Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers and Chemical Engineering 29 (2005) 367–377

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Comparison and pitfalls of different discretised solution methods for population balance models: a simulation study

Ingmar Nopens*, Daan Beheydt, Peter A. Vanrolleghem

Biomath, Ghent University, Coupure Links 653, 9000 Gent, Belgium

Received 27 February 2004; received in revised form 20 October 2004; accepted 27 October 2004

Abstract

One way of solving population balance equations in a time efficient way, necessary for using the model in non-linear parameter estimation, process control or for combination with computational fluid dynamics (CFD), is by means of discretisation of the population property of interest. Several methods are available in literature, but which one should be used in which case? Do they all perform equally well or do some of them have unexpected pitfalls? In this study, three discretisation methods (the Hounslow method, the fixed pivot and the moving pivot) were compared for three different processes (pure aggregation, pure binary breakage into equally sized daughters and combined aggregation/breakage) and six different initial conditions (two monodispersions, one uniform polydispersion and three partially uniform polydispersions). Their performances were compared in terms of accuracy and calculation speed and their pitfalls addressed and explained where possible. Overall, it was concluded that the moving pivot with a geometric factor of 2 performs good in all studied cases (not for finer or coarser grids). The eventual choice of discretisation scheme is highly dependent on the particular goal of the study and is a compromise between accuracy and calculation speed.

PACS: 02.60

Keywords: Population balances; Discretisation; Aggregation; Breakage; Numerical solution

1. Introduction

Population balance models (PBM) have been widely applied to describe processes involving dynamical behaviour of population properties. Applications can be found in a variety of scientific areas such as crystallisation, flocculation of inorganic dispersed systems, polymerisation, precipitation, flotation, granulation, cell culture dynamics, bioflocculation and aerosol dynamics, to name but a few. Depending on the number of properties being described by the model, a PBM can be categorized as either one- or multidimensional. The general format of a one-dimensional number-based PBM looks like (Hulburt & Katz, 1964; Ramkrishna, 2000):

$$\frac{\partial n(x,t)}{\partial t} + \frac{\partial}{\partial x}(\dot{X}(x,t)n(x,t)) = h(x,t) \tag{1}$$

where x is a property of the individuals, in our case particle size (expressed as volume), $n(x, t)$ is the number-based property distribution function (L^{-3}), $\dot{X}(x,t)$ is the time derivative of the property x (T^{-1}) and $h(x, t)$ the net generation rate of particles ($L^{-3} \cdot T^{-1}$). The right-hand-side of Eq. (1) often contains integral functions of $n(x, t)$ describing aggregation and/or breakage processes, turning it into an integro-differential equation which is hard to solve analytically. Alternative, numerical, techniques to solve this type of equations are summarised in Ramkrishna (2000). One of these techniques resulting in an acceptable calculation time and accuracy is the discretisation of the particle size x. Acceptable

* Corresponding author. Tel.: +32 9 2645935; fax: +32 9 2646220.
E-mail address: ingmar.nopens@biomath.ugent.be (I. Nopens).
URL: http://biomath.UGent.be/∼ingmar.

0098-1354/$ – see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2004.10.007

calculation speeds are important when using the model for parameter estimation, process control or in combination with CFD. This was the authors' main drive to solely investigate discretisation techniques to solve the PBE in this work, along with the ease of implementation and compatibility with our existing modelling and simulation software platform WEST (Hemmis NV, Belgium).

Discretisation techniques divide the property range of interest into a finite number of classes (M), transforming the integro-differential equation into a set of M ordinary differential equations that can be solved simultaneously. Several discretisation schemes exist, mainly differing in terms of (1) freedom of discretisation grid choice and (2) conserved properties (at least two) during the discretisation (Hounslow, Ryall, & Marshall, 1988; Kumar & Ramkrishna, 1996a, 1996b; Litster, Smit, & Hounslow; 1995; Nopens & Vanrolleghem, 2003; Vanni, 2000). In this study three discretisation algorithms were compared in terms of accuracy to describe different processes (pure aggregation, pure breakage and combined aggregation/breakage) using different initial conditions and grid densities. Pitfalls were addressed and explained where possible.

2. The population balance model

Since this work fits in the authors' study of activated sludge flocculation, a PBM without growth, but with aggregation and breakage included was selected. The growth term can be omitted because biological growth dynamics are significantly slower than aggregation/breakage dynamics.

A Smoluchowski type aggregation model was used (Ramkrishna, 2000; Thomas, Judd, & Fawcett, 1999):

$$h(x,t)_{\text{agg}} = \frac{1}{2}\int_0^x \alpha\beta(x-x',x')n(x-x',t)n(x',t)\,\mathrm{d}x' - n(x,t)\int_0^\infty \alpha\beta(x,x')n(x',t)\,\mathrm{d}x' \qquad (2)$$

where $\beta(x,x')(L^3\cdot T^{-1})$ is the collision frequency for particles of volume x and x' and α (−) is the collision efficiency. The former describes the transport of particles towards one another, whereas the latter describes the probability that these collisions lead to aggregation taking into account short-range forces like van der Waals attraction, charge repulsion and hydrodynamic interaction. In this study an orthokinetic kernel $\beta(x-x',x)$ was borrowed from Spicer and Pratsinis (1996a):

$$\beta(x-x',x) = 0.31\bar{G}[(x-x')^{1/3}+x^{1/3}]^3 \qquad (3)$$

α was chosen to be a constant between 0 and 1. In the latter case, all collisions are considered to be successful. Other aggregation efficiency expressions are available in literature (Adler, 1981; Ducoste, 2002; Kusters, Wijers, & Thoenes, 1997) but not considered here since this was not the goal of this work.

Breakage models typically look like (Ramkrishna, 2000):

$$h(x,t)_{\text{break}} = \int_x^\infty n(x',t)S(x')\upsilon(x')\Gamma(x,x')\,\mathrm{d}x' - n(x,t)S(x) \qquad (4)$$

where $S(x)$ is the breakage rate of particles of size x (s^{-1}), $\upsilon(x')$ the average number of particles resulting from a breakage event and $\Gamma(x,x')$ the breakage distribution function. $S(x)$ was assumed to be a power law (Spicer & Pratsinis, 1996b):

$$S(x) = Ax^a \qquad (5)$$

where a is a constant (=1/3) and A is the breakage rate coefficient $(L^{-1}\cdot T^{-1})$. Binary breakage into equally sized daughters is assumed

$$\upsilon(x')\Gamma(x,x') = 2\delta(x-\tfrac{1}{2}x') \qquad (6)$$

Other more complex breakage functions can be found in literature (Ducoste, 2002; Konno, Aoki, & Saito, 1983; Kramer & Clark, 1999; Vanni, 2000) but were not considered in this particular study focusing on solution methods.

3. Discretisation techniques

In this study, three different discretisation techniques were compared: (1) the method of Hounslow et al. (1988), (2) the fixed pivot technique (Kumar & Ramkrishna, 1996a) and the moving pivot technique (Kumar & Ramkrishna, 1996b). In this section the different methods are briefly presented.

3.1. *The method of Hounslow*

Hounslow et al. (1988) used a geometric grid with factor 2 based on volume ($v_{i+1}=2v_i$, with v the particle volume) and were the first to develop a set of equations conserving both numbers and mass for purely aggregating systems:

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = N_{i-1}\sum_{j=1}^{i-2}2^{j-i+1}\beta_{i-1,j}N_j + \frac{1}{2}\beta_{i-1,i-1}N_{i-1}^2 - N_i\sum_{j=1}^{i-1}2^{j-i}\beta_{i,j}N_j - N_i\sum_{j=i}^{\infty}\beta_{i,j}N_j \qquad (7)$$

Litster et al. (1995) expanded the method for finer geometric grids, whereas Hill and Ng (1995) developed similar equations for breakup and finer grids. The main disadvantage of these methods is their inflexibility in terms of grid and conservation of distribution properties, which is restricted to numbers and mass. When the properties of interest are different, new equations need to be derived from scratch which is quite straightforward but time consuming.

3.2. The fixed pivot

In order to overcome the grid flexibility problem, Kumar and Ramkrishna (1996a) proposed a more general framework. The main difficulty that arises is that aggregation or breakage leads to particles whose property x does not coincide with one of the existing grid representative diameters or pivots. Two property balances are used to reallocate such particles to the adjoining pivots (schematically shown in Fig. 1). In that way two arbitrarily chosen properties of the distribution can be conserved. The set of equations needed to conserve both mass and numbers is given by (Kumar & Ramkrishna, 1996):

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \sum_{\substack{j,k \\ x_{i-1}\le(x_j+x_k)\le x_{i+1}}}^{j\ge k} \left[1-\frac{1}{2}\delta_{j,k}\right]\eta_i\beta_{x_j,x_k}N_jN_k - N_i\sum_{k}\beta_{x_i,x_k}N_k + \sum_{\substack{k \\ j\ge i}} n_{i,k}S(x_i)N_j - S(x_i)N_i \qquad (8)$$

where

$$\eta_i = \begin{cases} \dfrac{x_{i+1}-(x_j+x_k)}{x_{i+1}-x_i} & \text{when } x_i \le (x_j+x_k) \le x_{i+1}, \\ \dfrac{(x_j+x_k)-x_{i-1}}{x_i-x_{i-1}} & \text{when } x_{i-1} \le (x_j+x_k) \le x_i \end{cases} \qquad (9)$$

and

$$n_{i,k} = \int_{x_i}^{x_{i+1}} \frac{x_{i+1}-v}{x_{i+1}-x_i}\Gamma(v,x_k)\,\mathrm{d}v + \int_{x_{i-1}}^{x_i} \frac{v-x_{i-1}}{x_i-x_{i-1}}\Gamma(v,x_k)\,\mathrm{d}v \qquad (10)$$

It should be noted that Eqs. (8)–(10) are the simplified equations that can only be used to conserve numbers and mass and will, hence, yield identical results as the ones derived by Hounslow et al. (1988) when a geometrical grid with factor 2 (volume-based) is used. For the general equations of the fixed pivot technique, the reader is referred to Kumar and Ramkrishna (1996a).

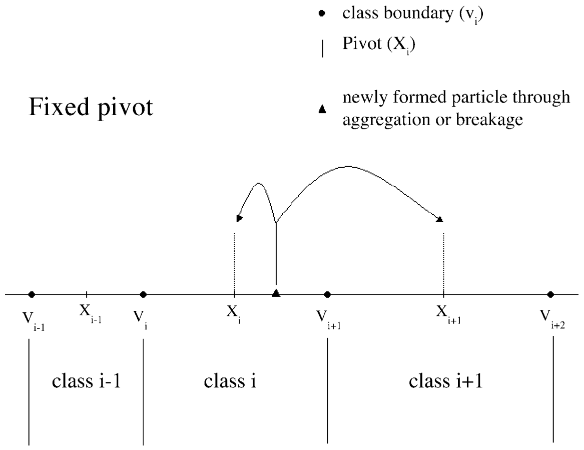

Fig. 1. Schematic representation of how the fixed pivot method deals with newly formed particles that do not coincide with an existing pivot.

3.3. The moving pivot

In order to overcome inaccurate model predictions in case of steep non-linear gradients present in the distribution, Kumar and Ramkrishna (1996b) proposed the so-called moving pivot technique. This technique accounts for the evolving non-uniformity of the distribution in each interval as a result of breakage and aggregation events by allowing a varying pivot location (Fig. 2). The arbitrary choice of grid was also maintained. Next to the equations for N_i, equations describing the changes in location of the pivots (x_i) are needed too. Essentially, the pivots move in a weighted manner towards the size of the newly formed particle. For the derivation of the equations, the reader is referred to Kumar and Ramkrishna (1996b). The equations conserving numbers and mass are given below:

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \sum_{\substack{j,k \\ v_i\le(x_j+x_k)\le v_{i+1}}}^{j\ge k} \left[1-\frac{1}{2}\delta_{j,k}\right]\beta_{x_j,x_k}N_jN_k - N_i\sum_{k=1}^{M}\beta_{x_i,x_k}N_k + \sum_{j\ge i} S(x_i)N_j\overline{B}^{(1)}_{i,j} - S(x_i)N_i \qquad (11)$$

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \frac{1}{N_i}\sum_{\substack{j,k \\ v_i\le(x_j+x_k)\le v_{i+1}}}^{j\ge k} \times \left[1-\frac{1}{2}\delta_{j,k}\right][(x_j+x_k)-x_i]\beta_{x_j,x_k}N_jN_k - \frac{1}{N_i}\sum_{j\ge i} S(x_j)N_j[\overline{B}^{(v)}_{i,j} - x_i\overline{B}^{(1)}_{i,j}] \qquad (12)$$

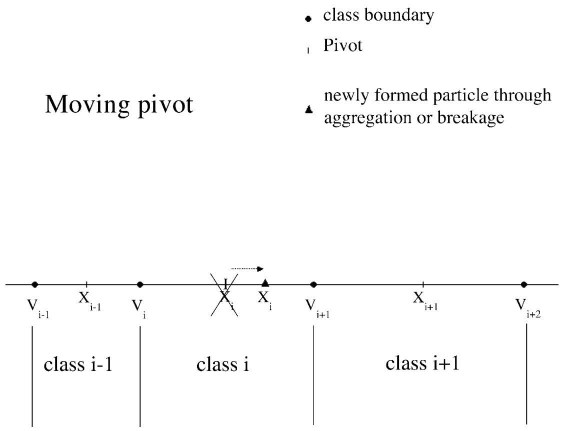

Fig. 2. Schematic representation of how the moving pivot deals with newly formed particles that do not coincide with an existing pivot.

where

$$\overline{B}_{i,j}^{(1)} = \int_{v_i}^{v_{i+1}} \Gamma(v, x_j)\,\mathrm{d}v \quad \text{and}$$

$$\overline{B}_{i,j}^{(v)} = \int_{v_i}^{v_{i+1}} v\Gamma(v, x_j)\,\mathrm{d}v \qquad (13)$$

4. Materials and methods

The three different discretisation algorithms that were described in the previous section, all conserving both numbers and mass, were implemented in the modelling and simulation platform WEST (Hemmis NV, Belgium).

The Hounslow algorithm is quite straightforward to implement. However, one should be cautious to set the boundary conditions properly to avoid mass leaks. At the breakage end, the breakage rate S for the lower boundary class should be explicitly set to 0. The same goes for the aggregation death term of the upper boundary class. An important consequence is that the summation in the final term of Eq. (7) should only take into account j-values up to the one but last class (and not the upper boundary class) in order to avoid a mass leak. This is illustrated in Fig. 3 for a simple grid of three classes. In order to avoid a mass leak from the upper boundary class (i.e. class 3) the three negative terms should be set to 0 since they cannot be born elsewhere because they are outside the particle range. However, the circled terms in the top two equations are the complementary terms of the collisions that result in particles outside the particle range, and, hence, these should also be put to 0 to avoid mass loss. It concerns all collisions with particles from the upper boundary class.

The implementation of the fixed pivot algorithm is a more tedious task and, here too, one should be cautious to set the boundary conditions properly. Particles should always be born (through either aggregation or breakage) between the smallest and the largest pivot. This means that some aggregation and breakage events must be ignored even though they would result in particles within the particle range.

The moving pivot is relatively easy to implement. There is, however, a subtle difference in boundary conditions since all particles born within the complete particle range can be taken into account. Hence, the covered range is between v_0 and v_M rather than between x_0 and x_{M-1}. Another practical, numerical, issue is the fact that none of the classes should be or become 0 since this would imply a division through 0 in Eq. (12). Therefore, a lower limit of 1E−13 was adopted in the initial conditions (also for the other algorithms). The value was chosen in order to not add significant mass to the system.

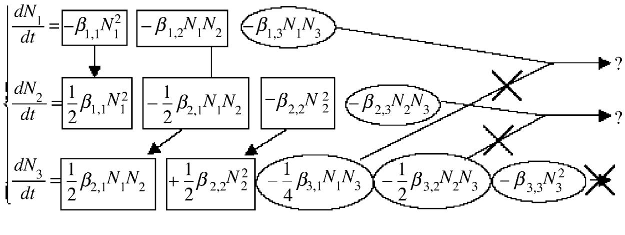
$\frac{dN_1}{dt} = -\beta_{1,1}N_1^2 \; -\beta_{1,2}N_1N_2 \; -\beta_{1,3}N_1N_3$
$\frac{dN_2}{dt} = \frac{1}{2}\beta_{1,1}N_1^2 \; -\frac{1}{2}\beta_{2,1}N_1N_2 \; -\beta_{2,2}N_2^2 \; -\beta_{2,3}N_2N_3$
$\frac{dN_3}{dt} = \frac{1}{2}\beta_{2,1}N_1N_2 \; +\frac{1}{2}\beta_{2,2}N_2^2 \; -\frac{1}{4}\beta_{3,1}N_1N_3 \; -\frac{1}{2}\beta_{3,2}N_2N_3 \; -\beta_{3,3}N_3^2$
?
?

Fig. 3. Illustration of the importance of correctly chosen boundary conditions of the Hounslow equations.

Table 1
Overview of different initial conditions that were used

Initial condition	Type of dispersion	Size range (μm^3)
1	Monodisperion	8.89E−14
2	Monodispersion	8.69E−17
3	Uniform polydisperion	1.70E−19 to 1.82E−10
4	Uniform polydisperion	1.70E−19 to 8.69E−17
5	Uniform polydisperion	3.56E−13 to 1.82E−10
6	Uniform polydisperion	1.74E−16 to 1.78E−13

For the application at hand the lower boundary of the entire size range was chosen to be 0.6 μm. The class boundary values were calculated based on volume and using a geometric factor of either 1.6, 2.0 or 2.4. The pivotal volumes were calculated as the arithmetic mean of the class boundary volumes. The pivotal diameters were calculated from the pivotal volumes assuming sphericity. The final class of which the pivotal diameter is smaller than 800 μm is chosen to be the upper boundary class. This resulted in, respectively, 25, 31 and 46 classes for the different geometric factors.

A fixed total particle volume of 1.82E−8 m^3 was chosen arbitrarily (=100 particles in the upper boundary class in the case of a geometric factor of 2.0). The different initial conditions for the geometric case with factor 2 are summarised in Table 1.

All initial conditions were defined for the case of a geometric factor of 2.0 and then recalculated for the other cases (geometric factors of 1.6 and 2.4) conserving both numbers and mass. This transformation is not always possible and depends on the relative location of the initial pivots of the boundary classes of both grids (problems only occur when particles are present in boundary classes, i.e. init3, init4 and init5). This can be illustrated by a simple example. Say a number of particles is present in the lower boundary class of a grid with geometric factor 2. If one wants to use a grid with geometric factor 2.4, one cannot redistribute these particles among the adjoining pivots since there is only one adjoining pivot (the volume of the particles to be redistributed is smaller than the volume of the smallest pivot of the 2.4 grid). In that case one cannot conserve two properties. These problems only occur with conversions to grids with geometric factor 2.4 (init3, 4 and 5).

The simulations were performed for three cases: (1) pure aggregation, (2) pure breakage and (3) combined aggregation/breakage. In the pure aggregation case, parameter values were chosen to be $\alpha = 1$, $\bar{G} = 100\,\mathrm{s}^{-1}$, $A = 0\,\mathrm{cm}^{-1}\,\mathrm{s}^{-1}$. For the pure breakage case, parameter values were chosen to be $\alpha = 0$, $\bar{G} = 100\,\mathrm{s}^{-1}$, $A = 1\mathrm{E}4\,\mathrm{cm}^{-1}\,\mathrm{s}^{-1}$. Finally, for the combined aggregation/breakage case, parameter values were chosen to be $\alpha = 0.2$, $\bar{G} = 100\,\mathrm{s}^{-1}$, $A = 1\mathrm{E}4\,\mathrm{cm}^{-1}\,\mathrm{s}^{-1}$. All simulations were performed on a Pentium IV-2.8 GHz ma-

Table 2
Overview of results for the pure aggregation case using the Hounslow algorithm

Init	Classes	Sim. time (s)	Comp. time (s)	SS/PSS	Sim. time to SS (s)
1	31	1.0E+4	1	SS	8.0E+3
2	31	3.0E+6	1	SS	2.7E+6
3	31	1.0E+5	1	SS	8.1E+4
4	31	1.0E+9	2	SS	3.0E+8
5	31	1.1E+3	1	SS	1.1E+3
6	31	1.0E+6	1	SS	8.4E+5

chine using the CVODE stiff solver (Cohen & Hindmarsh, 1994), unless otherwise stated. As default settings, the absolute tolerance was chosen to be 0 and the relative tolerance to be 0.005. The Adams linear multistep method and functional iteration method were used.

The simulation results are presented either as cumulative oversize numbers (CON), cumulative undersize numbers (CUN), cumulative oversize volume (COV) or cumulative undersize volume (CUV):

$$\mathrm{CON}(v,t) = \int_{v}^{\infty} n(v',t)\,\mathrm{d}v' \tag{14}$$

$$\mathrm{CUN}(v,t) = \int_{v}^{0} n(v',t)\,\mathrm{d}v' \tag{15}$$

$$\mathrm{COV}(v,t) = \int_{v}^{\infty} n(v',t)v'\,\mathrm{d}v' \tag{16}$$

$$\mathrm{CUV}(v,t) = \int_{v}^{0} n(v',t)v'\,\mathrm{d}v' \tag{17}$$

5. Results and discussion

5.1. The pure aggregation case

For a pure aggregation process, one expects all particles to move into the upper boundary class at *steady state* (SS). SS is defined to be reached when 99.99% of the particle volume is present in the upper boundary class and the number of particles in all other classes has dropped below 1. The total simulation times for the different cases were chosen in order to reach the SS. Results of the pure aggregation case for the Hounslow method, the fixed pivot and the moving pivot are summarised in, respectively, Tables 2–4.

In all cases the Hounslow method reaches SS without mass losses. However, the simulation time needed to reach SS is different for the different initial conditions. They seem to be dependent on the initial degree of aggregation of the system. The calculation speed is very high and seems to be independent of the initial condition. It should be noted that these results were obtained by using the correct boundary conditions. When the last summation in Eq. (7) is performed up to the upper boundary class, mass losses between 40 and 60% were observed.

Table 3
Overview of results for the pure aggregation case using the fixed pivot algorithm

Init	Classes	Sim. time (s)	Comp. time (s)	SS/PSS	Sim. time to SS (s)
1	25	1.0E+4	4	SS	8.2E+3
	31	1.0E+4	6	SS	8.0E+3
	46	3.0E+3	174	PSS	3.0E+3
2	25	3.0E+6	4	SS	2.9E+6
	31	3.0E+6	13	SS	2.7E+6
	46	3.0E+3	61	PSS	3.0E+3
3	31	1.0E+5	7	SS	8.1E+4
	46	1.0E+3	174	PSS	1.0E+3
4	31	1.0E+9	20	SS	3.0E+8
	46	1.0E+4	182	PSS	1.0E+4
5	31	1.1E+3	1	SS	1.1E+3
	46	1.0E+3	138	PSS	1.0E+3
6	25	1.0E+6	3	SS	9.0E+5
	31	1.0E+6	6	SS	8.4E+5
	46	4.0E+3	169	PSS	4.0E+3

The fixed pivot reaches SS for all cases using 31 classes. The simulation times required to reach steady state are the same as the ones observed using the Hounslow method and, hence, again depend on the initial condition. The calculation times needed to perform the simulations are, however, significantly larger, but still acceptable. Unlike the Hounslow algorithm, the calculation speed seems to be dependent on the initial condition (evidently, they are proportional to the required simulation time). The difference is probably due to the fact that quite some if-clauses are present in the fixed

Table 4
Overview of results for the pure aggregation case using the moving pivot algorithm for a simulation time of 1000 s

Init	Classes	Comp. time (s)	SS/PSS	Sim. time to SS (s)
1	25	14	SS	1000
	31	20	SS	1000
	46	52	PSS	1000
2	25	781	SS	1000
	31	1745	SS	1000
	46	4293	PSS	1000
3	31	14	SS	1000
	46	44	PSS	90
4	31	279933	SS	1000
	46	706719	PSS	1000
5	31	15	SS	1000
	46	44	PSS	230
6	25	20	SS	1000
	31	18	SS	1000
	46	49	PSS	1000

pivot algorithm implementation whereas these are directly integrated in the Hounslow equations.

In the cases using 25 classes the simulation time to reach SS is somewhat larger (but the same order of magnitude), which might be explained by the fact that the pivotal volume of the upper boundary class is somewhat larger compared to the 31 classes grid. The computation time is somewhat smaller compared to the cases using 31 classes (lower amount of equations to solve) and seems to be independent of the simulation time.

The cases with 46 classes did not reach the SS as defined above. However, after a certain simulation time, the system reaches another SS, with some particles being trapped in the one but last class. In order to distinguish this state from a normal SS, it was called a *pseudo steady state* (PSS), meaning that a steady state has been reached, but not all particles have reached the upper boundary class. This situation occurs when the system evolves into a state where no longer valid collisions are possible between particles of the one but last class and any other particle in the system. This invalidity is due to the fact that the sum of volumes of the colliding particles exceeds the volume of the upper boundary class pivot. Since aggregation is the only working mechanism here, there is no possibility for these particles to escape from their state and, hence, they are trapped in the one but last class. This phenomenon is illustrated in Fig. 4.

The PSS can only occur when geometric grids with factors smaller than 2 are used. Indeed, in the case of a geometric grid with factor 2, the particles of the one but last class can always aggregate with themselves, still producing particles that have a valid volume. This is illustrated in Fig. 5. The simulation times needed to reach the PSS are substantially shorter compared to the cases using 25 and 31 classes and, therefore, the total simulation times were adjusted accordingly. The PSS were somewhat different for the different initial conditions, i.e. the number of particles that are trapped are different. Especially when particles are initially present in the last but one class (init3 and init5), a considerably larger amount of particles is trapped. Although simulation times were substantially shorter, calculation times were larger for the cases using 46 classes due to the increased number of equations that needs to be solved.

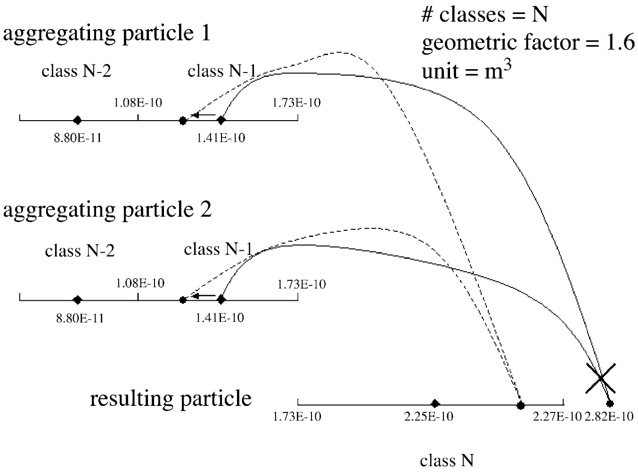

Fig. 4. Illustration of the pseudo steady state that occurs with the fixed/moving pivot using a geometric factor of 1.6.

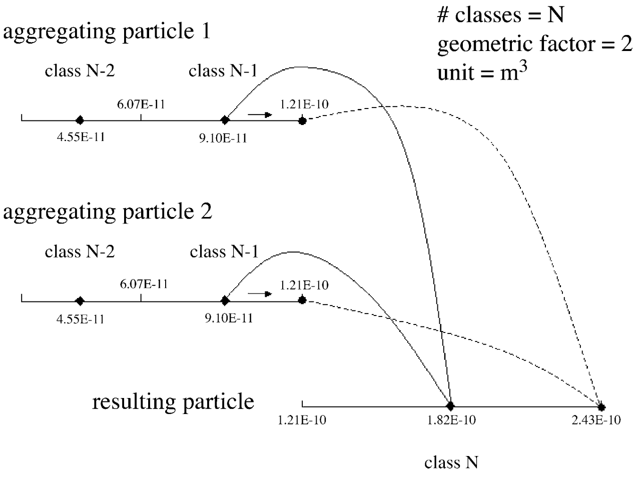

Fig. 5. Illustration that a pseudo steady state can never occur for a fixed/moving pivot using a geometric factor of 2.0.

The moving pivot reaches SS for all cases using 31 classes. The simulation times to reach SS are, however, substantially smaller compared to the ones that were needed in the Hounslow and fixed pivot cases. For example, to reach a state where 99.99% of the volume is present in the upper class takes 8200 s for init1 using either the Hounslow and fixed pivot algorithms, whereas for the moving pivot SS is reached in 1000 s. This is caused by the subtle difference in valid size range for particle formation between the different methods, i.e. the range between the smallest and the largest pivot for Hounslow and the fixed pivot versus the entire size range from the lower boundary up to the upper boundary of the system for the moving pivot. In the moving pivot case, a lot of aggregations are allowed that have to be ignored in the other two cases. For example, collisions with particles from the upper class are not ignored as long as the resulting particle's volume does not exceed the upper system boundary. Evidence of this can also be found in the fact that the pivot of the upper boundary class moves towards the upper system boundary in all simulated cases. The fact that the fixed pivot (and the Hounslow algorithm) slow down the simulation seems to be in disagreement with the findings of Kumar and Ramkrishna (1996b), who found a slower moving front when using the moving pivot. However, they only compared the methods when the front had not yet reached the upper classes of the system. Indeed, in some cases, where the upper classes are initially empty, the front indeed moves faster for the fixed and Hounslow algorithm. Once the upper classes get filled with particles, the abovementioned findings are observed, and the moving pivot will reach SS faster. Although simulation times to reach SS are substantially smaller, increased calculation times are required. This is most probably caused by the fact that twice as many equations need to be solved. The calculation time dependency on the initial condition is also much more pronounced. This is caused by the extremely small values of the pivots combined with extreme initial conditions, forcing the solver to take very small steps to avoid numerical errors. Especially when empty classes are initially present and get filled by a number of particles, one needs to force the solver to take small steps to force the pivots to stay within the class bounds. These small steps also cause the calculation time to increase, which is definitely a drawback of the method.

Similar results were found for the cases using 25 classes. The calculation times are substantially smaller due to the smaller set of equations that needs to be solved. No clear dependency on the initial condition was observed.

The cases using 46 classes again showed PSS, which can be explained in the same way as the fixed pivot case. Again, the number of particles trapped in the one but last class was different for the different initial conditions. It is very much governed by the movement of the pivots of the two upper classes. One way for the particles of the one but last class to escape is when the value of the pivot would drop below half of that of the upper boundary. However, this did not happen in any of the cases. The number of particles that eventually gets trapped will be determined by the number of particles present in the classes just below the one but last class. Also, the decrease of particles in these classes is highly influenced by the movement of the pivot of the upper boundary class. When the latter remains rather small, collisions with these particles will still be valid and will result in a fast depletion of them, making them unavailable for collisions with particles from the one but last class, causing these to get trapped. The calculation times are again higher compared to the fixed pivot and really blow up in extreme cases.

Fig. 6 shows the CON(v, t) for the different grid densities and solution techniques for init1 at SS/PSS. Due to the difference in pivotal volume of the upper boundary class for the different grids, the total number of particles (N_{tot}) is different. Adopting the case with geometric factor 2 (31 classes) as reference, N_{tot} will be smaller than 100 for upper boundary pivotal volumes that are larger than the reference pivotal volume.

Hounslow and the fixed pivot using 31 classes exhibit exactly the same behaviour (coinciding CON-curves). At the time of assumed SS, there are still some particles that have not reached the upper boundary class, which is also the case for the fixed pivot using 25 classes (indicated by the arrows). In all other cases the particles have reached the upper boundary class, except for the fixed pivot, where one can clearly observe the PSS, and the moving pivot using 46 classes (less clear). Fig. 7 shows that the total volume is conserved for all cases. Here too, for the cases using 46 classes the PSS can be observed. All other initial conditions yield similar results (not shown).

To conclude, pure aggregation processes are best simulated using a moving pivot technique in combination with a grid using a geometric factor of 2 or higher. Hounslow and the fixed pivot slow down the aggregation rate due to the smaller allowed particle range. Hence, the numerical solution significantly affects the model outcome. Moreover, when using a finer grid (46 classes), particle entrapment in the one but last class occurs resulting in a PSS. A coarser grid (25 classes) sometimes results in a significantly faster calculation time, but is, on the other hand, less accurate. The choice is, therefore, governed by the initial condition and the goal of the simulation work (i.e. parameter estimation, control, combination with CFD, etc.).

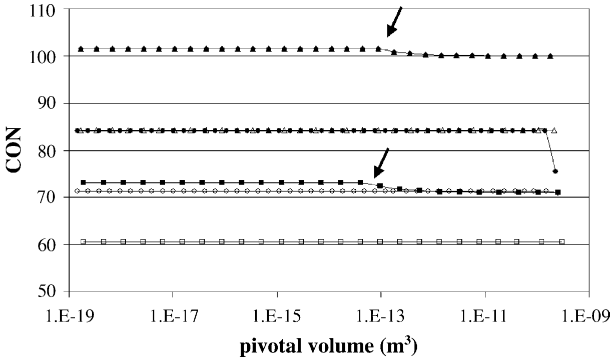

Fig. 6. CON(v, t) for init1 for the pure aggregation case: Hounslow (◇), fixed 25 (■), fixed 31 (▲), fixed 46 (●), moving 25 (□), moving 31 (△) and moving 46 (○).

5.2. The pure breakage case

For a pure breakage process (in this work binary into equally sized daughters), one would expect all particles to move into the lower boundary class at SS. Hence, SS is defined to be reached when all particles are present in the lower boundary class and the number of particles in the second class has dropped below 0.1. As above, the total simulation times for the different cases were chosen in order to reach the defined SS. Results of the pure breakage case for the fixed pivot and the moving pivot are summarised in, respectively,

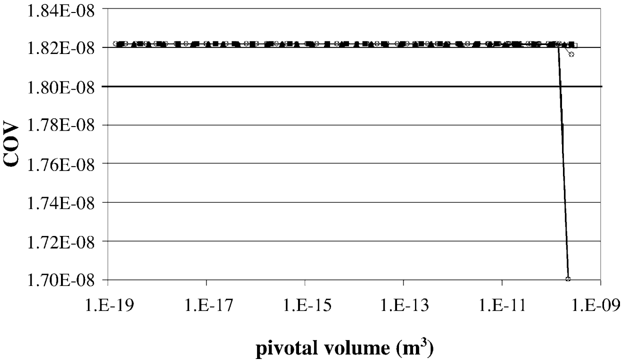

Fig. 7. COV(v, t) for init1 for the pure aggregation case: Hounslow (◇), fixed 25 (■), fixed 31 (▲), fixed 46 (●), moving 25 (□), moving 31 (△) and moving 46 (○).

Table 5
Overview of results for the pure breakage case using the fixed pivot algorithm and a simulaton time of 6000 s

Init	Classes	Comp. time (s)	SS/PSS	Sim. time to SS (s)
1	25	577	SS	6000
	31	1217	SS	4900
	46	17184	PSS	4800
2	25	589	SS	6000
	31	1222	SS	4800
	46	19725	PSS	4700
3	31	1544	SS	4900
	46	4339	PSS	4800
4	31	1214	SS	4700
	46	3692	PSS	4700
5	31	8693	SS	4800
	46	4377	PSS	4800
6	25	575	SS	6000
	31	1216	SS	4800
	46	3847	PSS	4800

Tables 5 and 6. The Hounslow case was not considered since the equations were developed for pure aggregation.

The fixed pivot method using 31 classes reaches SS in all cases without mass losses. However, the simulation time needed to reach SS is slightly different for the different initial conditions. They seem to be dependent on the initial degree of aggregation of the system. Calculation speed is rather low compared to the pure aggregation case and seems also to be dependent on the initial condition.

Using a coarser grid (25 classes) increases the required time to reach SS. The reason for this is quite obvious and is related to the type of breakage that was used (binary into equally sized daughters). When a particle of class i breaks, it will produce two particles of class $i-1$ when a geometric grid with factor 2 is used. No reallocation of particles is needed since the particles that are born always coincide with an existing pivot. However, this is not the case when a coarser grid is used. A particle of class i that breaks up will produce particles somewhere in between the pivots of class i and $i-1$, implying that they will be partly reallocated to class i, which slows down the breakage process. Again, the model output is sensitive to the numerical method. The calculation times are again lower due to the lower number of equations. Here too, they are dependent on the initial condition.

Table 6
Overview of results for the pure breakage case using the moving pivot algorithm and a simulation time of 6000 s (empty lines indicate slow calculations or numerical problems)

Init	Classes	Comp. time (s)	SS/PSS	Sim. time to SS (s)
1	25	–	–	–
	31	350	SS	4800
	46	–	–	–
2	25	–	–	–
	31	204	SS	4800
	46	–	–	–
3	31	256	SS	4900
	46	778	PSS	4800
4	31	201	SS	4700
	46	615	PSS	4500
5	31	257	SS	4900
	46	–	–	–
6	25	–	–	–
	31	202	SS	4800
	46	–	–	–

When using a finer grid (46 classes) the phenomenon of particle entrapment again occurs, resulting in a PSS. The reasoning is exactly the same as in the aggregation case, but now occurs at the lower end of the particle range. The PSS was defined to be reached when the number of particles in the third class drops below 0.1. Simulation times required for reaching PSS are in the same order of magnitude as the ones for 31 classes. Calculation times are again higher due to the increased number of equations to be solved simultaneously. They are dependent of the initial condition.

The moving pivot using 31 classes yields exactly the same result as the fixed pivot for 31 classes. This is caused by the specific type of breakage that is used (i.e. binary breakage). Hence, the simulation times to SS are the same. However, the calculation speed is faster compared to the fixed pivot, which seems odd since twice as many equations need to be solved. A possible explanation is that the fixed pivot needs to calculate reallocation fractions. A dependency on the initial condition is also observed.

When coarsening the grid for the moving pivot technique, the CVODE-solver experiences problems due to the fact that it attempts to reduce the step size to levels where no convergence is found. Also other solvers with adaptive step size failed. In the end the simulation was performed using a Runge Kutta fourth order solver (RK4) using a small step size (1E−4), resulting in long calculation times (several days), which is not the intention of the discretised methods under study. Therefore, the simulations were not completed and it is recommended not to use the moving pivot with a coarse grid for this breakage problem.

Using finer grids combined with the moving pivot also resulted in problems with the CVODE-solver in some cases. Neither changing solver settings nor using a different solver could solve the numerical problems. For the same reason mentioned before, the simulations with the RK4 solver were not completed. In the cases where it did work, PSSs were observed and calculation times were larger compared to the fixed pivot using 31 classes, due to the increased number of equations.

Fig. 8 shows the CUN(v, t) for the different grids and solution techniques for init1 at SS (note that no results are shown for the moving pivot using 25 and 46 classes due to the numerical problems). The difference in pivotal volume

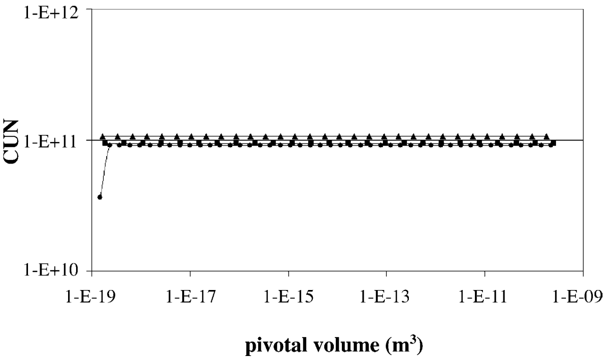

Fig. 8. CUN(v, t) for init1 for the pure breakage case: fixed 25 (■), fixed 31 (▲), fixed 46 (●) and moving 31 (△).

of the lower boundary class for the different cases causes the total number of particles (N_{tot}) to be different. However, the fixed and moving pivot with 31 classes yield exactly the same result, obviously caused by the binary breakage into equally sized daughters in combination with a geometric factor of 2. The fixed pivot with a fine grid exhibits particle entrapment.

Fig. 9 shows that volume is conserved and exactly the same for all cases. Again, one can observe the fixed pivot with 46 classes suffering from particle entrapment.

All other initial conditions yielded similar results (not shown).

In conclusion, the pure binary breakage process into equally sized daughters is best solved using a geometric grid with factor 2. The solution method does not really matter since all result in exactly the same solution in exactly the same simulated time. However, the moving pivot algorithm needs the least calculation time.

5.3. The combined aggregation/breakage case

The steady state that develops in combined aggregation/breakage cases is solely dependent on the model parameters and was shown to be independent of the initial condition when the total volume in the system is constant (Chen, Fisher, & Berg, 1990). For this case the Hounslow equations were accompanied with breakage equations, similar to those used by the fixed pivot, but less general. Results of the combined case for the Hounslow method, the fixed pivot and the moving pivot are summarised in, respectively, Tables 7–9.

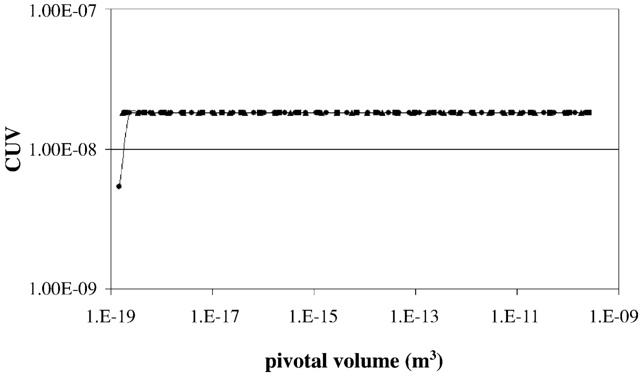

Fig. 9. CUV(v, t) for init1 for the pure breakage case: fixed 25 (■), fixed 31 (▲), fixed 46 (●) and moving 31 (△).

Table 7
Overview of results for the combined case using the Hounslow algorithm and a simulation time of 500 s

Init	Classes	Comp. time (s)	Sim. time to SS (s)
1	31	11	220
2	31	11	290
3	31	11	240
4	31	12	280
5	31	11	250
6	31	11	250

The Hounslow method results in exactly the same SS for all cases. The simulation time to SS is slightly variable, but the calculation times are similar and very fast.

The fixed pivot using 31 classes yields the same results as the Hounslow method in terms of SS and simulation time needed to reach SS. The calculation time is, however, significantly larger and depends on the initial condition. This larger calculation time is partly caused by the difference in aggregation equations, but also by the more general implementation of the breakage equations compared to the Hounslow method.

Coarsening the grid results in a slightly different SS, especially in the tail accuracy (Figs. 10 and 11). Simulated number concentrations in both the upper and lower tail are higher. The simulation time to reach SS is similar. Calculation times are smaller due to the smaller number of equations. Refinement of the grid also results in a slightly different SS in the tails. Simulated number concentrations are now lower in the upper tail, whereas they are still higher in the lower tail. Simulation times are similar and calculation times larger.

Table 8
Overview of results for the combined case using the fixed pivot algorithm and a simulation time of 500 s (all reach a normal SS)

Init	Classes	Comp. time (s)	Sim. time to SS (s)
1	25	60	220
	31	753	220
	46	519	220
2	25	62	290
	31	148	290
	46	579	290
3	31	143	240
	46	523	240
4	31	146	280
	46	534	280
5	31	145	250
	46	541	250
6	25	60	250
	31	142	250
	46	533	250

Table 9
Overview of results for the combined case using the moving pivot algorithm and a simulation time of 500s (all reach a normal SS)

Init	Classes	Comp. time (s)	Sim. time to SS (s)
1	25	–	–
	31	96	220
	46	2159	220
2	25	2969	290
	31	129	290
	46	35820	290
3	31	96	240
	46	1389	240
4	31	205	280
	46	–	–
5	31	87	250
	46	868	250
6	25	2955	250
	31	92	250
	46	2133	250

The moving pivot using 31 classes yields similar results in the lower tail as the Hounslow and fixed pivot with 31 classes, but predicts smaller number concentrations in the upper tail (Figs. 10 and 11). These findings were also reported by Kumar and Ramkrishna (1996b) and Nopens and Vanrolleghem (2003), who concluded that the moving pivot algorithm is more accurate even for coarse grids. Indeed, one observes that using the fixed pivot with finer grids results in solutions that move towards the moving pivot solution. This difference in accuracy was proved to become important when performing parameter estimations, since the different methods resulted in significantly different parameter estimates for exactly the same model (Nopens, Koegst, Mahieu, & Vanrolleghem, 2005). The simulation times needed to reach SS are similar, whereas calculation speed is higher compared to the fixed pivot but lower compared to the Hounslow algorithm. Hence, the question to the user is whether the increased accuracy justifies the increased calculation time. For a single simulation the answer is yes, but for an optimisation of 300 runs, the answer might not be so straightforward.

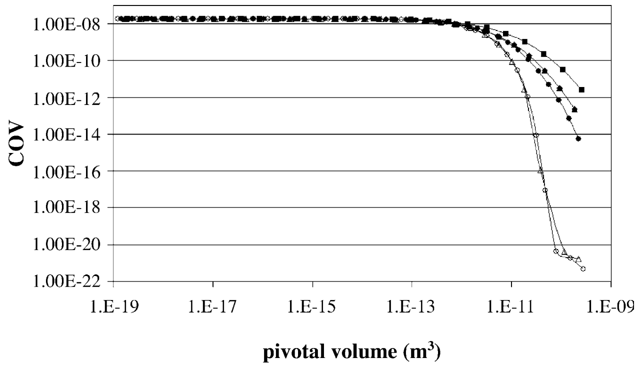

Fig. 10. COV(v, t) for init1 for the combined case: Hounslow (◇), fixed 25 (■), fixed 31 (▲), fixed 46 (●), moving 31 (△) and moving 46 (○).

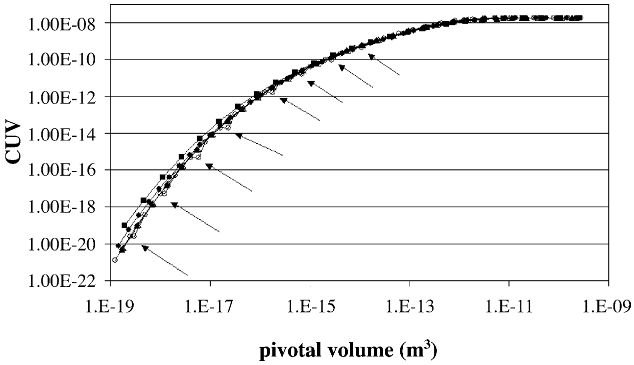

Fig. 11. CUV(v, t) for init1 for the combined case: Hounslow (◇), fixed 25 (■), fixed 31 (▲), fixed 46 (●), moving 31 (△) and moving 46 (○).

Coarsening the grid resulted in problems with the CVODE solver for init1. In the cases where it was possible to perform the simulation, no significant difference in solution was observed. However, the calculation time increased significantly.

Refining the grid resulted in similar SS and required the same simulation times. However, the accuracy is not significantly improved by refining the grid and the calculation time increases significantly. Moreover, empty classes appear in the distribution due to the specific pivot movement as was reported earlier by Nopens and Vanrolleghem (2004) and as is indicated by the arrows in Fig. 11. When pivot x_i becomes smaller than twice the value of the boundary of the underlying class v_{i-1}, particles break into class $i-2$ instead of class $i-1$. Init4 resulted in very long calculation times and could not be completed.

When comparing the different initial conditions, it was observed that the fixed pivot with 31 classes yields exactly the same steady state distribution, independent of the initial distribution. Moreover, the movement of the pivots and the steady state distributions when using the moving pivot with 31 classes are also independent of the investigated initial conditions.

In conclusion, the moving pivot with a geometric factor of 2 seems to be the most accurate solution method for the combined aggregation/breakage case. However, the goal of the study will determine whether the increased accuracy justifies the increased calculation time.

6. Conclusions

Three different solution methods for solving a PBE based on discretisation (the Hounslow method, the fixed pivot and the moving pivot) were compared for three different processes (pure aggregation, pure breakage and combined aggregation/breakage) using six different initial conditions with a fixed total system volume.

For pure aggregation, the moving pivot with a geometric factor of 2 turned out to be the best solution method.

The Hounslow method and the fixed pivot slowed down the aggregation rate due to the fact that some aggregations are (mathematically) not allowed. In other words, the numerical technique influences the model output. When using a finer grid, both the fixed and moving pivot suffered from particle entrapment in the one but last class. A coarser grid (25 classes) sometimes resulted in a significantly faster calculation time, but was, on the other hand, less accurate. The choice is, therefore, governed by the initial condition and the goal of the simulation work (i.e. parameter estimation, control, combined with CFD, etc.).

The pure binary breakage processes into equally sized daughters is best solved using a geometric grid with factor 2. Coarser grids resulted in slower breakage for the fixed pivot case and in numerical difficulties in the moving pivot case, whereas finer grids gave rise to particle entrapment. The fixed and moving pivot resulted in exactly the same solution in exactly the same simulated time. However, the moving pivot algorithm needs the least calculation time and might, therefore, be preferred.

The moving pivot with a geometric factor of 2 is the most accurate solution method for the combined aggregation/breakage case. The Hounslow and fixed pivot method resulted in a less accurate approximation of the steady state simulation solution, although Hounslow's algorithm performed the simulation faster. Refining the grid using the fixed pivot resulted in an improvement of the accuracy, but it still is worse compared to the moving pivot. In the end, the goal of the study (including the desired accuracy) will determine whether the increased accuracy justifies the increased calculation time.

Acknowledgements

This research was financially supported by the Fund for Scientific Research-Flanders (project G.0032.00) and the Ghent University Research Fund (BOF 01111001).

References

Adler, P. (1981). Heterocoagulation in shear flow. *Journal of Colloids and Interface Science*, *83* (1), 106–115.

Chen, W., Fisher, R., & Berg, J. (1990). Simulation of particle size distribution in an aggregation-breakup process. *Chemical Engineering Science*, *45* (9), 3003–3006.

Cohen, S., & Hindmarsh, A. (1994). *CVODE user guide*. Lawrence Livermore National Laboratory.

Ducoste, J. (2002). A two-scale PBM for modeling turbulent flocculation in water treatment processes. *Chemical Engineering Science*, *57*, 2157–2168.

Hill, P., & Ng, K. (1995). New discretisation procedure for the breakage equation. *AIChE Journal*, *41* (5), 1204–1216.

Hounslow, M., Ryall, R., & Marshall, V. (1988). A discretized population balance for nucleation, growth and aggregation. *AIChE Journal*, *34* (11), 1821–1832.

Hulburt, H., & Katz, S. (1964). Some problems in particle technology. A statistical mechanical formulation. *Chemical Engineering Science*, *19*, 555–574.

Konno, M., Aoki, M., & Saito, S. (1983). Scale effect on breakup process in liquid–liquid agitated tanks. *Journal of Chemical Engineering of Japan*, *16* (4), 312–319.

Kramer, T., & Clark, M. (1999). Incorporation of aggregate breakup in the simulation of orthokinetic coagulation. *Journal of Colloid and Interface Science*, *216*, 116–126.

Kumar, S., & Ramkrishna, D. (1996a). On the solution of population balance equations by discretisation I. A fixed pivot technique. *Chemical Engineering Science*, *51* (8), 1311–1332.

Kumar, S., & Ramkrishna, D. (1996b). On the solution of population balance equations by discretisation II. A moving pivot technique. *Chemical Engineering Science*, *51* (8), 1333–1342.

Kusters, K., Wijers, J., & Thoenes, D. (1997). Aggregation kinetics of small particles in agitated vessels. *Chemical Engineering Science*, *52* (1), 107–121.

Litster, J., Smit, D., & Hounslow, M. (1995). Adjustable discretised population balance for growth and aggregation. *AIChE Journal*, *41* (3), 591–603.

Nopens, I., Koegst, T., Mahieu, K., & Vanrolleghem, P. (2005). Population balance model and activated sludge flocculation: from experimental data to a calibrated model. *AIChE Journal*.

Nopens, I., & Vanrolleghem, P. (2003). Comparison of discretisation methods to solve a population balance model of activated sludge flocculation including aggregation and breakage. In *Proceedings of the IMACS Fourth MATHMOD Conference*.

Nopens, I., & Vanrolleghem, P., in press. Discretising a population balance model with binary breakage using the moving pivot technique. *Chemical Engineering Science*.

Ramkrishna, D. (2000). *Population balances: theory and applications to particulate systems in engineering*. London, UK: Academic Press.

Spicer, P., & Pratsinis, S. (1996a). Coagulation and fragmentation: universal steady-state particle size distribution. *AIChE Journal*, *42* (6), 1612–1620.

Spicer, P., & Pratsinis, S. (1996b). Shear-induced flocculation: the evolution of floc structure and the shape of the size distribution at steady state. *Water Research*, *30* (5), 1049–1056.

Thomas, D., Judd, S., & Fawcett, N. (1999). Flocculation modelling: a review. *Water Research*, *33* (7), 1579–1592.

Vanni, M. (2000). Approximate population balance equations for aggregation-breakage processes. *Journal of Colloid and Interface Science*, *221*, 143–160.