(l, l') = a(v, v') = N_0 = v_0 = 1 )</td>
</tr>
<tr>
<td>( n_v(v, 0) = N_0 \exp(-v/v_0) )</td>
</tr>
<tr>
<td>( \hat{n}(v, t) = ah(t) \exp\left(-\frac{v}{v_0}\right) )</td>
</tr>
<tr>
<td>( g(t) = \frac{(aN_0t + 2)v_0}{2} )</td>
</tr>
<tr>
<td>( h(t) = \frac{4}{aN_0t + 2v_0} )</td>
</tr>
<tr>
<td>( \hat{n}(l, t) = 4\pi\hat{n}(v, t)^2 )</td>
</tr>
<tr>
<td>( v = \frac{4}{3}\pi l^3 )</td>
</tr>
</tbody>
</table>

Volume and number based error measures were computed on the final distribution:

\[
e_n = \sqrt{\int_0^\infty dl (n(l, t) - \hat{n}(l, t))^2} / \int_0^\infty dl \hat{n}(l, t)
\]

\[
e_v = \sqrt{\int_0^\infty dl l^2 (n(l, t) - \hat{n}(l, t))^2} / \int_0^\infty dl l^2 \hat{n}(l, t)
\]

Since the grid position is fixed, the aggregation tensor \( \mathbf{H} \) was evaluated only once, which took nearly as much computational time as the simulation (9.3 versus 14.3 s—the conditions for all times are noted in Table 5). The number and size of steps taken by the adaptive routine and the total number of Newton iterations were largely unchanged as the number of panels was varied from 50 to 200. The total execution time varied as \( N_p^{1/2} \). Evaluation of the tensor of the tensor product \( \mathbf{n}^\dagger \mathbf{H} \mathbf{n} \) accounts for 75% of the computational time, and is proportional to the sparsity of \( \mathbf{H} \), or \( N_p^{1/2} \). For large systems (\( N_p > 750 \) here), the full Jacobian matrix inversion is expected to dominate with complexity proportional to \( N_p^3 \). The solutions converged as (error decreased as) \( N_p^3 \), the expected rate for linear interpolates.
To evaluate the demands on quadrature accuracy, the integrals were computed using 3rd to 13th order (2–7 points) Gaussian formulae in each dimension. These changes had no noticeable effect on the accuracy of the solution. This is likely to be because of the already small regions of integration used for the application of quadrature formula (see Section 3.1), but should be re-evaluated for particular models of interest. As the time required for computation of $H$ scales with the square of the number of quadrature points, execution time is greatly affected. The use of a midpoint rule for quadrature produced solutions oscillating in time that did not converge. The success of low-order quadrature in capturing the solutions means that collocation and Galerkin’s techniques have similar computational costs per time step, so performance must be used for evaluation.

Growth simulation was evaluated using the conditions in Table 4. The parameters were chosen so that both the number density $n(l, t)$ and its first derivative with respect to $l$ are continuous along the separatrix. Solutions at $t = 8$ with and without interface tracking are shown in Fig. 4. Without a node tracking the interface, the solution converges with increasing $N_p$, but not to the analytic solution. While the discrepancy is local to the separatrix, it grows with continued simulation, and suggests that interface tracking must be considered for high-accuracy simulation. With the addition of an interface along the separatrix, the solution converges to the analytic solution with error $e_a$ scaling as $N_p^{-2}$.

Computational speed is often of concern when choosing solution techniques. Especially in concert with real-time optimization it can be of key importance. Table 5 shows the execution times for calculations on representative systems. By far the most expensive part of the calculation is generating the aggregation tensor, $H$. If the aggregation dependence is separable and the integrals are not re-calculated at each time there is a dramatic reduction in simulation time. Even incorporating only the changes for a moving interface is an improvement over a full calculation, but is still the dominant cost of the simulation. Unless true discontinuities are present in the system, the value of the extra accuracy must be weighed against that of faster simulation, and intermediate measures may be evaluated, such as updating $H$ only once per time step rather than each iteration, or calculating the updates with lower order quadrature.

The scaling of time with the number of panels for generation of the aggregation tensor is lower than the expected cubic dependence due to the sparsity of the tensor. This sparsity increases with the number of panels. Correcting the tensor for a moving interface requires only the values lying along intersecting planes to be recalculated, reducing the order of complexity. The speed of execution is also affected by the quadrature formula. The time to calculate all integrals but $H$ scale as the number of quadrature points, while the time to compute $H$ scales as the square.

Adaptive mesh generation can be used to improve the accuracy of finite element calculations by concentrating mesh points in regions of sharp changes (often measured by the second derivative). The use of adaptive mesh generation requires re-calculation of the coefficients with each new mesh. If this processes is occasional, or affects only a portion of the domain (as in interface tracking), significant
Table 5
Average execution times for Galerkin’s FEM simulation

<table>
<thead>
<tr>
<th>Step</th>
<th>Galerkin’s computing time (s)</th>
<th>Scaling expected</th>
<th>Scaling found</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate M, P, G, etc.</td>
<td>0.0023</td>
<td>$N_P$</td>
<td>$N_P$</td>
</tr>
<tr>
<td>Assemble J, R, no interface</td>
<td>0.0022</td>
<td>$N_P$</td>
<td>$N_P$</td>
</tr>
<tr>
<td>Generate $H$</td>
<td>7.3</td>
<td>$N_P^3$</td>
<td>$N_P^3$</td>
</tr>
<tr>
<td>Correct $H$ for interface</td>
<td>0.73</td>
<td>$N_P^2$</td>
<td>$N_P^2$</td>
</tr>
<tr>
<td>Other interface calculations, with aggregation</td>
<td>0.33</td>
<td>$N_P^2$</td>
<td>$N_P^2$</td>
</tr>
<tr>
<td>Calculate $J^{-1}R$ with aggregation</td>
<td>0.0016</td>
<td>$N_P^3$</td>
<td>$N_P^2$</td>
</tr>
</tbody>
</table>

*Code compiled and run on a Sun Sparm Ultra-5 (1 400 MHz processor, 128MB memory) using C++ and FORTRAN Sun Workshop Compilers 5.0 with the -fast compiler option. Times listed are for 100 nominal panels, integrals computed by 3 point Gaussian quadrature. Scaling was observed to hold over 25–200 panels.

savings may still be obtained re-using integral coefficients. In repeated batch simulations, it may be advantageous to use a pre-programmed series of grid discretizations, each with a pre-computed aggregation tensor.

These computations show a gradual tradeoff between accuracy and computational time for Galerkin’s method on linear finite element basis functions. The use of higher order basis functions will further reduce the error for given computational complexity, and seems required for convergence of collocation methods, but also adds structural complexity that makes it more difficult to apply interface tracking and other refinements.

6. Case study: calcium carbonate precipitation

Calcium carbonate grades produced by precipitation are used in applications from paper pulp to pharmaceuticals by a variety of production techniques and applications. Widespread commercial importance has also made it one of the most studied precipitation systems. Here we consider the simulation of a continuous precipitation reactor with the order of hours rather than seconds, allowing the equilibrium assumption. Activities are corrected using the Davies correction to the extended Debye–Huckel equation (Davies, 1962) (this correlation is incorrectly given in Collier and Hounslow, 1999)

\[
\log y_i = -0.5085z_i^2 \left( \frac{\sqrt{I}}{1 + \sqrt{I}} - 0.3I \right),
\]

where $z$ is the charge on a species and $I$ is the ionic strength, given by

\[
I = \frac{1}{2} \sum_i c_iz_i^2.
\]

These were solved by iteration on the ionic strength coupled with solution of the calcium, carbonate, and charge balances for three key concentrations. The relative supersaturation is a measure of the disequilibrium of the continuous phase with the solid phase and gives the driving force for nucleation, growth and aggregation. It is given by

\[
\sigma = \sqrt{\frac{a_{Ca^{2+}}a_{CO_3^{2-}}}{K_{sp}}} - 1.
\]

The growth and aggregation laws used are taken from studies done by Colliner and Hounslow, which they reconciled with previous studies (Collier and Hounslow, 1999). The model
Table 6
Nominal models and parameters for simulation of calcium carbonate precipitation

\[ G(l, \sigma) = 1.28 \times 10^{-10} \sigma^2 \text{m}^2/\text{s} \quad (\text{Collier & Hounslow, 1999}) \]
\[ a(l', l, \sigma) = 8.71 \times 10^{-6} \rho_c \int_0^\infty \rho_c \, dl \, n(l, t) \, \text{m}^3/\text{s} \quad (\text{Collier & Hounslow, 1999}) \]
\[ b(n, \sigma) = 5.07 \times 10^7 \sigma^{1.82} \int_0^\infty \rho_c \, dl \, n(l, t) \, \text{particles/s} \quad (\text{Tai & Chen, 1995}) \]
\[ D = \frac{1}{5h} \]
\[ c_{Ca,in} = c_{CO3,in} = 1.03 \text{ mol/m}^3 \]
\[ K_{sp} = 4.7 \times 10^{-3} \text{ mol}^2/\text{m}^6 \]

The form of these laws, along with the parameters used are shown in Table 6. While the ionic environment is not the same as in either case, they are considered representative of the system behavior.

Simulations of two phenomena are considered. The first is secondary nucleation is taken from Tai and Chen (1995).

The second change investigated was inlet concentration. This is one of the few variables that may be used to affect the system (agitation rate and dilution rate are others), and would be evaluated repeatedly in real-time optimization based control. In this case, a doubling of the inlet reactant concentrations was performed, and again the results were compared each 10 min of simulated times. In this case, the error between the two simulations is much lower, due to additional attenuation of the disturbance by the system and a smaller magnitude slope discontinuity. The maximum deviation occurs at \( t = 450 \pm 5 \text{ min} \), and is shown in Fig. 6. This deviation is expected to be insignificant in most applications. The time for the solution varied significantly, taking \( \sim 9 \text{ s} \) for the case without interface and 830 with interface (note these do not include the 8 s for generating \( H \), which needs to be done only once regardless of the number of simulations). As the evaluation of multiple step changes would lead to disturbances moving through the system, this is a conservative estimate of the time required in both cases, but tracking additional interfaces will impact the solution time more dramatically.
7. Conclusions

The use of Galerkin’s method on finite elements is a powerful one for the solution of population balance equations. The computational complexity of generating the double integrals required is no greater than that for a similar order collocation scheme. In cases where the aggregation rate is separable, the cost is effectively eliminated by pre-computation of the integrals.

For systems where discontinuities are present, these can be tracked by appending of additional node points that move along the growth characteristics. All of the discontinuity sources can be easily identified before simulation starts or as they enter the system. This tracking does require recalculation of a number of the integrals involved, but only a comparatively small subset that grows at a significantly slower rate than matrix inversion. While in seeded systems, an interface is nearly always required along the separatrix during startup, in many other cases the accuracy loss from lower order discontinuities may be tolerable, especially in continuous systems where the error is less likely to accumulate. The simulations without interfaces nodes took 500–1000 times less computation than those just only recalculating the integrals along the interface. These in turn are 10 times faster than computing the entire aggregation tensor at every step. The performance for the interface free case is likely to be adequate for dynamic optimization requirements, particularly for long response time systems. Process and controller design will also benefit, especially for batch systems with significant aggregation that are currently difficult to solve.

Appendix A. Discontinuities arising from aggregation

Discontinuities can arise from aggregation only if \( h(l,n,y) \) is discontinuous with respect to \( l \). Examining this in volume coordinates involves simpler terms and is mathematically equivalent to length coordinates. The function \( h \) is continuous at \( v \) if and only if

\[
\lim_{\delta \to 0} [h(v + \delta, n, y) - h(v, n, y)] = 0. \tag{A.1}
\]

For notational simplicity, dependence on time and continuous variables will be considered implicit: \( n(v) \) and \( a(v, v') \) will be written for \( n(v, t) \) and \( a(v, v', y(t)) \). Splitting up the aggregation terms into source and sink terms, \( h = h^+ - h^- \), and examining the sink term first,

\[
h^-(v + \delta) - h^-(v) = \int_0^\infty dv' n(v + \delta)n(v')a(v + \delta, v')
- \int_0^\infty dv' n(v)n(v')a(v, v') \tag{A.2}
- \int_0^\infty dv' n(v')\{n(v + h)a(v + h, v')
- n(v)a(v, v')\}. \tag{A.3}
\]

This is discontinuous only where \( n(v) \) or \( a(v + h, v') \) are discontinuous. If \( a \) is continuous, discontinuity in \( h^-(v) \) occurs only where there is already a discontinuity in \( n(v) \), and contributes nothing more.
The source term appears as
\[
2(h^+(v + \delta) - h^+(v)) = \int_0^{v+\delta} dv' n(v + \delta - v')n(v')\alpha(v + \delta - v', v') \\
-\int_{v-\epsilon}^{v-\epsilon+\delta} dv' n(v - v')n(v')\alpha(v - v', v') + \int_{v-\epsilon}^{v-\epsilon} dv' n(v - v')n(v')\alpha(v - v', v') + \int_{v-\epsilon}^{v-\epsilon} dv' n(v - v')n(v')\alpha(v - v', v').
\]
(A.4)

Since \(v\) is singular at \(v \to 0\), there are singularities at both ends of each integral. These are isolated by separating domains of finite width \(\epsilon\):
\[
2(h^+(v + \delta) - h^+(v)) = \int_0^{v+\delta-\epsilon} dv' n(v + \delta - v')n(v')\alpha(v + \delta - v', v') \\
+ \int_{v-\epsilon}^{v+\delta-\epsilon} dv' n(v + \delta - v')n(v')\alpha(v + \delta - v', v') + \int_{v-\epsilon}^{v+\delta-\epsilon} dv' n(v + \delta - v')n(v')\alpha(v + \delta - v', v') \\
- \int_{v-\epsilon}^{v-\epsilon} dv' n(v - v')n(v')\alpha(v - v', v') + \int_{v-\epsilon}^{v-\epsilon} dv' n(v - v')n(v')\alpha(v - v', v') + \int_{v-\epsilon}^{v-\epsilon} dv' n(v - v')n(v')\alpha(v - v', v') + \int_{v-\epsilon}^{v-\epsilon} dv' n(v - v')n(v')\alpha(v - v', v').
\]
(A.5)

Performing a change of variables \(v' \to v - v'\) on the terms from the top of the integrals and combining like terms gives
\[
2(h^+(v + \delta) - h^+(v)) = 2\int_0^{v+\delta} dv' n(v')[n(v + \delta - v')\alpha(v + \delta - v', v')] \\
- n(v - v')\alpha(v - v', v') + \int_{v-\epsilon}^{v-\epsilon} dv' n(v')n(v + \delta - v')\alpha(v + \delta - v', v') \\
- n(v - v')\alpha(v - v', v') + \int_{v-\epsilon}^{v-\epsilon} dv' n(v - v')n(v')\alpha(v - v', v').
\]
(A.6)

As \(\delta \to 0\), continuity of \(n(v - v')\) and \(\alpha(v - v', v')\) ensures that the first and second integrands go to zero almost everywhere. They are non-zero, but finite, only when there is a discontinuity at \(n(v - v')\), so the discontinuities contribute nothing to the integral. This can be seen by isolating the portion of the integral containing the discontinuity and allowing the width to approach zero. The final term becomes an integral of vanishingly small width, now in a location where the integrand is non-singular, and also vanishes.

Thus aggregation in a system with discontinuities in the number density does not cause the formation of new discontinuities. Note that the number density was considered bounded except at \(v = 0\), and so this conclusion does not apply to systems with Dirac delta spikes in the number density such as monodisperse systems, where new particles of volume \(2v\), etc. are expected to arise.

References


