

# Moving Finite Elements Method Applied to Dynamic Population Balance Equations

**Belmiro P. M. Duarte**

Dept. of Chemical Engineering, ISEC, Polytechnic Institute of Coimbra, R. Pedro Nunes,  
3030-199 Coimbra, Portugal

**Cristina M. S. G. Baptista**

Dept. of Chemical Engineering, University of Coimbra, R. Sílvio Lima, Pólo II,  
3030-790 Coimbra, Portugal

DOI 10.1002/aic.11384

Published online January 7, 2008 in Wiley InterScience (www.interscience.wiley.com).

*A moving finite elements scheme is developed and used for solving 1-D dynamic population balance equations (PBE). The method stands on the weighted finite-elements based approach, and the local solutions are represented by cubic Hermite polynomials. The weighting function is the gradient of residuals with respect to time derivatives of the solution at the nodes and nodal velocities. The general PBE considered includes nucleation, growth, aggregation and breakage terms. The accuracy of the moving finite elements method (MFEM) is evaluated by comparing the results to the analytical solution in problems involving combinations of the first three phenomena considered. The formulation addressed was successful when used for solving a two-phase system representing a semibatch precipitation reactor. The MFEM enables one to achieve accurate results at reasonable CPU times, thus, showing to be adequate for these kind of problems. © 2008 American Institute of Chemical Engineers AICHE J, 54: 673–692, 2008*

*Keywords: moving finite elements, population balance equations, adaptive mesh, particulate systems*

## Introduction

During recent decades the particulate and emulsion processes gained great prominence in Chemical Engineering, mainly because they became economically advantageous and technologically attractive for some production units/goods. These systems, designated as particulate systems, are characterized by both a continuous and a dispersed phase. The analysis of these systems aims at describing the behavior of the particles' population distribution forming the dispersed phase and the continuous environment surrounding it. This popula-

tion is commonly represented by an extensity density, usually the number of particles, dependent on the spatial dimension that represents them, such as the volume or the length, here called characteristic dimension. The population balance equations allow to calculate the particle-size distribution (PSD) dynamics, which is essential to understand the processes. Population balance equations are of use in a vast number of fields ranging from astrophysics to chemistry, chemical engineering and biophysics. Typical chemical engineering applications cover crystallization systems, gas-liquid and liquid-liquid dispersion columns, and some grinding units. Moreover, with the increasing relevance of biochemistry and biophysics in chemical engineering other problems have been addressed, such as the cells growth behavior<sup>1</sup> and the simulation of emulsion based processes.<sup>2</sup>

Correspondence concerning this article should be addressed to B. P. M. Duarte at [bduarte@isec.pt](mailto:bduarte@isec.pt).

The general PBE described by Eqs. 1–3<sup>3,4</sup> represents four different phenomena—particles growth, nucleation, aggregation and breakage

$$\frac{\partial n(v,t)}{\partial t} = -\frac{\partial[G(v) n(v,t)]}{\partial v} + \frac{1}{2} \int_0^v n(v-v',t) n(v',t) q(v-v',v') dv' - n(v,t) \int_0^\infty n(v',t) q(v,v') dv' + \int_v^\infty b(v') \sigma(v') \eta(v',v) n(v',t) dv' - b(v) n(v,t) + S(v,t) \quad (1)$$

$$n(0,t) = \beta(n(v,t), v, t) \quad (2)$$

$$n(v,0) = f(v) \quad (3)$$

where  $v$  stands for the characteristic dimension of the PSD,  $n(v,t)dv$  is the number of particles with size between  $v$  and  $v + dv$ ,  $G(v)$  is the growth rate of particles,  $q(v',v)$  is the aggregation kernel rate,  $b(v)$  is the breakage rate,  $\sigma(v)$  stands for the distribution of the number of particles produced due to the breakage of one particle of characteristic dimension  $v$ ,  $\eta(v,v')$  is the daughter distribution of particles produced by breakage,  $S(v,t)$  is the nucleation rate,  $\beta(\bullet)$  is the boundary condition, and  $f(v)$  is the initial size distribution. The first term on the rhs of Eq. 1 accounts for growth of particles, the second term represents the production of particles of characteristic dimension  $v$  due to the aggregation of particles of  $v - v'$  and  $v'$ , the third quantifies the particles disappearance due to the aggregation phenomena, the fourth represents the production of particles of characteristic dimension  $v$  by breakage, the fifth the consumption due to breakage, and the last accounts for the nucleation phenomena.

In recent years, as the particulate based processes gained relevance in the chemical industry, researchers have been paying an increasing attention to these problems. Moreover, since many of the processes are used for manufacturing high-value products, the development of efficient control strategies became essential. The design of model based control strategies cannot be achieved if process behavior is not known, and this information can be obtained by solving PBE (see the works of El-Farra et al.<sup>5</sup> and Immanuel and Doyle III<sup>6</sup>). In most cases the analytical solution of the PBE is not available, therefore, its numerical solution has emerged as a demanding challenge and a requisite for implementing sophisticated control strategies.

The general PBE has the form of an integropartial-differential hyperbolic equation whose numerical solution is not easy to reach, particularly due to instability and numerical diffusion. Moreover, it is rather common that its solution gives rise to moving fronts and sharp changes originated by either steep initial distributions or the combination of the particulate mechanisms involved. To cope with the development of fronts that travel all around the spatial domain moving grid techniques can be used allowing one to reach accurate results without a prohibitive increase in the computational time required.

This article is organized as follows. The next section presents a review of previous works focused on the numerical solution of PBE equations, and on the moving finite elements method applied to the solution of partial differential equations (PDE). The approach proposed for solving the PBE is developed in the following section and its performance is tested in several case studies that involve growth, nucleation and aggregation terms. Later in the article this formulation is extended to systems described by PBE coupled with other extensity balances representing the dynamics of the continuous phase. The conclusions are summarized in the last section.

### Literature review

The classification of numerical approaches for PBE solution is perfectly established in the literature.<sup>3,4</sup> It contains two categories:

- Methods where global basis functions associated to time dependent coefficients are used to approximate the solution. The sum of the basis functions weighted by local time dependent coefficients will satisfy the solution of the PBE. This group of strategies encompasses all finite-element schemes, those based on global functions and those based on local approximations.
- Methods which are based on the discretization of the spatial dimension into a given number of intervals, thus, converting the original PDE into a set of ordinary differential equations (ODE) later solved with respect to the number of particles contained in each interval, commonly designated as bin. This framework is usually referred as discretized population balances (DPB) methods.

The state of the art on DPB methods can be found in the remarkable work of Ramkrishna.<sup>3</sup> The first schemes in the literature denoted the difficulty in conserving the volume and droplet number,<sup>7</sup> and this led to the development of methodologies to overcome this problem. However, most of them are problem dependent since they use properties of the PBE to derive the equations arising from discretization (see the work of Hill and Ng<sup>8</sup>). This disadvantage was overcome by Kumar and Ramkrishna<sup>9</sup> with a strategy that ensures internal consistency of two predefined moments of the population and some generality in treatment (see also the work of Kumar and Ramkrishna<sup>10</sup>). The key concept introduced to guarantee internal consistency is called pivot—a discrete size delimiting a bin—which allows one to convert the PBE into a set of partial differential equations by redistributing the total property associated to each bin. This framework is able to provide the desired population distribution instead of sophisticated approximations of the density number.

Problems involving growth and nucleation are demanding challenges for fixed grid approaches, such as the fixed pivot discretization scheme, mainly due to the development of moving fronts not accurately described by coarse fixed grids. To overcome this problem Hounslow et al.<sup>11</sup> and Marchal et al.<sup>12</sup> used a set of expressions based on finite difference approximations. To deal with the same growth and nucleation phenomena Kumar and Ramkrishna<sup>13</sup> extended the pivot method by combining it with the method of characteristics

enabling it to handle any grid distribution, and, therefore, avoiding instability problems. The key idea behind this approach is to derive discrete equations taking into account the velocity of the moving fronts, and this information is obtained from the equation of characteristics corresponding to the original PBE. Recently Bürger et al.<sup>14</sup> also used a finite differences based scheme to reach a solution for a problem dealing with the wear of steel balls in grinding mills.

Gelbard and Seinfeld<sup>15</sup> pioneered the derivation of local finite-elements approaches with regard to exploiting the flexibility of basis functions in capturing moving fronts and discontinuities, thereby improving the performance of the methods available to accurately describe those features. However, at the time, this class of methods presented a lower computational efficiency, due to the complexity of the algorithms involved and the limitations in terms of CPU velocity. The exponential increase in computational velocity together with an improved robustness of ODE integrators provided finite-elements methods with a remarkable applicability. This feature combined with the generality and the accuracy they guarantee increased the interest of other researchers to this family of methods, thereby improving it and demonstrating its ability to cope with different types of particulate phenomena. Among others, Nicmanis and Hounslow<sup>16</sup> proposed a Galerkin finite elements approach, based on cubic Lagrange polynomials to describe the basis functions. Later, Rigopoulos and Jones<sup>4</sup> derived and used a finite-elements based scheme standing on orthogonal collocation combined with local linear approximations. An approximated solution methodology based on the method of characteristics was used by Lee et al.<sup>17</sup> to solve the model of a semibatch precipitation reactor and derive a model predictive control strategy. Immanuel et al.<sup>18</sup> also used with success a finite-elements scheme coupled with local approximations given by fifth order Legendre polynomials.

The literature on the application of adaptive grid strategies to PBE solution is scarce. However, in problems where the initial distribution or the combination of phenomena involved generates moving fronts and sharp zones, adaptive grids are a good choice. They allow to move and concentrate the nodes representing the bins boundaries in the zones where an indicator of the solution magnitude, such as the first order spatial derivative, is higher, therefore, avoiding the use of tight fixed grids in the whole domain. This reduces the CPU time without compromising the accuracy of the solution. The moving pivot approach proposed by Kumar and Ramkrishna<sup>10</sup> falls into this group of methods. To preserve the number of ordinary differential equations arising from discretization, the positions of the pivots are allowed to move in order to follow the changes in the number density, according to the ratio between the moments of order 1 and 0. This strategy was successfully used by Crowley et al.<sup>19</sup> to solve the PBE describing the dynamics of an emulsion polymerization process enclosed in the calculation of the optimal control strategy. To minimize the finite domain error when solving problems that include breakage and aggregation Attarakih et al.<sup>20</sup> combined moving grids, based on a moving pivot scheme with a procedure to locate the position of the nodes. Eyre et al.<sup>21</sup> derived a spline collocation technique complemented with an adaptive grid. Mahoney and Ramkrishna<sup>22</sup>

combined a Galerkin finite-elements formulation with the method of characteristics enabling it to track discontinuities by associating its position to the position of a given node. A moving finite elements explicit approach was presented by Tsang and Rao<sup>23</sup> and used to solve PBE.

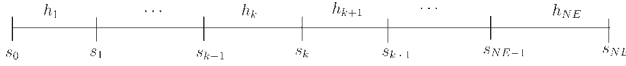
The seminal idea of using the moving finite elements method for solving evolutionary PDE is owed to Miller and Miller<sup>24</sup> (complimentarily see the work of Miller and Miller<sup>25</sup>). This methodology is based on computing simultaneously the node positions, and the solution of the PDE at the nodes in order to minimize the square norm of the residuals across the spatial domain. The domain is partitioned into finite elements, delimited by nodes, and the local approximations of the solution in each element are piecewise linear polynomials. The ODE system arising from the application of the method of lines is solved with respect to nodal velocities and time dependent coefficients associated to local basis functions, which represent the solution of the PDE at the nodes. Baines and Wathen<sup>26</sup> presented a global explicit formulation of the MFEM, based on the spatial projection of the  $L_2$  norm of the residuals. The key step of the method is to carry out a “mass” matrix inversion through the conjugate gradient method. This procedure includes a first step in which such matrix is conditioned employing algebraic manipulation to explicitly identify and overcome node coalescence and parallelism.

The original space of basis functions employed was considered nonadequate to handle partial differential equations containing diffusion terms, since the approximation of second-order spatial derivatives leads to  $\delta$ -functions, and the integrals generated by the minimization of the residuals square norm in the domain are difficult to calculate (can only be approximated). To widen the complexity of the problems handled and the accuracy, high-order polynomial spaces have been considered. Among others, Hansen and Hassager<sup>27</sup> used quadratic functions, Herbst et al.<sup>28</sup>, Pipilis<sup>29</sup> and Duarte<sup>30</sup> used cubic Hermite polynomials, Sereno et al.<sup>31</sup> and Coimbra et al.<sup>32</sup> used variable order Lagrange polynomials.

### Formulation of Moving Finite Elements Based Algorithm

The algorithm proposed in this section to solve 1-D PBE stands on the moving finite elements method combined with cubic Hermite polynomials to represent local approximations of the solution. The choice of a finite-elements based scheme aims at achieving a flexible framework that can easily cope with any combination of particulate mechanisms, any structure of aggregation kernel or growth rate, and provide the complete PSD in terms of density number, with good resolution.

The cubic Hermite polynomials' space, here represented as  $\mathcal{H}_3$ , is chosen due to its ability to represent the solution and its first order spatial derivative, and the ease of interpretation. Furthermore, this space of functions assures the continuity of both the solution and its first-order spatial derivative at the nodes, thus, allowing no need for additional algebraic equations to represent continuity conditions. This is relevant because for this the differential algebraic equations (DAE) system generated by discretization includes the algebraic equations arisen from boundary condition approximation, thus, leading to better integration properties and requiring



**Figure 1. Domain discretization scheme.**

less computational effort. Each node is associated with two time-dependent coefficients, one representing the solution of the PDE at the node, the other its spatial derivative. The advantages of using this space of functions regarding the interpretability are straightforward.

For the sake of compactness and considering the nomenclature already introduced, the general PBE described by Eqs. 1–3 is redefined as

$$n_t = \mathcal{L}\left(n, n_v, \int_{\Omega} n \, d\Omega, \dots, v, t\right) \quad (4)$$

$$n(0, t) = \mathcal{B}\left(n, n_v, \int_{\Omega'} n \, d\Omega', \dots, v, t\right) \quad (5)$$

$$n(v, 0) = f(v) \quad (6)$$

$$t \in [0, t_f], \quad v \in \Omega \quad (7)$$

where  $n$  is the number of particles,  $n_t$  the time derivatives of  $n$ ,  $\mathcal{L}(\bullet)$  and  $\mathcal{B}(\bullet)$  are operators including nonlinear transformations of  $n$ , spatial derivatives represented by  $n_v$ , and integral terms in domains  $\Omega$  and  $\Omega'$ , with  $\Omega' \subseteq \Omega$ . The residual of the PDE equation, is, therefore, given by:

$$r = n_t - \mathcal{L}\left(n, n_v, \int_{\Omega} n \, d\Omega, \dots, v, t\right) \quad (8)$$

Figure 1 represents the discretization scheme of a general domain  $\Omega$  into  $NE$  finite elements of length  $h_k(t)$ ,  $k \in \{1, \dots, NE\}$ , delimited by nodes located at positions  $s_j(t)$ ,  $j \in \{0, \dots, NE\}$ .

Global approximations of the terms included in operators  $\mathcal{L}(\bullet)$  and  $\mathcal{B}(\bullet)$  are defined as follows:

$$N = \sum_{k=1}^{NE} A_k(t) \phi_k[u, h_k(t)] \quad (9)$$

$$N_t = \sum_{k=1}^{NE} \left\{ \phi_k[u, h_k(t)] \dot{A}_k(t) + A_k(t) \frac{\partial \phi_k[u, h_k(t)]}{\partial u} \frac{\partial u}{\partial h_k(t)} \right. \\ \left. \times \dot{h}_k(t) + A_k(t) \frac{\partial \phi_k[u, h_k(t)]}{\partial h_k(t)} \dot{h}_k(t) \right\} \quad (10)$$

$$N_v = \sum_{k=1}^{NE} \left\{ A_k(t) \frac{\partial \phi_k[u, h_k(t)]}{\partial u} \frac{\partial u}{\partial v} \right. \\ \left. + A_k(t) \frac{\partial \phi_k[u, h_k(t)]}{\partial h_k(t)} \frac{\partial h_k(t)}{\partial v} \right\} \quad (11)$$

$$\int_{\Omega} N \, d\Omega = \int_{\Omega} \left\{ \sum_{k=1}^{NE} A_k(t) \phi_k[u, h_k(t)] \right\} d\Omega \quad (12)$$

where  $N$  is the approximation of  $n$ ,  $N_t$  the approximation of  $n_t$ ,  $N_v$  the approximation of  $n_v$ ,  $A_k(t)$  time dependent coefficients,  $\phi_k[\bullet]$  basis functions dependent on a spatial dimen-

sion normalized with respect to the length of the finite elements,  $h_k(t) = s_k(t) - s_{k-1}(t)$ , which are also time dependent due to node mobility,  $\dot{A}_k(t)$  stands for time derivatives of  $A_k(t)$ ,  $\dot{h}_k(t) = \dot{s}_k(t) - \dot{s}_{k-1}(t)$  for time derivatives of  $h_k(t)$ , and  $u$  is the normalized spatial variable in each of the finite elements,  $u = (v - s_{k-1}(t))/(s_k(t) - s_{k-1}(t))$ ,  $v \in [s_{k-1}(t); s_k(t)]$ . The approximation of residuals becomes

$$R = N_t - \mathcal{L}\left(N, N_v, \int_{\Omega} N \, d\Omega, \dots, v, t\right) \quad (13)$$

with terms defined by Eqs. 9–12.

The finite elements formulation here presented stands on the minimization of the square norm of the residuals with respect to  $\dot{A}_k(t)$ ,  $\forall k$ , and node velocities  $\dot{s}_j(t)$ ,  $\forall j$ . The method falls into the classic formulation of finite-element approaches. The weighting function employed to generate local discretized equations is the derivative of residuals with respect to the time derivative of coefficients and nodal velocities, respectively. The most common schemes are the orthogonal collocation, the method of Galerkin and the method of moments.<sup>33</sup> This weighting function increases the CPU time required, because the integrals in the weighted residuals have to be evaluated at every finite element. Nevertheless, it guarantees the equidistribution of the error at local and global scales, which is appropriate for problems giving way to complex evolutionary dynamics.

As the space  $\mathcal{H}_3$  was chosen to represent local approximations it is now possible to rewrite Eq. 9

$$N = \sum_{k=1}^{NE} \sum_{i=1}^4 a_{k,i}(t) H_i[u, h_k(t)] \quad (14)$$

where  $a_{k,i}(t)$  are time dependent coefficients associated with each of the cubic Hermite polynomials that represent the solution in the element  $k$ , and  $H_i$  is the cubic Hermite polynomial (see the book of Abramowitz and Stegun<sup>34</sup>). The formulas of cubic Hermite polynomials and their derivatives with respect to  $h_k(t)$ , and  $u$  can be found in the works of Pipilis<sup>29</sup> and Duarte.<sup>30</sup> Notice that the time dependent coefficients  $a_{k,i}(t)$  replace the parameters  $A_k(t)$  when the formulation based on general specified functions is applied in the space of orthogonal polynomials  $\mathcal{H}_3$ . Local approximations of the terms  $N_t$ ,  $N_v$  and  $\int_{\Omega} N \, d\Omega$  (Eqs. 10–12) are obtained similarly, thus, converting the global feature of the method to local-coefficient dependent equations. The resulting global equations are

$$\int_{\Omega} R \frac{\partial R}{\partial \dot{a}_{k,i}} \, d\Omega = 0, \quad k \in \{1, \dots, NE\}, i \in \{1, \dots, 4\} \quad (15)$$

$$\int_{\Omega} R \frac{\partial R}{\partial \dot{s}_j} \, d\Omega = 0, \quad j \in \{0, \dots, NE\}, \quad (16)$$

leading to local equations

$$\int_{s_{k-2}}^{s_{k-1}} R_{k-1} \frac{\partial R_{k-1}}{\partial \dot{a}_{k-1,i}} \, dh_{k-1} + \int_{s_{k-1}}^{s_k} R_k \frac{\partial R_k}{\partial \dot{a}_{k,i}} \, dh_k = 0 \\ k \in \{2, \dots, NE - 1\}, i \in \{1, \dots, 4\} \quad (17)$$



$$\int_{s_{k-1}}^{s_k} R_k \frac{\partial R_k}{\partial \dot{a}_{k,i}} dh_k = 0 \quad k \in \{1\}, i \in \{1, \dots, 4\} \quad (18)$$

$$\int_{s_{k-1}}^{s_k} R_k \frac{\partial R_k}{\partial \dot{a}_{k,i}} dh_k = 0 \quad k \in \{NE\}, i \in \{1, \dots, 4\} \quad (19)$$

$$\int_{s_{k-2}}^{s_{k-1}} R_{k-1} \frac{\partial R_{k-1}}{\partial \dot{s}_j} dh_{k-1} + \int_{s_{k-1}}^{s_k} R_k \frac{\partial R_k}{\partial \dot{s}_j} dh_k = 0 \quad k \in \{2, \dots, NE-1\}, j \in \{k-2, k-1, k\} \quad (20)$$

$$\int_{s_{k-1}}^{s_k} R_k \frac{\partial R_k}{\partial \dot{s}_j} dh_k = 0 \quad k \in \{1\}, j \in \{k-1, k\} \quad (21)$$

$$\int_{s_{k-1}}^{s_k} R_k \frac{\partial R_k}{\partial \dot{s}_j} dh_k = 0 \quad k \in \{NE\}, j \in \{k-1, k\} \quad (22)$$

where  $R_k$  is the residual approximation in the  $k$ th finite element.

The original formulations of the moving finite elements method referred that its application is conditioned by two major problems which increase the difficulty of integrating the arising system of ODEs (Eqs. 17–22). The first, called parallelism, is due to the occurrence of solution linearities in neighbor finite elements that lead to linearly dependent equations. The second, designated as node overtaking, causes grid tangling. The strategies proposed to overcome each of these problems depend on the basic formulation of the MFEM used, with the most common strategies being mesh redistribution, addition of a penalty function to the square norm of residuals to be minimized, and node avoidance. The approach adopted here was originally proposed by Miller and Miller<sup>25</sup>—the addition of a penalty function to avoid that internodal viscosity forces lead to nodal coalescence. Several authors proposed different penalty terms, some of them introducing more than one parameter which have to be set (see for instance the work of Gelinis et al.<sup>35</sup>). The choice of a penalty function based on a single parameter allows reducing the dependence of the MFEM on parameters to tune, which would be problem dependent. The internodal viscosity force,  $\varepsilon$ , which can be understood as the degree of freedom given to the nodes, is the only parameter included in the penalty function, and Eq. 16 becomes

$$\int_{\Omega} R \frac{\partial R}{\partial \dot{s}_j} d\Omega + \varepsilon (\dot{s}_j - \dot{s}_{j-1}) = 0, \quad j \in \{0, \dots, NE\} \quad (23)$$

thus, leading to

$$\int_{s_{k-2}}^{s_{k-1}} R_{k-1} \frac{\partial R_{k-1}}{\partial \dot{s}_j} dh_{k-1} + \int_{s_{k-1}}^{s_k} R_k \frac{\partial R_k}{\partial \dot{s}_j} dh_k + \varepsilon (2 \dot{s}_{k-1} - \dot{s}_{k-2} - \dot{s}_k) = 0 \quad k \in \{2, \dots, NE-1\}, j \in \{k-2, k-1, k\} \quad (24)$$

$$\int_{s_{k-1}}^{s_k} R_k \frac{\partial R_k}{\partial \dot{s}_j} dh_k + \varepsilon (\dot{s}_k - \dot{s}_{k-1}) = 0 \quad k \in \{1\}, j \in \{k-1, k\} \quad (25)$$

$$\int_{s_{k-1}}^{s_k} R_k \frac{\partial R_k}{\partial \dot{s}_j} dh_k + \varepsilon (\dot{s}_k - \dot{s}_{k-1}) = 0 \quad k \in \{NE\}, j \in \{k-1, k\} \quad (26)$$

Equations 24–26 replace the relations (20–22). Eqs. (17–19) and (24–26) together with the relations obtained by discretizing the boundary condition  $\mathcal{B}(\bullet)$ , and the equations that describe the movement of the nodes at the boundaries of domain  $\Omega$  form the DAE system obtained by PDE discretization. In problems with fixed boundaries the nodes associated with domain extremes are not allowed to move, therefore

$$\dot{s}_0(t) = 0 \quad (27)$$

$$\dot{s}_{NE}(t) = 0 \quad (28)$$

The formulation presented in this section is written in terms of generic operators and functional forms aiming to reduce the complexity. In the Appendix the closed-form coded DAEs for Case 1 are presented to enlighten the methodology application.

The algebraic equations included in the DAE system result from the discretization of the boundary conditions. The complete DAE system, of 3 ( $NE+1$ ) equations, falls into the general form

$$\Phi(y, \dot{y}, v, t) = 0 \quad (29)$$

where  $\Phi$  is a vector of nonlinear functions, and  $y = [\{a_{k,i}(t), k \in \{1, \dots, NE\}, i \in \{1, \dots, 4\}\}, \{s_j(t), j \in \{0, \dots, NE\}\}]^T$ . The cubic Hermite polynomials allows one to take advantage of the equality of the time dependent coefficients at the nodes

$$a_{k-1,3}(t) = a_{k,1}(t) \quad (30)$$

$$a_{k-1,4}(t) = a_{k,2}(t) \quad (31)$$

resulting in three variables per node ( $a_{k,1}(t), \forall k, a_{k,2}(t), \forall k$ , and  $s_j(t), \forall j$ ), with  $a_{k,1}(t)$  as the solutions' approximation at the nodes,  $N(v,t)$ , and  $a_{k,2}(t)$  its first-order spatial derivative,  $N_v(v,t)$ . The initialization of the coefficients  $a_{k,1}(t)$  is based on the initial PSD, represented as  $f(v)$ . By their turn, the initialization of  $a_{k,2}(t)$  is based on the spatial derivative of  $f(v)$ , represented as  $df(v)/dv$ , evaluated numerically employing the perturbation method.

The grid is initialized according to two distinct strategies:

- The initial assignment of the nodes' positions taking into account the combination of the phenomena involved and the type of dynamics forecasted;

- The calculation of a geometric grid by employing the algorithm owed to Nicmanis and Hounslow.<sup>16</sup> This requires choosing the length of the first element and of the spatial domain considered.

The first methodology takes advantage of the moving character of the grid that conforms to the solution in a few time steps. It allows using smaller values of the internodal viscosity force parameter, increasing the freedom of the nodes to move in pursuit of the fronts in the later stages. However, it depends on the problem structure and the information on the dynamic features of the PBE being handled.

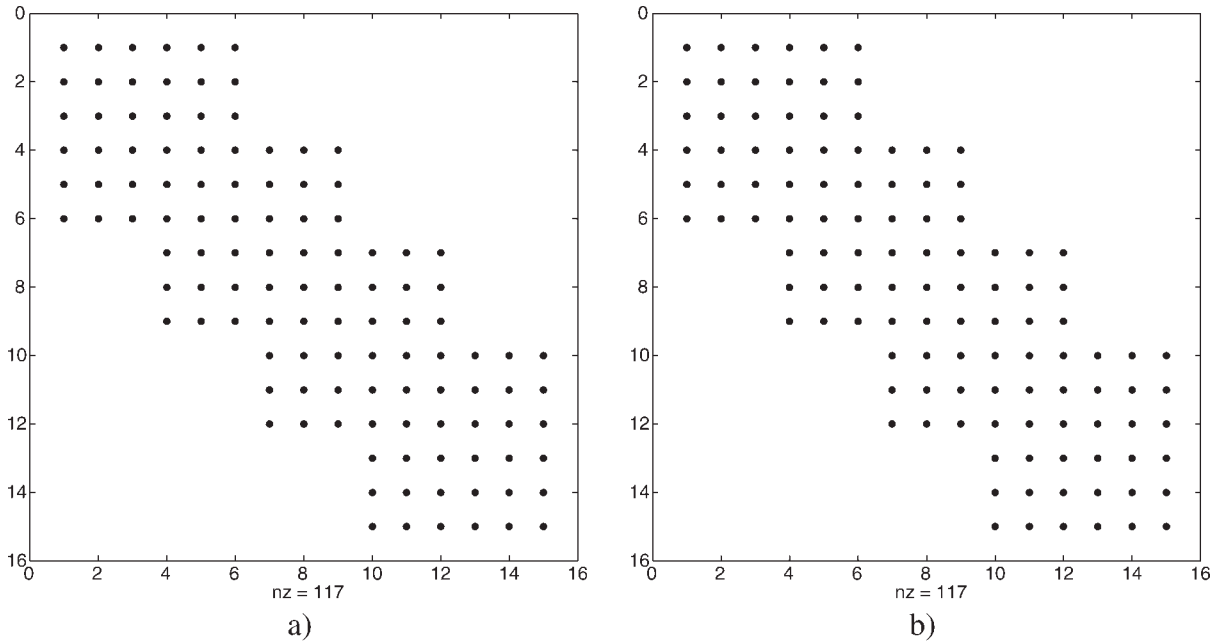


Figure 2. Jacobian structure for pure growth PBE ( $NE = 5$ ).

(a) Differential; and (b) Algebraic.

The initial values of  $y$  provided to the solver are

$$\begin{aligned}
 s_j(0) &= \mathcal{I}(j, NE) \\
 a_{k,1}(0) &= f(s_{k-1}(0)) \quad k \in \{1, \dots, NE - 1\} \\
 a_{k,3}(0) &= f(s_k(0)) \quad k \in \{NE\} \\
 a_{k,2}(0) &= \frac{df(s_{k-1}(0))}{dv} \quad k \in \{1, \dots, NE - 1\} \\
 a_{k,4}(0) &= \frac{df(s_k(0))}{dv} \quad k \in \{NE\}
 \end{aligned} \quad (32)$$

where  $\mathcal{I}(\bullet)$  is the initial grid distribution, and  $df(s_k(0))/dv$  is the spatial derivative of  $f(v)$  at  $v = s_k(0)$ .

The solver DASOLV,<sup>36</sup> which is based on backward difference formulas of variable order is used. It enables exploiting the system sparsity in order to reduce the computational effort, and includes a procedure to find a consistent initial solution. The absolute and relative tolerance values were set to  $10^{-6}$ .

The integrals in every differential-algebraic equation are evaluated by means of Gaussian Quadrature based on eight collocation points in each finite element, located on the zeros of seventh order Jacoby polynomials. Two of them coincide with the extremes of the finite element. The same approach is used to calculate the integrals describing the breakage and aggregation mechanisms.

### Numerical tests

The moving finite elements method is applied to simulate PBE describing particulate processes including several

combinations of nucleation, growth, and aggregation phenomena. In order to evaluate the performance of the method and its ability in dealing with each mechanism the tests were carried out using models with a known analytical solution. The following list of combinations of phenomena was considered:

- (a) Pure growth;
- (b) Simultaneous growth and nucleation;
- (c) Simultaneous growth and aggregation.

### PBE simulating pure growth

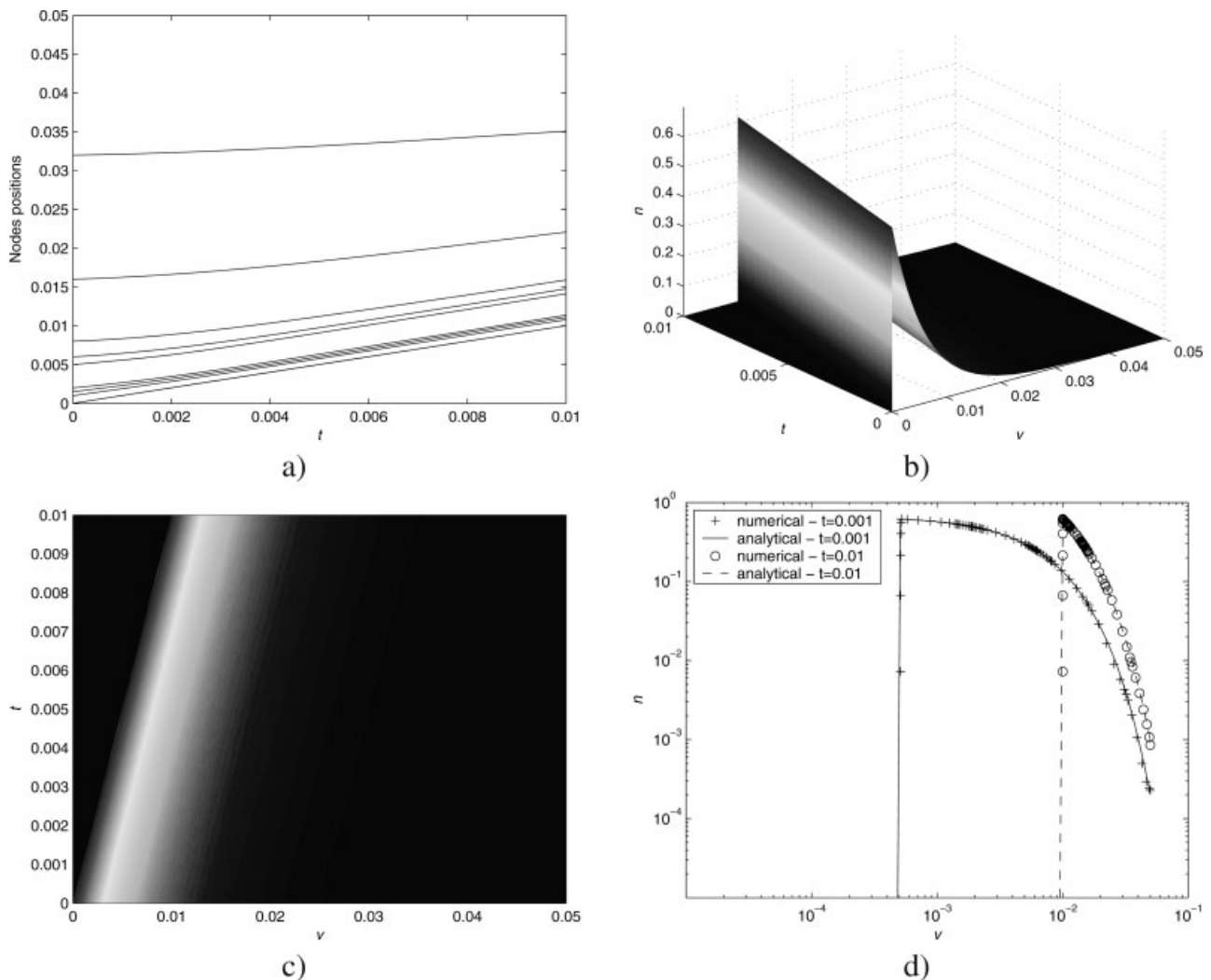
The PBE used to test the MFEM in dealing with pure growth mechanisms is described by the following equations

$$\frac{\partial n(v, t)}{\partial t} = - \frac{\partial [G(v) n(v, t)]}{\partial v} \quad (33)$$

$$n(0, t) = 0 \quad (34)$$

$$n(v, 0) = \begin{cases} 0 & \text{if } v < 10^{-5} \\ \frac{\chi_{0,i}}{v_{0,i}} \exp\left(-\frac{v}{v_{0,i}}\right) & \text{if } v \geq 10^{-5} \end{cases} \quad (35)$$

The parameters involved in the initial distribution,  $\chi_{0,i}$  and  $v_{0,i}$ , were set to 0.00384 and 0.0062, respectively. It is assumed that the minimum characteristic dimension of the particles is  $10^{-5}$ . The differential and algebraic Jacobian matrices of the ODE system, arisen from discretization, have the same structure, which is band-diagonal with five non-zeros to the left of the diagonal, and five immediately to its right (see Figure 2).



**Figure 3. Solution of the PBE simulating growth at constant growth rate.**

(a) Nodes movement; (b) 3-D solution; (c) Solution surface plot; and (d) Numerical vs. analytical solutions.

**Case 1** –  $G(v) = 1$ ,  $t \in [0,0.01]$ ,  $v \in [0, 0.05]$

This case represents a constant growth rate and, according to Kumar and Ramkrishna,<sup>13</sup> the analytical solution is

$$n(v, t) = \begin{cases} 0 & \text{if } v < t \\ \frac{\lambda_{0,i}}{v_{0,i}} \exp\left(-\frac{v-t}{v_{0,i}}\right) & \text{if } v \geq t \end{cases} \quad (36)$$

This problem gives rise to an evolutionary front with a velocity equal to 1 unit of characteristic dimension/unit of time associated to the discontinuity of the initial distribution. It is a challenging problem for PDE solvers because of the numerical instability it originates near the shock.

Figure 3a shows the movement of the nodes which conforms to the movement of the front and fits the discontinuity in Figure 3b. This figure represents the 3-D solution, and Figure 3c illustrates the contour surface that highlights the movement of the front corresponding to the growth of particles, perfectly captured by MFEM. Along the time interval the PSD moves toward the larger particles zone, and its distribution gets narrower, as shown in Figure 3d, where the

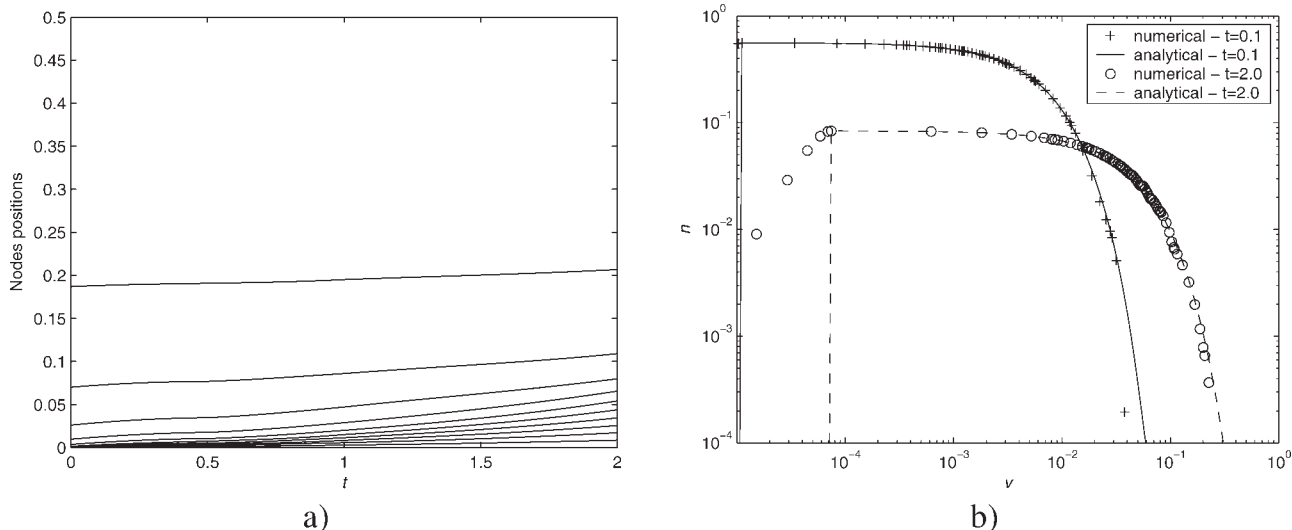
good agreement between analytical and numerical solutions is also evident. The ability of MFEM in capturing the front location without numerical diffusion is due to the placement of two nodes immediately before and ahead of the discontinuity, in order to handle the large magnitude of the spatial derivative, and follow the residuals maximal gradient.

**Case 2** –  $G(v) = v$ ,  $t \in [0,2.0]$ ,  $v \in [0, 0.5]$

The growth rate of a particulate growth process,  $G(v)$  is now considered as linear. In their work Kumar and Ramkrishna<sup>13</sup> also give the analytical solution for the linear growth rate case which is

$$n(v, t) = \frac{\lambda_{0,i}}{v_{0,i}} \exp\left(-\frac{v \exp(-t)}{v_{0,i}}\right) \exp(-t) \quad (37)$$

As  $G(v)$  is not constant, it is expected to obtain a broader PSD than in the previous case and this is shown in Figure 4b. As a consequence the discontinuities are attenuated, thus, producing lower residuals dispersed all over the spatial domain. This phenomenon brings the nodes, initially placed in the region of smaller particles, to move along following the front, as can be seen in Figure 4a. Figure 4b



**Figure 4. Solution of the PBE simulating growth at linear growth rate.**

(a) Nodes movement; and (b) Numerical vs. analytical solutions.

also highlights the agreement between numerical and analytical solutions.

Although similar, the two models studied (Cases 1 and 2) give origin to quite different dynamics. The first leads to PSD narrowing, in the second the distribution gets broader and this is followed by a reduction in particles density, comparing to initial distribution. Table 1 lists computational parameters used to simulate Cases 1 and 2, emphasizing the low computational effort required to get very accurate results.

In these two case studies the MFEM proved to be able to handle growth mechanisms with different dynamics, and, therefore, its adequacy to solve PBE involving pure growth can be generalized.

**Table 1. Parameters Employed in the Simulation of Pure Growth PBE**

Case	Nodes	Grid Initialization	$\varepsilon$	CPU Time <sup>†</sup>
1	12	$s = [0.0, 9.9 \times 10^{-6},$ $2.0 \times 10^{-5},$ $1.0 \times 10^{-3},$ $1.5 \times 10^{-3},$ $2.0 \times 10^{-3},$ $5.0 \times 10^{-3},$ $6.0 \times 10^{-3},$ $8.0 \times 10^{-3},$ $1.6 \times 10^{-2},$ $3.2 \times 10^{-2},$ $5.0 \times 10^{-2}]^T$	0.2	1.000
2	13	$s = [0.0, 1.0 \times 10^{-5},$ $1.0 \times 10^{-4},$ $1.0 \times 10^{-3},$ $1.5 \times 10^{-3},$ $2.0 \times 10^{-3},$ $5.0 \times 10^{-3},$ $6.0 \times 10^{-3},$ $8.0 \times 10^{-3},$ $1.6 \times 10^{-2},$ $3.2 \times 10^{-2},$ $0.1, 0.5]^T$	0.002	2.812

<sup>†</sup>On a Windows Pentium 2.80 GHz computer.

### PBE simulating growth and nucleation

The PBE representing the combination of particles growth and nucleation are

$$\frac{\partial n(v,t)}{\partial t} = -\frac{\partial[G(v)n(v,t)]}{\partial v} + S(v) \quad (38)$$

$$n(0,t) = 0 \quad (39)$$

$$n(v,0) = \begin{cases} 0 & \text{if } v < 10^{-5} \\ \frac{\chi_{0,i}}{v_{0,i}} \exp\left(-\frac{v}{v_{0,i}}\right) & \text{if } v \geq 10^{-5} \end{cases} \quad (40)$$

with  $S(v)$  standing for the nucleation rate. The initial and boundary conditions used in Cases 1 and 2 will be used in this subsection, in order to enable comparing and analyzing the complexity introduced by the nucleation term. The structure of both jacobian matrices originated by this kind of problems is similar to those presented in Figure 2. The nucleation rate expression employed is

$$S(v) = \frac{\chi_{0,n}}{v_{0,n}} \exp\left(-\frac{v}{v_{0,n}}\right) \quad (41)$$

with  $\chi_{0,n} = 5 \times 10^{-3}$  and  $v_{0,n} = 0.001$ .

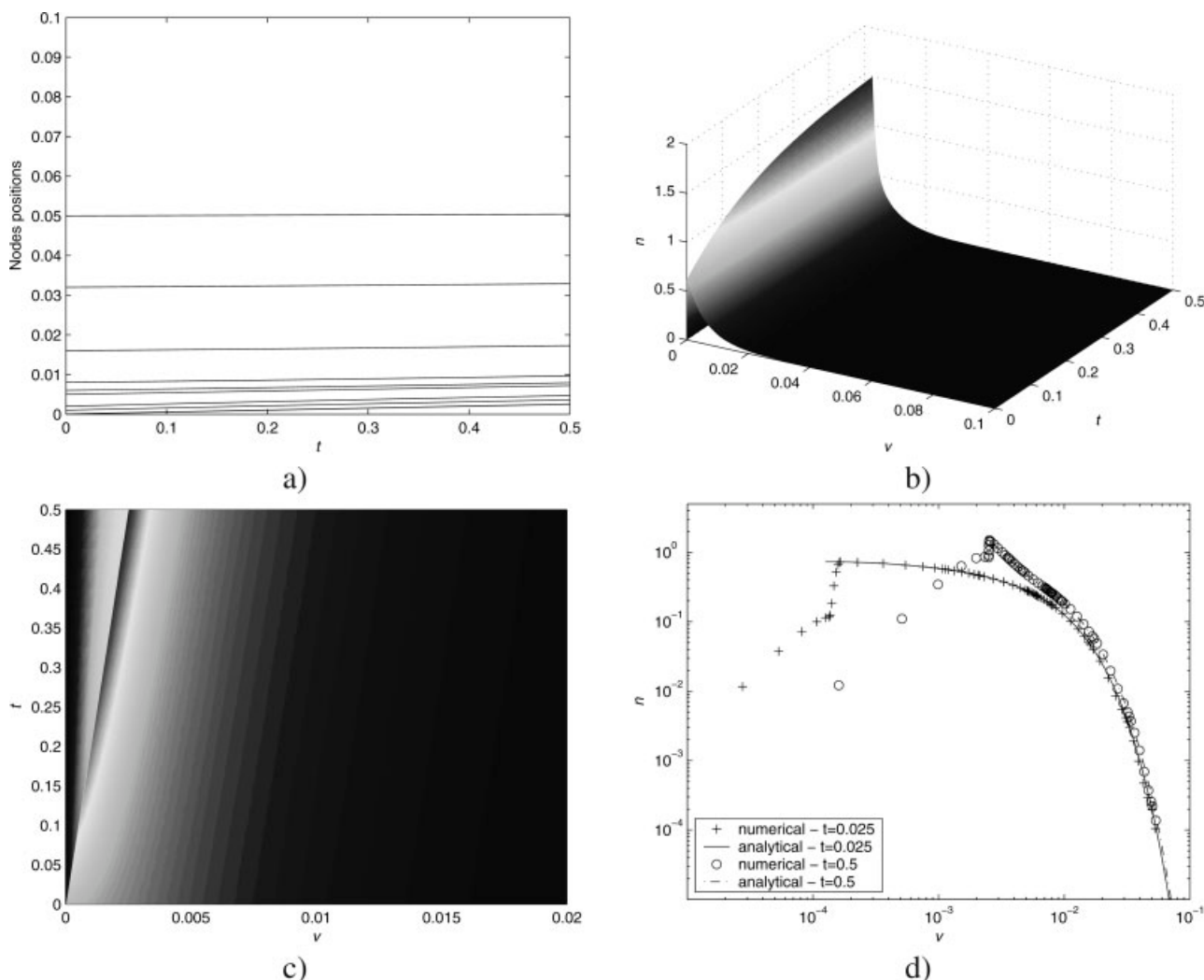
**Case 3** –  $G(v) = 0.005$ ,  $t \in [0,0.5]$ ,  $v \in [0,0.1]$

The growth rate, which remains constant, was set to a lower value than in previous case studies in order to simulate a situation where growth and nucleation phenomena have similar contributions to the system dynamics. The analytical solution for this problem is given by Kumar and Ramkrishna<sup>13</sup>

$$n(v,t) = \frac{\chi_{0,i}}{v_{0,i}} \exp\left(-\frac{v-0.005t}{v_{0,i}}\right) + \frac{\chi_{0,n}}{0.005} \left[ \exp\left(-\frac{v_l}{v_{0,n}}\right) - \exp\left(-\frac{v}{v_{0,n}}\right) \right] \quad (42)$$

$v_l = \max(1 \times 10^{-5}, v - 0.005t)$





**Figure 5. Solution of the PBE simulating nucleation and growth at constant growth rate.**

(a) Nodes movement; (b) 3-D solution; (c) Solution surface plot; and (d) numerical vs. analytical solutions.

The displacement of the nodes as they follow the growth front is illustrated in Figure 5a. In agreement with the lower growth rate the front moves slower than in Case 1. Due to nucleation, the density of particles in the front increases with time, and the discontinuity becomes sharper causing the PSD to narrow. Figure 5b shows the increase in the density number of particles at the front during the time this process was simulated, and Figure 5c emphasizes the smooth movement of the front due to both the lower growth rate and the increase in the density of particles.

Moreover, Figure 5c shows that the particles with characteristic dimension smaller than those in the front are continuously generated by nucleation, and its density number is not null, as was observed in pure growth cases. Moreover, once particles are generated by nucleation, the growth mechanism provokes its growth, thus, leading to an increase in density number for characteristic dimension values lower than front position. This behavior was also reported by Kumar and Ramkrishna,<sup>13</sup> and can also be seen in Figure 5d where, at any instant, the growth phenomenon gives origin to a front that moves toward larger characteristic dimension values.

The good performance of the numerical technique in this example is evident in this last figure.

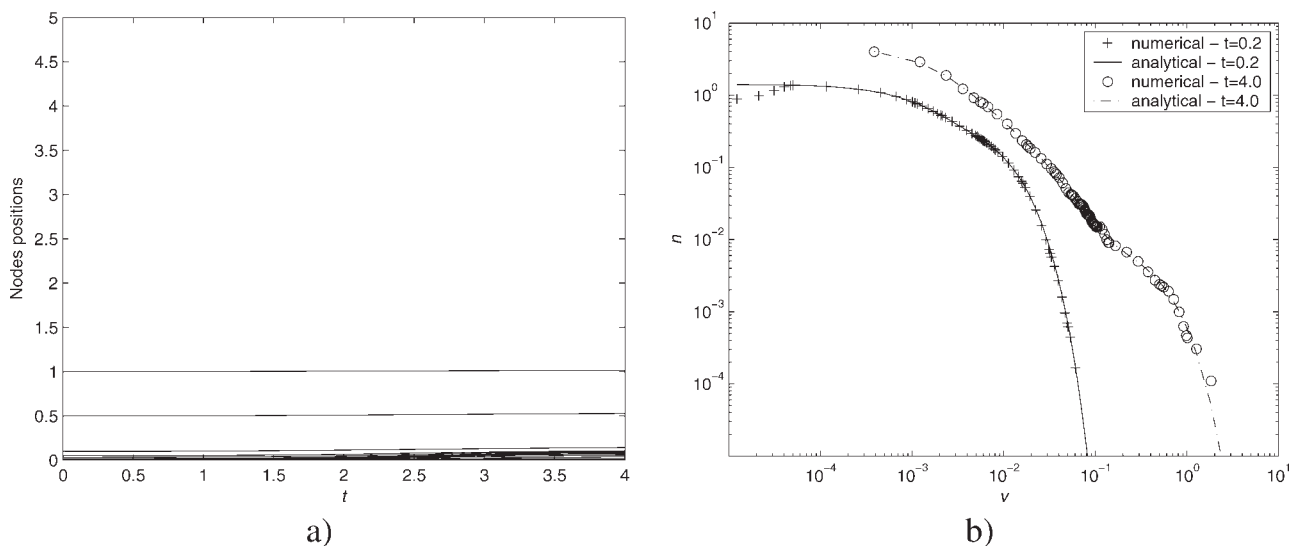
**Case 4** –  $G(v) = v$ ,  $t \in [0,4.0]$ ,  $v \in [0,5.0]$

In this case study a linear growth rate and an exponentially based nucleation rate are considered, and the analytical solution is given by Kumar and Ramkrishna<sup>13</sup>

$$n(v, t) = \frac{\chi_{0,i}}{v_{0,i}} \exp\left(-\frac{v \exp(-t)}{v_{0,i}}\right) \exp(-t) + \frac{\chi_{0,n}}{v} \left[ \exp\left(-\frac{v_l}{v_{0,n}}\right) - \exp\left(-\frac{v}{v_{0,n}}\right) \right] \quad (43)$$

$$v_l = \max(1 \times 10^{-5}, v \exp(-t))$$

The results obtained in Case 2 allow to expect that this solution will also have a broader distribution due to the growth dynamics, and this is confirmed in Figure 6b. Each phenomenon dominates in a different zone, the nucleation for smaller particles, and the growth for larger (see Figure 6b). In this figure, this behavior is very clear, particularly for  $t = 4.0$ , and one can easily locate the characteristic dimension for



**Figure 6. Solution of the PBE simulating nucleation and growth at linear growth rate.**

(a) Nodes movement; (b) Numerical vs. analytical solutions.

which both phenomena are equally important, and, therefore, profiles associated to each phenomenon meet. Figure 6a illustrates the smooth motion of the nodes to adapt to the slowness of the dynamics developed. This motion will be remarkably faster if the contribution of the growth term is larger than nucleation. Table 2 summarizes the runs carried out to simulate the PBE described by Cases 3 and 4.

### PBE simulating growth and aggregation

The general PBE model used to test the MFEM performance when growth and aggregation mechanisms are present is

$$\frac{\partial n(v, t)}{\partial t} = -\frac{\partial[G(v) n(v, t)]}{\partial v} + \frac{1}{2} \int_0^v n(v-v', t) n(v', t) q(v-v', v') dv' - n(v, t) \int_0^\infty n(v', t) q(v, v') dv' \quad (44)$$

$$n(0, t) = 0 \quad (45)$$

$$n(v, 0) = \frac{\chi_{0,i}}{v_{0,i}} \exp\left(-\frac{v}{v_{0,i}}\right) \quad (46)$$

with  $q(v, v')$  standing for the rate of aggregation of particles of characteristic dimension  $v - v'$  with particles of  $v'$  to produce particles of characteristic dimension  $v$ . The values used for the parameters  $\chi_{0,i}$  and  $v_{0,i}$  are those used when the combination of growth and nucleation phenomena was studied. The second integral is evaluated at each finite element employing Gaussian quadrature

$$\int_0^\infty n(v', t) q(v, v') dv' = \sum_{k=1}^{NE} h_k(t) \sum_{m=1}^{NC} N(u_m, t) q(v, v_m) w_m \quad (47)$$

$$v_m = s_{k-1}(t) + h_k(t) u_m$$

where  $w_m$  is the Gaussian quadrature weight,  $NC$  the number of collocation points, and  $u_m$  the roots of the collocation

points corresponding to the characteristic dimension  $v_m$  within the finite element  $k$ . This approximation is subject to finite domain error (FDE), due to discarding the portion of the integral outside the domain of discretization. Nevertheless, this error is kept low by considering large  $\Omega$  domains in order to capture most of the aggregation phenomena.

The evaluation of the integral

$$z(v, t) = \int_0^v n(v-v', t) n(v', t) q(v-v', v') dv' \quad (48)$$

involves additional complexity, particularly when adaptive grid methods are chosen. Two major difficulties arise, the first related to the evaluation of the approximation of  $n(v'', t)$ , where  $v'' = v - v'$ , the second associated to the need to compute the integral  $z(v, t)$  when  $v$  does not coincide with a node. The latest occurs when  $v$  coincides with an internal collocation point employed for integral calculation.

**Table 2. Parameters Employed in the Simulation of Growth plus Nucleation PBE**

Case	Nodes	Grid Initialization	$\varepsilon$	CPU Time <sup>†</sup>
3	12	$s = [0.0, 9.9 \times 10^{-6}, 1.0 \times 10^{-3}, 5.0 \times 10^{-3}, 6.0 \times 10^{-3}, 8.0 \times 10^{-3}, 0.016, 0.032, 0.05, 0.1, 0.5, 1.0]^T$	0.002	3.453
4	15	$s = [0.0, 9.9 \times 10^{-6}, 4.0 \times 10^{-5}, 1.0 \times 10^{-3}, 2.0 \times 10^{-3}, 5.0 \times 10^{-3}, 6.0 \times 10^{-3}, 8.0 \times 10^{-3}, 0.016, 0.032, 0.05, 0.1, 0.5, 1.0, 5.0]^T$	0.08	4.234

<sup>†</sup>On a Windows Pentium 2.80 GHz computer.

The first problem is handled with a three step algorithm, here designated as Algorithm 1.

*Algorithm 1*

- (1) Location of the finite element, denoted as  $\bar{k}$ , that contains  $v''$ , satisfying the condition  $s_{\bar{k}-1}(t) \leq v'' \leq s_{\bar{k}}(t)$ ;
- (2) Evaluation of the normalized spatial variable corresponding to  $v''$ , designated as  $u''$ , by using the equation

$$u'' = \frac{v'' - s_{\bar{k}-1}(t)}{s_{\bar{k}}(t) - s_{\bar{k}-1}(t)} \quad (49)$$

- (3) Evaluation of the approximation of the solution at  $v''$

$$N(v'', t) = \sum_{i=1}^4 a_{\bar{k},i}(t) H_i(u'', h_{\bar{k}}(t)) \quad (50)$$

The location step involved in this procedure is rather demanding in terms of computational effort.

The residuals' integrals involved in the MFEM fundamental equations (Eqs. 17–19 and 24–26) require the evaluation of  $z(v, t)$  at different values of  $v$ , which either coincide with the nodes (when the collocation points are at the extremes of the finite elements), or are located inside the finite elements. When  $v$  coincides with a node  $K$  of the grid, the integral  $z(v, t)$  is approximated by  $Z(v, t)$

$$Z(v, t) = \sum_{k=1}^K h_k(t) \sum_{m=1}^{NC} N(v''_m, t) N(v_m, t) q(v''_m, v_m) w_m$$

$$v_m = s_{k-1}(t) + h_k(t) u_m; v''_m = s_{\bar{k}-1}(t) + h_{\bar{k}}(t) u''_m \quad (51)$$

employing Algorithm 1 to calculate the terms  $N(v''_m, t)$  and  $q(v''_m, v_m)$ . Whenever  $v \neq s_j, \forall j$ , the procedure for its evaluation requires its partition into two terms

$$z(v, t) = \int_0^{s_{K-1}} n(v - v', t) n(v', t) q(v - v', v') dv' + \int_{s_{K-1}}^v n(v - v', t) n(v', t) q(v - v', v') dv' \quad (52)$$

here  $s_{K-1}(t)$  stands for the node immediately to the left of  $v$ . The first integral is calculated by Eq. 51. The second is evaluated following the Algorithm 2 described as follows.

*Algorithm 2*

- (1) Location of the finite element, generically designated by  $K$ , that contains  $v$ , satisfying the condition  $s_{K-1}(t) \leq v \leq s_K(t)$ ;
- (2) Determination of the corresponding normalized spatial variable

$$u' = \frac{v - s_{K-1}(t)}{s_K(t) - s_{K-1}(t)}; \quad (53)$$

- (3) Evaluation of  $N(v, t)$  and  $N_v(v, t)$  at  $v$

$$N(v, t) = \sum_{i=1}^4 a_{K,i}(t) H_i(u', h_K(t)) \quad (54)$$

$$N_v(v, t) = \sum_{i=1}^4 \frac{a_{K,i}(t)}{h_K(t)} \frac{dH_i(u', h_K(t))}{du} \quad (55)$$

- (4) Construction of a “pseudo-finite” element, designated as  $K^*$ , standing for the portion of domain  $[s_{K-1}(t); v]$ . The

time dependent coefficients associated to cubic Hermite polynomials for finite element  $K^*$  are set by using the equality relations

$$a_{K^*,1}(t) = a_{K+1,1}(t) \quad a_{K^*,2}(t) = a_{K+1,2}(t)$$

$$a_{K^*,3}(t) = N(v, t) \quad a_{K^*,4}(t) = N_v(v, t)$$

- (5) Location of the finite element  $\bar{k}$ , that contains  $v'' = v - v'$ , satisfying the condition  $s_{\bar{k}-1}(t) \leq v'' \leq s_{\bar{k}}(t)$ ;
- (6) Evaluation of the normalized spatial variable corresponding to  $v''$ , designated as  $u''$ , by the relation

$$u'' = \frac{v'' - s_{\bar{k}-1}(t)}{s_{\bar{k}}(t) - s_{\bar{k}-1}(t)} \quad (56)$$

- (7) Determination of the approximation of the solution at  $v''$ .
- (8) Computation of the integral term in the “pseudo-finite” element  $K^*$  through Gaussian quadrature

$$\int_{s_K}^v n(v - v', t) n(v', t) q(v - v', v') dv' = (v - s_{K-1}(t)) \times \sum_{m=1}^{NC} [N(v''_m, t) N(v_m, t) q(v''_m, v_m)] w_m$$

$$v_m = s_{K-1}(t) + (v - s_{K-1}(t)) u_m$$

$$v''_m = s_{\bar{k}-1}(t) + h_{\bar{k}}(t) u''_m \quad (57)$$

This procedure emphasizes the advantages of cubic Hermite polynomials regarding interpretability, since they allow one to easily build local approximations of the solution and its spatial derivative, and to use those values as time dependent coefficients of any finite element delimited by such a point.

The differential Jacobian matrix of PBE problems including aggregation have a similar structure to that obtained for growth and nucleation phenomena cases (Figure 2a). Due to the integral terms involved, the residual at each node depends on the solution all over the domain considered, thus, leading to full algebraic Jacobian matrices. This, together with the need to evaluate the integral terms involved in the PBE, and the location procedure previously discussed (Algorithm 1), increases the computational effort involved. Moreover, the aggregation phenomenon gives origin to slower dynamics than nucleation and growth mechanisms, since the integral terms tend to stabilize the dynamics. A PBE problem involving a fast dynamics phenomenon coupled with aggregation presents two distinct behavior features. At the beginning the “fast” phenomenon prevails, while the later stages of the process are dominated by the aggregation.<sup>37</sup>

**Case 5** –  $G(v) = v$ ,  $q(v', v) = v' + v$ ,  $t \in [0, 2.0]$ ,  $v \in [0, 10.0]$

In this case study the growth rate was again considered as linear and an additive aggregation kernel rate was used. The analytical solution derived by Ramabhadran et al.<sup>38</sup> is

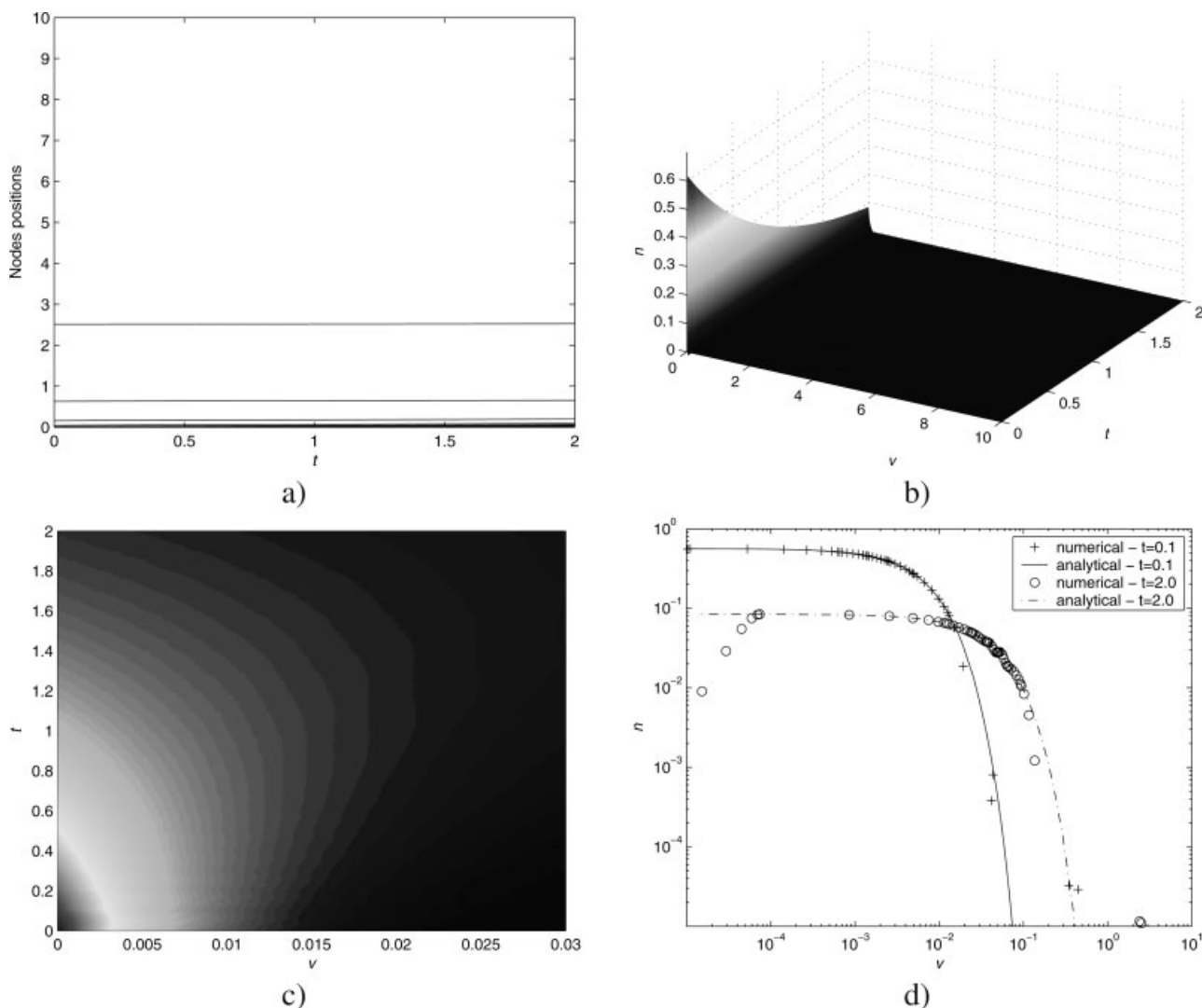
$$n(v, t) = \frac{M_0^2}{M_1} \exp\left[-\frac{M_0}{M_1} \left(\frac{2\chi_{0,i}}{M_0} - 1\right) v\right] \times I_1\left(2 \sqrt{1 - \frac{M_0}{\chi_{0,i}}} \frac{\chi_{0,i}}{M_1} v\right)$$

$$\frac{M_0}{M_1} v \sqrt{1 - \frac{M_0}{\chi_{0,i}}}$$

$$M_0 = \chi_{0,i} \exp[\chi_{0,i} v_{0,i} (1 - \exp(t))]$$

$$M_1 = \chi_{0,i} v_{0,i} \exp(-t) \quad (58)$$

where  $I_1(\bullet)$  is the modified Bessel function of the first kind and first-order. Figure 7a highlights the smoothness of the



**Figure 7. Solution of the PBE simulating growth at constant growth rate combined with an additive aggregation kernel.**  
 (a) Nodes movement; (b) 3-D solution; (c) solution surface plot; and (d) numerical vs. analytical solutions.

movement of the nodes, particularly those located in the zones with larger particles, due to the low residuals developed by the PBE, and the attenuation in particles density shown in Figure 7b. The front associated to particles growth moves toward the region of larger particles (see Figure 7d), where it can also be confirmed that the particles aggregation causes the PSD to broaden. Figure 7b presents the 3-D solution and Figure 7c the contour surface plot of the solution. Figure 7d shows the agreement between analytical and numerical solutions, with a lower performance detected on the far rightside, and associated to larger particles. This inaccuracy can be partly due to FDE which occurs when the particles originally located in the larger size finite elements aggregate and give origin to particles with a characteristic dimension not considered in the range of the integrals (Eq. 47).

**Case 6** –  $G(v) = v$ ,  $q(v',v) = 1$ ,  $t \in [0,2.0]$ ,  $v \in [0,5.0]$

This case involves a linear growth rate combined with a constant aggregation kernel rate. The analytical solution derived by Ramabhadran et al.<sup>38</sup> is

$$n(v,t) = \frac{M_0^2}{M_1} \exp\left(-\frac{M_0}{M_1} v\right)$$

$$M_0 = \frac{2 \chi_{0,i}}{2 + \chi_{0,i} t} \tag{59}$$

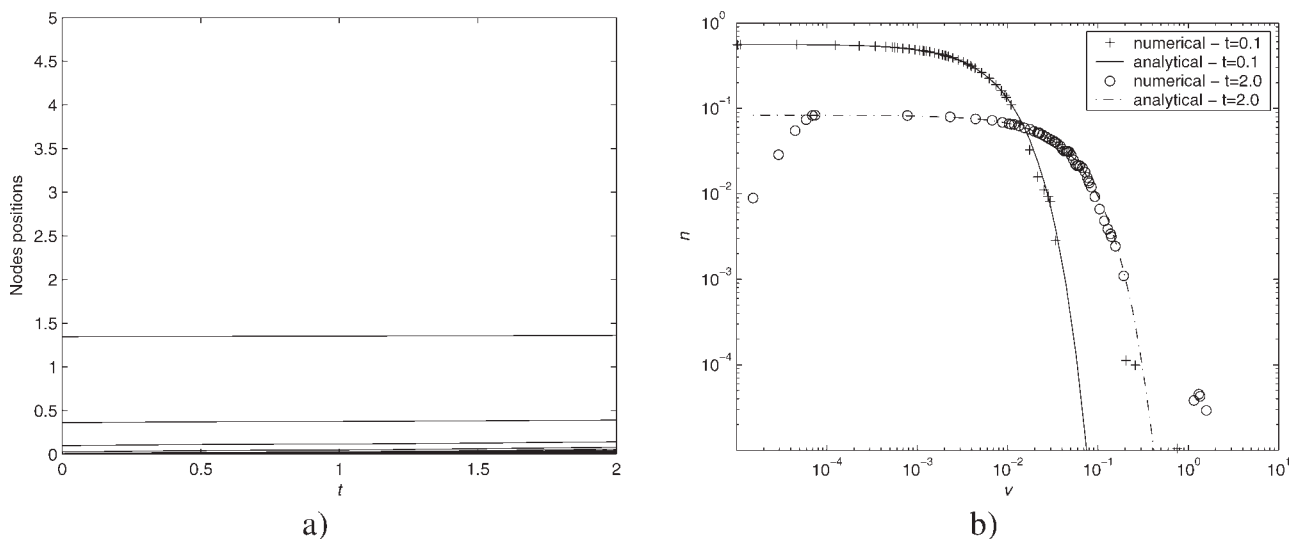
$$M_1 = \chi_{0,i} v_{0,i} \exp(-t)$$

Figure 8a shows the nodes displacement corresponding to the narrowing of the PSD. The numerical solution is also in good agreement with the analytical one except in the zone of larger particles, where differences can be seen (Figure 8b).

Table 3 summarizes the simulation conditions employed in Cases 5 and 6. The CPU time is a few orders of magnitude larger than in the previous phenomena combination studied. Nevertheless, this was expected because an iterative procedure is required to locate the finite element that contains  $v$ , and now the algebraic Jacobian matrix is full.

#### Simulation of a particulate process unit dynamics

Having tested the performance of MFEM for solving PBE models that include several combinations of mechanisms, in



**Figure 8. Solution of the PBE simulating growth at linear growth rate combined with a constant aggregation kernel.**

(a) Nodes movement; (b) Comparison of numerical vs. analytical solutions.

most cases with accurate results, this section is devoted to applying MFEM to simulate the dynamics of a unit in which a particulate process occurs. The models for these units involve a PBE, to represent the dispersed phase behavior, combined with a few ODEs which describe the dynamics of the continuous phase. The case used for testing was presented by Lee et al.<sup>17</sup> to show the adequacy of a MPC based algorithm in controlling a particulate process. It describes the dynamics of a semibatch precipitation reactor where the reaction  $\text{Ca}(\text{OH})_2 + \text{Na}_2\text{CO}_3 \rightarrow \text{CaCO}_3 + 2\text{NaOH}$  takes place. The authors solved it employing a finite elements based scheme, but implemented a different technique standing on the method of characteristics for supporting the control algorithm. Such a technique was not able to cope with the dynamics accounting for breakage and aggregation phenomena, but was computationally more efficient than finite elements based scheme.

The reactor is fed with solutions of calcium carbonate and sodium hydroxide, and precipitation occurs when the concentrations of calcium and carbonate ions are above the supersaturated conditions. Supersaturation requires exceeding the solubility of both ionic species. The unit model was built assuming that the ionization reactions are fast, compared to the precipitation, and perfect mixing is reached. Additionally, the aggregation and breakage phenomena are neglected due to the flow conditions. However, to extend the formulation of the MFEM presented to problems comprising dispersed and continuous phases with the first involving the aggregation of particles, the PBE presented by Lee et al.<sup>17</sup> is reformulated to include such a phenomenon (Case 8).

**Case 7 – Calcium carbonate semibatch reactor without aggregation**

According to Lee et al. the normalized model representing the unit dynamics is<sup>17</sup>

$$\frac{\partial n(v,t)}{\partial t} + \frac{n(v,t)}{V} \frac{dV}{dt} = - \frac{\partial [G(v,t) n(v,t)]}{\partial v} \quad (60)$$

$$\frac{d(V C_n)}{dt} = q_n^f C_n^f - k_a V \int_0^\infty G(v,t) n(v,t) v^2 dv, \quad n=1,2 \quad (61)$$

$$\frac{dV}{dt} = q_1^f + q_2^f \quad (62)$$

$$n(0,t) = \frac{1}{G(0,t)} \int_0^\infty a_n C_s^{b_n} n(v,t) v^{2.5} dv \quad (63)$$

$$G(v,t) = \frac{a_s C_s^{b_t}}{1 + \exp(-a_L (v - b_L))} \quad (64)$$

$$n(v,0) = \frac{0.01}{2} v^2 \exp(-v) \quad (65)$$

$$C_s = \sqrt{C_1 C_2} - 1 \quad (66)$$

$$t \in [0, 600.0], \quad v \in [0, 20.0]$$

In this example the characteristic dimension  $v$  represents the length of the particles. The normalized values of the parameters and variables used in the model are listed in Table 4.

The model for the dispersed phase accounts for growth and nucleation phenomena, the last are only accounted for in the boundary condition (BC), and this explains why Eq. 63 is used. Particles of all sizes contribute to nucleation, and the BC accounts for the ratio between the nucleation and the growth rates for the null size particles, as was established by

**Table 3. Parameters Employed in the Simulation of Growth plus Aggregation PBE**

Case	Nodes	Grid Initialization	$\varepsilon$	CPU Time <sup>†</sup>
5	12	Geometrical, $h_1(0) = 1.0 \times 10^{-5}$	0.004	913.902
6	12	Geometrical, $h_1(0) = 1.0 \times 10^{-5}$	0.003	1317.178

<sup>†</sup>On a Windows Pentium 2.80 GHz computer.



**Table 4. Parameters Used by the Model Representing the Semibatch Precipitation Reactor Processing CaCO<sub>3</sub>**

Description	Designation	Value
Inlet concentrations	$C_n^f, n = 1, 2$	6.0, 6.0
Inlet flow rates	$q_n^f, n = 1, 2$	0.3, 0.3
Initial concentrations	$C_n(0), n = 1, 2$	5.0, 5.0
Initial solution volume	$V(0)$	10.0
Nucleation rate parameters	$a_n, b_n$	0.001, 1
Growth rate parameters	$a_i, b_i, a_L, b_L$	0.03, 1, -3, 17
Area factor	$k_a$	1

Eek et al.<sup>39</sup> Its discretization leads to an algebraic equation including integral terms, which are evaluated by Gaussian quadrature in the same fashion described in the third section. The structure of the Jacobian matrices of this problem is illustrated in Figure 9. The first line of the matrices represents the BC, including an artificial differential zero added to avoid DAE systems of index 1, the three last lines represent the global balance equations included in the model (Eqs. 68 and 69).

Figure 10 shows how the MFEM is capable of dealing with the dynamics of this particulate process. The initial solution (Figure 10c) is a  $\Gamma(\bullet)$  distribution with the maximum at  $v = 2$  and a discontinuity at  $v = 0$ , due to the boundary condition. The growth mechanism causes the initial  $\Gamma(\bullet)$  distribution to move toward the larger particles region, as registered in this figure. On the other hand, the nucleation increases the density of the smaller sized particles, whose size augments due to the growth mechanism. One may see that in the initial stages, when the concentration of calcium and carbonate ions is high and supersaturation occurs, the nucleation increases the number of particles in the system. The dynamic solution includes a moving point that can be seen in Figure 10c, at which both phenomena are balanced. This pattern was already registered in Case 4. The growth mechanism is over-expressed ahead of the discontinuity, and on the leftside the nucleation is dominant.

The trade-off between both phenomena leads to the motion of the discontinuity point (Figures 10a and 10c). As a consequence of the development of large gradients of the residuals in the vicinity of such a point the nodes tend to concentrate there and follow the shock, as it is shown in Figure 10a. Once the saturation of ionic species is reached the rates of both phenomena become very low (Figure 10b) due to  $\lim_{t \rightarrow \infty} C_s = 0$  (Figure 10d) at the end of the simulation. On the other hand, the saturation leads to a local maximum on the steady state PSD at the point where  $d[G(v,t)]/dv$  reaches a minimum.

**Case 8** – Calcium carbonate semibatch reactor with aggregation

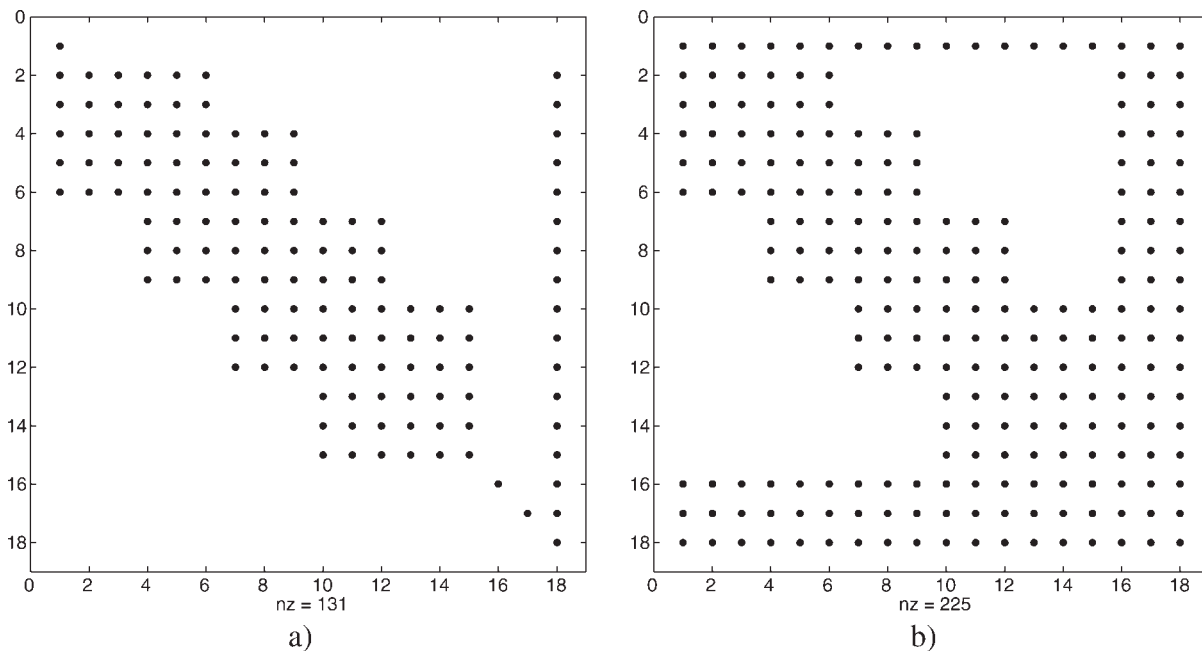
The model employed to represent the semibatch reactor dynamics comprising the aggregation of particles is based on Eqs. 60–66 previously introduced in Case 7, to which the aggregation terms were added

$$\frac{\partial n(v,t)}{\partial t} + \frac{n(v,t)}{V} \frac{dV}{dt} = - \frac{\partial [G(v,t) n(v,t)]}{\partial v} + \frac{1}{2} \int_0^v n(v',t) \times n(\sqrt[3]{v^3 - v'^3}, t) q(\sqrt[3]{v^3 - v'^3}, v') \frac{v^2}{(v^3 - v'^3)^{2/3}} dv' - n(v,t) \int_0^\infty n(v',t) q(v, v') dv' \quad (67)$$

$$\frac{d(V C_i)}{dt} = q_i^f C_i^f - k_a V \int_0^\infty G(v,t) n(v,t) v^2 dv, \quad i = 1, 2 \quad (68)$$

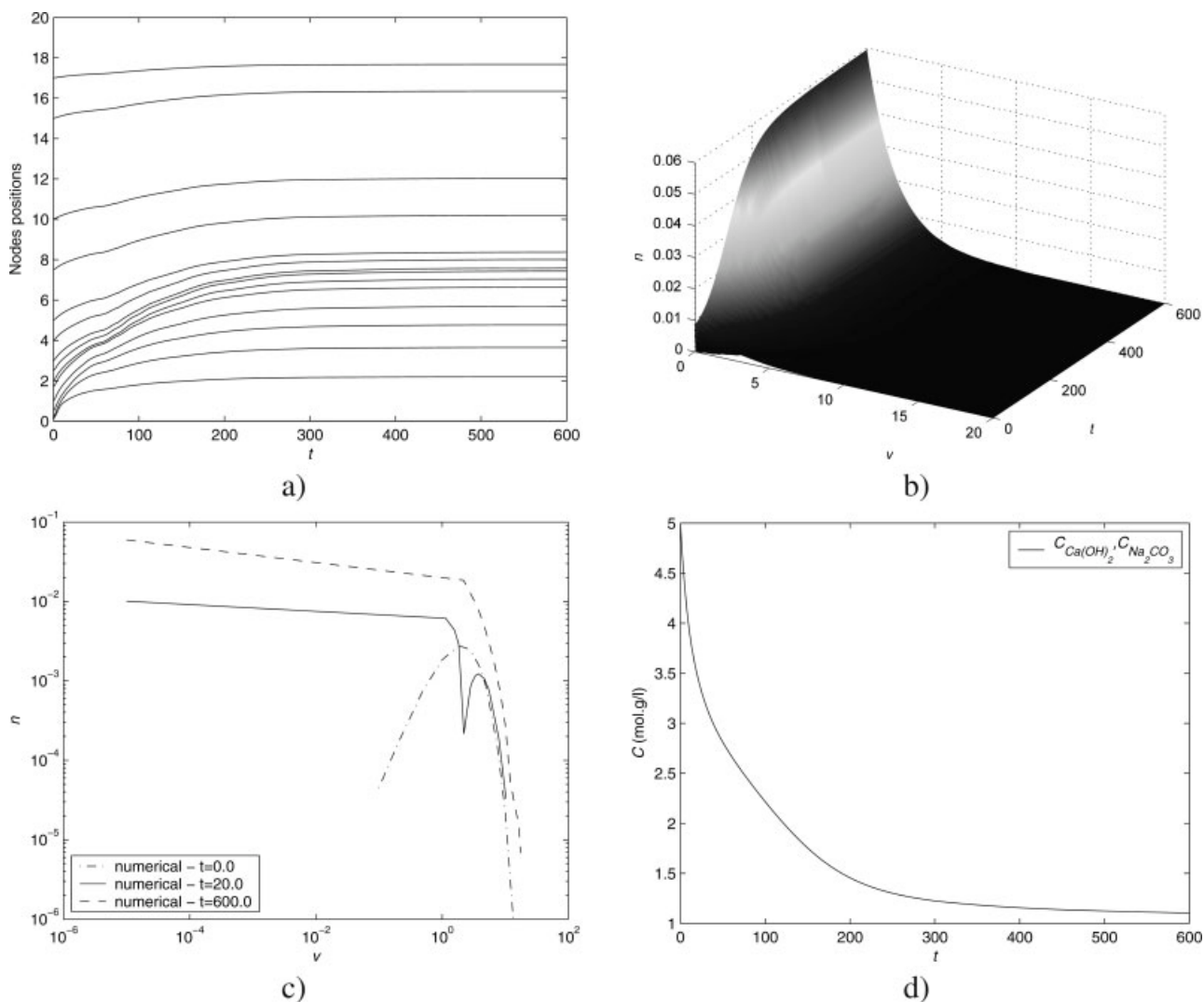
$$\frac{dV}{dt} = q_1^f + q_2^f \quad (69)$$

$$n(0,t) = \frac{1}{G(0,t)} \int_0^\infty a_n C_s^{b_n} n(v,t) v^{2.5} dv \quad (70)$$



**Figure 9. Jacobian structures for semibatch reactor model (NE = 5).**

(a) Differential; and (b) Algebraic.



**Figure 10. Solution of the dynamic model for the semibatch precipitation of  $\text{CaCO}_3$ .**

(a) Nodes movement; (b) 3-D solution; (c) numerical solution; and (d) reactant concentrations.

$$G(v, t) = \frac{a_s C_s^{b_s}}{1 + \exp(-a_L (v - b_L))} \quad (71)$$

$$n(v, 0) = \frac{0.01}{2} v^2 \exp(-v) \quad (72)$$

$$q(v, v') = 3.0 \times 10^{-5} \left( \sqrt{\frac{C_1 C_2}{k_{sp}}} - 1 \right)^2 \quad (73)$$

$$C_s = \sqrt{C_1 C_2} - 1 \quad (74)$$

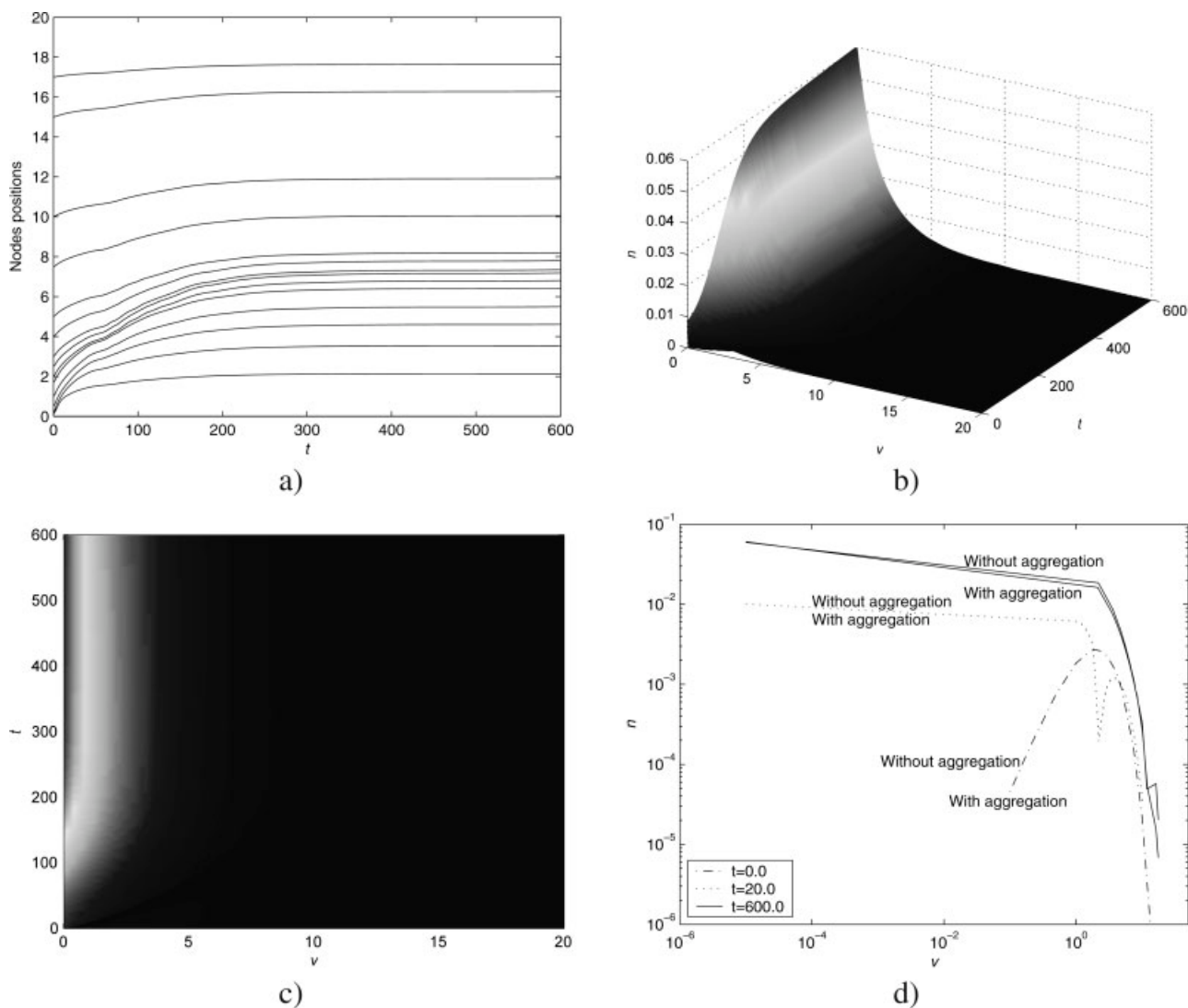
$$t \in [0, 600.0], \quad v \in [0, 20.0]$$

where  $k_{sp}$  is the equilibrium constant of calcium carbonate ( $4.7 \times 10^{-3} \text{ mol}^2/\text{m}^6$ ),<sup>40</sup> and other data are listed in Table 4. The aggregation rate expression used is owed to Collier and Hounslow,<sup>40</sup> and stands on the assumption that supersaturation is the driving force for growth, nucleation and aggrega-

tion.<sup>22</sup> Notice that the original kernel rate was normalized with respect to Lee et al.<sup>17</sup> model domains.

The numerical treatment of the integrals modelling the aggregation phenomenon is similar to the one described to handle PBE with aggregation terms. The algebraic Jacobian matrix has a full structure similar to the one used in Cases 5 and 6, meaning it does not share the structure with Case 7.

The movement of the nodes following the discontinuity, due to the growth of previously nucleated particles, can be seen in Figure 11a, and the dynamic solution is shown in Figure 11b. The increase in particles' density in the vicinity of the left boundary is due to nucleation, and is followed by an increase in their dimension, due to growth, this evolution is highlighted in the surface plot in Figure 11c. On the other hand, the increase in growth leads to an increase in nucleation rate, since it is proportional to  $v^{2.5}$ . The combination of both phenomena gives origin to a front that broadens until the supersaturation conditions are nearly reached. Figure 11d allows comparing the solutions for Cases 7 and 8. The strik-



**Figure 11. Solution of the dynamic model for the semibatch precipitation of  $\text{CaCO}_3$ .**

(a) Nodes movement; (b) 3-D solution; (c) solution surface plot; and (d) comparison of numerical solutions of Cases 7 and 8.

ing feature in this figure is that in the initial stages differences are not registered, and this was expected because the nucleation and growth phenomena dominate the solution in Case 8. In the later stages the solutions obtained are slightly different, and the PSD for Case 8 is marginally broader than when aggregation was neglected. When aggregation phenomenon is considered the lower density of particles of smaller size is due to their aggregation producing particles of larger dimension, therefore, leading to a decrease on the total number of particles inside the reactor. As the density of particles of larger size increases, the PSD broadens.

Table 5 lists the information on the performance of the MFEM employed in Cases 7 and 8, emphasizing once more the low-computational effort required by the method to handle intricate problems commonly found in industrial units. The CPU time required to solve the problem described by Case 8 is one-order of magnitude larger than that in Case 7. The main reasons for such an increase are the need to numerically determine all integral terms modelling the aggre-

gation, and to determine a dense algebraic Jacobian matrix required by the DAE solver at each time step. Although the CPU time needed for solving Case 8 is larger the computational effort is not dramatically increased and the MFEM provides a stable and accurate solution

**Table 5. Parameters Employed in the Simulation of the Semibatch Precipitation Reactor Dynamics**

Case	Nodes	Grid Initialization	$\varepsilon$	CPU Time <sup>†</sup>
7	16	$s = [0.0, 0.1, 0.2, 0.5, 1.0, 1.7, 2.0, 2.5, 3.0, 4.0, 5.0, 7.5, 10.0, 15.0, 17.0, 20.0]^T$	$7.0 \times 10^{-6}$	10.864
8	16	$s = [0.0, 0.1, 0.2, 0.5, 1.0, 1.7, 2.0, 2.5, 3.0, 4.0, 5.0, 7.5, 10.0, 15.0, 17.0, 20.0]^T$	$7.0 \times 10^{-6}$	321.451

<sup>†</sup>On a Windows Pentium 2.80 GHz computer.

## Conclusions

This article presents a moving finite elements method focused on the solution of population balance equations. The method is based on the minimization of the square norm of the residuals all over the spatial domain, and uses cubic Hermite polynomials as local basis functions. The performance of this methodology is evaluated by comparing the numerical solution to the analytical solution of PBE, when these are available. The MFEM proved to be able to handle a vast range of phenomena and combinations of mechanisms and kernels. In most cases it produces accurate solutions and, although CPU time is increased, prohibitive values are not attained. The case studies presented show the ability of the method to deal with problems developing moving fronts and near discontinuities by automatically concentrating the nodes on the zones where residuals' gradients are higher. The performance of the MFEM was also tested when simulating a semibatch reactor used to produce calcium carbonate particles. The model of the unit includes a PBE, to describe the particles behavior, coupled with global balance differential equations that represent the continuous phase dynamics. The method confirmed its robustness and accuracy when handling this problem. The accuracy it guarantees, and the reduced computational effort it requires enables one to consider it as a promising alternative to solve PBE based systems. This method has also the potential to be enclosed into robust control algorithms that often use process model forecasts to compute a set of optimal control strategies of units where particulate processes occur.

## Literature Cited

1. Fredrickson AG. Population balance equations for cell and microbial cultures revisited. *AIChE J.* 2003;49:1050–1059.
2. Immanuel CD, Doyle III FJ, Cordeiro CF, Sundaram SS. Population balance PSD model for emulsion polymerization with steric stabilizers. *AIChE J.* 2003;49:1392–1404.
3. Ramkrishna D. *Population balances—Theory and applications to particulate systems in engineering.* San Diego: Academic Press; 2000.
4. Rigopoulos S, Jones AG. Finite-element scheme for solution of the dynamic population balance equation. *AIChE J.* 2003;49:1127–1139.
5. El-Farra NH, Chiu TY, Christofides PD. Analysis and control of particulate processes with input constraints. *AIChE J.* 2001;47:1849–1865.
6. Immanuel CD, Doyle III FJ. Hierarchical multiobjective strategy particle-size distribution control. *AIChE J.* 2003;49:2383–2399.
7. Kostoglou M, Karabelas AJ. Evaluation of zero order methods for simulating particle coagulation. *J Colloid and Interface Sci.* 1994;163:420–431.
8. Hill PG, Ng KM. New discretization procedure for the breakage equation. *AIChE J.* 1995;41:1204–1216.
9. Kumar S, Ramkrishna D. On the solution of population balance equations by discretization – I. A fixed pivot technique. *Chem Eng Sci.* 1996;51:1311–1332.
10. Kumar S, Ramkrishna D. On the solution of population balance equations by discretization – II. A moving pivot technique. *Chem Eng Sci.* 1996;51:1333–1342.
11. Hounslow MJ, Ryall RL, Marshall VR. A discretized population balance for nucleation, growth and breakage. *AIChE J.* 1988;34:1821–1832.
12. Marchal P, David R, Klein JP, Villermaux J. Crystallization and precipitation engineering – 1. An efficient method for solving population balances in crystallization with agglomeration. *Chem Eng Sci.* 1990;43:59–67.
13. Kumar S, Ramkrishna D. On the solution of population balance equations by discretization – III. Nucleation, growth and aggregation of particles. *Chem Eng Sci.* 1997;52:4659–4679.
14. Bürger R, Karlsen KH, Towers JD. Closed-form and finite difference solutions to a population model of grinding mills. *J Eng Mathem.* 2005;51:165–195.
15. Gelbard F, Seinfeld JH. Numerical solution of the dynamic equation for particulate systems. *J Comp Phys.* 1978;28:357–375.
16. Nicmanis M, Hounslow MJ. Finite-element methods for steady-state population balance equations. *AIChE J.* 1998;44:2258–2272.
17. Lee K, Lee JH, Yang DR, Mahoney AW. Integrated run-to-run and on-line model-based control of particle size distribution for a semi-batch precipitation reactor. *Comput Chem Eng.* 2002;26:1117–1131.
18. Immanuel CD, Cordeiro CF, Sundaram SS, Meadows ES, Crowley TJ, Doyle III FJ. Modeling of particle size distribution in emulsion co-polymerization: comparison with experimental data and parametric sensitivity studies. *Comput Chem Eng.* 2002;26:1133–1152.
19. Crowley TJ, Meadows ES, Doyle III FJ. Numerical issues in solving population balance equations for particle size distribution control in emulsion polymerization. In: *Proc. of the American Control Conference*; 1138–1142, 1999.
20. Attarakih MM, Bart H-J, Faqir NM. Optimal moving and fixed grids for the solution of discretized population balances in batch and continuous systems: droplet breakage. *Chem Eng Sci.* 2003;58:1251–1269.
21. Eyre D, Wright CJ, Reuter G. Spline-collocation with adaptive mesh grading for solving the stochastic collection equation. *J Comput Phys.* 1988;78:288–304.
22. Mahoney AW, Ramkrishna D. Efficient solution of population balance equations with discontinuities by finite elements. *Chem Eng Sci.* 2002;57:1107–1119.
23. Tsang TH, Rao A. A Moving Finite Element Method for the population balance equation. *Int J Numer Methods in Fluids.* 1990;10:753–769.
24. Miller K, Miller RN. Moving finite elements. I. *SIAM J Numerical Analysis.* 1981;18:1019–1032.
25. Miller K, Miller RN. Moving finite elements. II. *SIAM J Numerical Analysis.* 1981;18:1033–1057.
26. Baines MJ, Wathen AJ. Moving finite elements methods for evolutionary problems. I. Theory. *J Comput Phys.* 1988;79:270–297.
27. Hansen JA, Hassager O. A new Moving Finite Element Method based on quadratic approximation functions. *Intern J Numerical Methods.* 1989;28:415–430.
28. Herbst BM, Schoombie SW, Griffiths DF, Mitchell D. Generalised Petrov-Galerkin methods for the numerical solution of Burger's equation. *J Comput Phys.* 1984;20:1273–1289.
29. Pipilis KG. *Higher order moving finite element methods for systems described by partial differential-algebraic equations.* Imperial College of Science, Technology and Medicine; 1991. PhD Thesis.
30. Duarte BPM. *Method of Moving Finite Elements applied to the solution of models of front reaction.* Universidade de Coimbra; 1995. PhD Thesis.
31. Sereno C, Rodrigues A, Villadsen J. Solution of partial differential equation systems by moving finite elements method. *Comput Chem Eng.* 1992;16:583–592.
32. Coimbra MC, Sereno C, Rodrigues AE. A moving finite element method for the solution of two-dimensional time dependent models. *Applied Numerical Analysis.* 2003;44:449–469.
33. Villadsen J, Michelsen ML. *Solution of differential equations models by polynomial approximation.* NJ: Prentice-Hall, Inc; 1978.
34. Abramowitz M, Stegun IA. *Handbook of mathematic functions with formulas, graphs and mathematical tables.* New York: Dover Publications, Inc; 1965.
35. Gelinas RJ, Doss SK, Miller K. The Moving Finite Element Method: Applications to general partial differential equations with multiple large gradients. *J Comput Phys.* 1981;40:202–249.
36. Jarvis RB, Pantelides CC. *DASOLV – A differential-algebraic equation solver. tech. rep.* Imperial College of Science, Technology and Medicine; 1992.
37. Immanuel CD, Doyle III FJ. Computationally efficient solution of population balance models incorporating nucleation, growth and coagulation: applications to emulsion polymerization. *Chem Eng Sci.* 2003;58:3681–3698.
38. Ramabhadran TE, Peterson TW, Seinfeld JH. Dynamics of aerosol coagulation and condensation. *AIChE J.* 1976;22:840–851.
39. Eek RA, Dijkstra S, Rosmalen GM. Dynamic modeling of suspension crystallizers using experimental data. *AIChE J.* 1995;41:571–584.
40. Collier AP, Hounslow MJ. Growth and aggregation rates for calcite and calcium oxalate monohydrate. *AIChE J.* 1999;45:2298–2305.

## Appendix

Equations 17–19 lead to

$$\int_{s_{k-2}}^{s_{k-1}} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{k-1,i}(t) H_i(u, h_{k-1}(t)) + a_{k-1,i}(t) \frac{\partial H_i(u, h_{k-1}(t))}{\partial u} \frac{\partial u}{\partial h_{k-1}(t)} \dot{h}_{k-1}(t) \right. \right. \\ \left. \left. + a_{k-1,i}(t) \frac{\partial H_i(u, h_{k-1}(t))}{\partial h_{k-1}(t)} \dot{h}_{k-1}(t) + \frac{a_{k-1,i}(t)}{h_{k-1}(t)} \frac{\partial H_i(u, h_{k-1}(t))}{\partial u} \right] H_i(u, h_{k-1}(t)) \right\} dh_{k-1} \\ + \int_{s_{k-1}}^{s_k} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{k,i}(t) H_i(u, h_k(t)) + a_{k,i}(t) \frac{\partial H_i(u, h_k(t))}{\partial u} \frac{\partial u}{\partial h_k(t)} \dot{h}_k(t) + a_{k,i}(t) \frac{\partial H_i(u, h_k(t))}{\partial h_k(t)} \dot{h}_k(t) \right. \right. \\ \left. \left. + \frac{a_{k,i}(t)}{h_k(t)} \frac{\partial H_i(u, h_k(t))}{\partial u} \right] H_i(u, h_k(t)) \right\} dh_k = 0 \quad k \in \{2, \dots, NE-1\} \quad (A1)$$

$$\int_{s_0}^{s_1} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{1,i}(t) H_i(u, h_1(t)) + a_{1,i}(t) \frac{\partial H_i(u, h_1(t))}{\partial u} \frac{\partial u}{\partial h_1(t)} \dot{h}_1(t) + a_{1,i}(t) \frac{\partial H_i(u, h_1(t))}{\partial h_1(t)} \dot{h}_1(t) \right. \right. \\ \left. \left. + \frac{a_{1,i}(t)}{h_1(t)} \frac{\partial H_i(u, h_1(t))}{\partial u} \right] H_i(u, h_1(t)) \right\} dh_1 = 0 \quad (A2)$$

$$\int_{s_{NE-1}}^{s_{NE}} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{NE,i}(t) H_i(u, h_{NE}(t)) + a_{NE,i}(t) \frac{\partial H_i(u, h_{NE}(t))}{\partial u} \frac{\partial u}{\partial h_{NE}(t)} \dot{h}_{NE}(t) \right. \right. \\ \left. \left. + a_{NE,i}(t) \frac{\partial H_i(u, h_{NE}(t))}{\partial h_{NE}(t)} \dot{h}_{NE}(t) + \frac{a_{NE,i}(t)}{h_{NE}(t)} \frac{\partial H_i(u, h_{NE}(t))}{\partial u} \right] H_i(u, h_{NE}(t)) \right\} dh_{NE} = 0 \quad (A3)$$

Employing Gaussian quadrature, based on *NC* collocation points, the following equations are obtained

$$\sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{k-1,i} H_i(u_m, h_{k-1}) - a_{k-1,i} \frac{dH_i(u_m, h_{k-1})}{du} \frac{u_m}{h_{k-1}} \dot{h}_{k-1} + a_{k-1,i} \frac{dH_i(u_m, h_{k-1})}{dh_{k-1}} \dot{h}_{k-1} + \frac{a_{k-1,i}}{h_{k-1}} \frac{dH_i(u_m, h_{k-1})}{du} \right] H_3(u_m, h_{k-1}) \right\} \\ \times h_{k-1} w_m + \sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{k,i} H_i(u_m, h_k) - a_{k,i} \frac{dH_i(u_m, h_k)}{du} \frac{u_m}{h_k} \dot{h}_k + a_{k,i} \frac{dH_i(u_m, h_k)}{dh_k} \dot{h}_k + \frac{a_{k,i}}{h_k} \frac{dH_i(u_m, h_k)}{du} \right] H_1(u_m, h_k) \right\} \\ h_k w_m = 0 \quad k \in \{2, \dots, NE-1\} \quad (A4)$$

$$\sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{k-1,i} H_i(u_m, h_{k-1}) - a_{k-1,i} \frac{dH_i(u_m, h_{k-1})}{du} \frac{u_m}{h_{k-1}} \dot{h}_{k-1} + a_{k-1,i} \frac{dH_i(u_m, h_{k-1})}{dh_{k-1}} \dot{h}_{k-1} + \frac{a_{k-1,i}}{h_{k-1}} \frac{dH_i(u_m, h_{k-1})}{du} \right] H_4(u_m, h_{k-1}) \right\} \\ \times h_{k-1} w_m + \sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{k,i} H_i(u_m, h_k) - a_{k,i} \frac{dH_i(u_m, h_k)}{du} \frac{u_m}{h_k} \dot{h}_k + a_{k,i} \frac{dH_i(u_m, h_k)}{dh_k} \dot{h}_k + \frac{a_{k,i}}{h_k} \frac{dH_i(u_m, h_k)}{du} \right] H_2(u_m, h_k) \right\} \\ h_k w_m = 0 \quad k \in \{2, \dots, NE-1\} \quad (A5)$$



$$\sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{1,i} H_i(u_m, h_1) - a_{1,i} \frac{dH_i(u_m, h_1)}{du} \frac{u_m}{h_1} \dot{h}_1 + a_{1,i} \frac{dH_i(u_m, h_1)}{dh_1} \dot{h}_1 + \frac{a_{1,i}}{h_1} \frac{dH_i(u_m, h_1)}{du} \right] H_1(u_m, h_1) \right\} h_1 w_m = 0 \quad (A6)$$

$$\sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{1,i} H_i(u_m, h_1) - a_{1,i} \frac{dH_i(u_m, h_1)}{du} \frac{u_m}{h_1} \dot{h}_1 + a_{1,i} \frac{dH_i(u_m, h_1)}{dh_1} \dot{h}_1 + \frac{a_{1,i}}{h_1} \frac{dH_i(u_m, h_1)}{du} \right] H_2(u_m, h_1) \right\} h_1 w_m = 0 \quad (A7)$$

$$\sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{NE,i} H_i(u_m, h_{NE}) - a_{NE,i} \frac{dH_i(u_m, h_{NE})}{du} \frac{u_m}{h_{NE}} \dot{h}_{NE} + a_{NE,i} \frac{dH_i(u_m, h_{NE})}{dh_{NE}} \dot{h}_{NE} + \frac{a_{NE,i}}{h_{NE}} \frac{dH_i(u_m, h_{NE})}{du} \right] H_3(u_m, h_{NE}) \right\} \times h_{NE} w_m = 0 \quad (A8)$$

$$\sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{NE,i} H_i(u_m, h_{NE}) - a_{NE,i} \frac{dH_i(u_m, h_{NE})}{du} \frac{u_m}{h_{NE}} \dot{h}_{NE} + a_{NE,i}(t) \frac{dH_i(u_m, h_{NE})}{dh_{NE}} \dot{h}_{NE} + \frac{a_{NE,i}}{h_{NE}} \frac{dH_i(u_m, h_{NE})}{du} \right] H_4(u_m, h_{NE}) \right\} \times h_{NE} w_m = 0 \quad (A9)$$

Equations 24–26 lead to

$$\int_{s_{k-2}}^{s_{k-1}} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{k-1,i}(t) H_i(u, h_{k-1}(t)) + a_{k-1,i}(t) \frac{\partial H_i(u, h_{k-1}(t))}{\partial u} \frac{\partial u}{\partial h_{k-1}(t)} \dot{h}_{k-1}(t) + a_{k-1,i}(t) \frac{\partial H_i(u, h_{k-1}(t))}{\partial h_{k-1}(t)} \dot{h}_{k-1}(t) + \frac{a_{k-1,i}(t)}{h_{k-1}(t)} \frac{\partial H_i(u, h_{k-1}(t))}{\partial u} \right] \times \frac{\partial H_i(u, h_{k-1}(t))}{\partial u} \right\} \times \left[ a_{k-1,i}(t) \frac{\partial H_i(u, h_{k-1}(t))}{\partial u} \left( \frac{\partial u}{\partial s_k(t)} \dot{s}_k(t) + \frac{\partial u}{\partial s_{k-1}(t)} \dot{s}_{k-1}(t) \right) + a_{k-1,i}(t) \frac{\partial H_i(u, h_{k-1}(t))}{\partial h_{k-1}(t)} (\dot{s}_k(t) - \dot{s}_{k-1}(t)) \right] \Big] dh_{k-1}$$

$$\int_{s_{k-1}}^{s_k} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{k,i}(t) H_i(u, h_k(t)) + a_{k,i}(t) \frac{\partial H_i(u, h_k(t))}{\partial u} \frac{\partial u}{\partial h_k(t)} \dot{h}_k(t) + a_{k,i}(t) \frac{\partial H_i(u, h_k(t))}{\partial h_k(t)} \dot{h}_k(t) + \frac{a_{k,i}(t)}{h_k(t)} \frac{\partial H_i(u, h_k(t))}{\partial u} \right] \times \left[ a_{k,i}(t) \frac{\partial H_i(u, h_{k-1}(t))}{\partial u} \left( \frac{\partial u}{\partial s_k(t)} \dot{s}_k(t) + \frac{\partial u}{\partial s_{k+1}(t)} \dot{s}_{k+1}(t) \right) + a_{k,i}(t) \frac{\partial H_i(u, h_k(t))}{\partial h_k(t)} (\dot{s}_{k+1}(t) - \dot{s}_k(t)) \right] \Big] dh_k = 0 \quad k \in \{2, \dots, NE-1\} \quad (A10)$$

$$\int_{s_0}^{s_1} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{1,i}(t) H_i(u, h_1(t)) + a_{1,i}(t) \frac{\partial H_i(u, h_1(t))}{\partial u} \frac{\partial u}{\partial h_1(t)} \dot{h}_1(t) + a_{1,i}(t) \frac{\partial H_i(u, h_1(t))}{\partial h_1(t)} \dot{h}_1(t) + \frac{a_{1,i}(t)}{h_1(t)} \frac{\partial H_i(u, h_1(t))}{\partial u} \right] \times \left[ a_{1,i}(t) \frac{\partial H_i(u, h_1(t))}{\partial u} \left( \frac{\partial u}{\partial s_0(t)} \dot{s}_0(t) + \frac{\partial u}{\partial s_1(t)} \dot{s}_1(t) \right) + a_{1,i}(t) \frac{\partial H_i(u, h_1(t))}{\partial h_1(t)} (\dot{s}_1(t) - \dot{s}_0(t)) \right] \Big] dh_1 = 0 \quad (A11)$$

$$\int_{s_{NE-1}}^{s_{NE}} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{NE,i}(t) H_i(u, h_{NE}(t)) + a_{NE,i}(t) \frac{\partial H_i(u, h_{NE}(t))}{\partial u} \frac{\partial u}{\partial h_{NE}(t)} \dot{h}_{NE}(t) + a_{NE,i}(t) \frac{\partial H_i(u, h_{NE}(t))}{\partial h_{NE}(t)} \dot{h}_{NE}(t) + \frac{a_{NE,i}(t)}{h_{NE}(t)} \frac{\partial H_i(u, h_{NE}(t))}{\partial u} \right] \times \left[ a_{NE,i}(t) \frac{\partial H_i(u, h_{NE}(t))}{\partial u} \left( \frac{\partial u}{\partial s_{NE}(t)} \dot{s}_{NE}(t) + \frac{\partial u}{\partial s_{NE-1}(t)} \dot{s}_{NE-1}(t) \right) + a_{NE,i}(t) \frac{\partial H_i(u, h_{NE}(t))}{\partial h_{NE}(t)} (\dot{s}_{NE}(t) - \dot{s}_{NE-1}(t)) \right] \Big] dh_{NE} = 0 \quad (A12)$$

which yields

$$\begin{aligned}
 & \sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{k-1,i} H_i(u_m, h_{k-1}) - a_{k-1,i} \frac{dH_i(u_m, h_{k-1})}{du} \frac{u_m}{h_{k-1}} \dot{h}_{k-1} + a_{k-1,i} \frac{dH_i(u_m, h_{k-1})}{dh_{k-1}} \dot{h}_{k-1} + \frac{a_{k-1,i}}{h_{k-1}} \frac{dH_i(u_m, h_{k-1})}{du} \right] \right. \\
 & \quad \times \left[ a_{k-1,i} \frac{dH_i(u_m, h_{k-1})}{du} \left( \frac{1-u_m}{h_{k-1}} \dot{s}_k + \frac{u_m}{h_{k-1}} \dot{s}_{k-1} \right) + a_{k-1,i} \frac{dH_i(u_m, h_{k-1})}{dh_{k-1}} (\dot{s}_k - \dot{s}_{k-1}) \right] \times h_{k-1} w_m \left. \right\} \\
 & \quad + \sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{k,i} H_i(u_m, h_k) - a_{k,i} \frac{dH_i(u_m, h_k)}{du} \frac{u_m}{h_k} \dot{h}_k + a_{k,i} \frac{dH_i(u_m, h_k)}{dh_k} \dot{h}_k + \frac{a_{k,i}}{h_k} \frac{dH_i(u_m, h_k)}{du} \right] \right. \\
 & \quad \times \left[ a_{k,i} \frac{dH_i(u_m, h_k)}{du} \left( \frac{u_m-1}{h_k} \dot{s}_k - \frac{u_m}{h_k} \dot{s}_{k+1} \right) + a_{k,i} \frac{dH_i(u_m, h_{k-1})}{dh_k} (\dot{s}_{k+1} - \dot{s}_k) \right] h_k w_m \left. \right\} = 0 \quad k \in \{2, \dots, NE-1\} \quad (A13)
 \end{aligned}$$

$$\begin{aligned}
 & \sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{1,i} H_i(u_m, h_1) - a_{1,i} \frac{dH_i(u_m, h_1)}{du} \frac{u_m}{h_1} \dot{h}_1 + a_{1,i} \frac{dH_i(u_m, h_1)}{dh_1} \dot{h}_1 + \frac{a_{1,i}}{h_1} \frac{dH_i(u_m, h_1)}{du} \right] \right. \\
 & \quad \times \left[ a_{1,i} \frac{dH_i(u_m, h_1)}{du} \left( \frac{u_m-1}{h_1} \dot{s}_0 - \frac{u_m}{h_1} \dot{s}_1 \right) + a_{1,i} \frac{dH_i(u_m, h_1)}{dh_1} (\dot{s}_1 - \dot{s}_0) \right] h_1 w_m \left. \right\} = 0 \quad (A14)
 \end{aligned}$$

$$\begin{aligned}
 & \sum_{m=1}^{NC} \left\{ \sum_{i=1}^4 \left[ \dot{a}_{NE,i} H_i(u_m, h_{NE}) - a_{NE,i} \frac{dH_i(u_m, h_{NE})}{du} \frac{u_m}{h_{NE}} \dot{h}_{NE} + a_{NE,i} \frac{dH_i(u_m, h_{NE})}{dh_{NE}} \dot{h}_{NE} + \frac{a_{NE,i}}{h_{NE}} \frac{dH_i(u_m, h_{NE})}{du} \right] \right. \\
 & \quad \times \left[ a_{NE,i} \frac{dH_i(u_m, h_{NE})}{du} \left( \frac{1-u_m}{h_{NE}} \dot{s}_{NE} + \frac{u_m}{h_{NE}} \dot{s}_{NE-1} \right) + a_{NE,i} \frac{dH_i(u_m, h_{NE})}{dh_{NE}} (\dot{s}_{NE} - \dot{s}_{NE-1}) \right] h_{NE} w_m \left. \right\} = 0 \quad (A15)
 \end{aligned}$$

The discretization of the boundary condition renders

$$\sum_{i=1}^4 a_{1,i}(t) H_i(0, h_1(t)) = 0 \quad (A16)$$

*Manuscript received Feb. 5, 2007, and revision received Oct. 22, 2007.*