Computers and Chemical Engineering 162 (2022) 107808

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Efficient numerical schemes for population balance models

Pavan K. Inguva, Kaylee C. Schickel, Richard D. Braatz*

Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 2 December 2021
Revised 15 April 2022
Accepted 16 April 2022
Available online 18 April 2022

Keywords:
Population balance models
Numerical methods
Species balances
Numerical simulation
Numerical analysis

ABSTRACT

Population balance models (PBM) describe a wide array of physical, chemical, and biological processes having a distribution over some intrinsic property, and are used to model cells, viruses, aggregates, bubbles, and crystals. The ubiquity of PBMs motivates generalizable and accurate approaches for their numerical solution. Typically, high-order finite difference or finite volume methods are used. We propose a finite difference scheme at the limit of numerical stability that results in discretization error that is *zero* for certain classes of PBMs and low enough to be acceptable in other applications. The scheme employs specially constructed meshes and, in some cases, variable transformations. The scheme has very low computational cost – sometimes as low as memory reallocation with no floating point operations. Case studies are presented throughout that demonstrate the scheme's performance in relation to other commonly employed schemes.

1. Introduction

The spatiotemporal dynamics of populations are well described by population balance models (PBMs). These partial differential equation (PDE) models are applicable to a wide variety of physical, chemical, and biological processes that include cells (Fredrickson et al., 1967), viruses (Sidorenko et al., 2008), aggregates (Arosio et al., 2012; Jeldres et al., 2018), bubbles (Wang et al., 2005), droplets (Schmidt et al., 2006), crystals (Abegg et al., 1968; Woo et al., 2006; Lapidot et al., 2019; Colucci et al., 2020), and precipitates (Szilágyi et al., 2015). In general, a species balance for a spatially well-mixed system can be written as an ordinary differential equation (ODE) of the form

$$\frac{df(t)}{dt} = h(t, f), \tag{1}$$

where f is the species density (e.g., concentration), t is time, and h is a forcing function that can be a function of f and t. For multiple species, the f and h are vectors, with each element corresponding to a single species. For the purposes of analyzing finite difference methods, f and h can be treated as scalars, as the extension to the vector case is straightforward.

When a species has an additional intrinsic variable such as size or age, the PDE model corresponding to the ODE can be written as

$$\frac{\partial f(t,a)}{\partial t} + \frac{\partial (Gf(t,a))}{\partial a} = h(t, a, f), \tag{2}$$

where a is an intrinsic variable and $G = da/dt$ is its rate of change of a with respect to time. The G is usually referred to as the "growth rate" in the PBM literature, irrespective of whether the G is a true growth rate (e.g., the growth of a crystal) or not (e.g., the aging of a cell), and that nomenclature is used here. The generalization of (2) to multiple intrinsic variables is

$$\frac{\partial f(t,a_1,\ldots,a_m)}{\partial t} + \sum_{i=1}^{m} \frac{\partial (G_i f(t,a_1,\ldots,a_m))}{\partial a_i} = h(t,a_1,\ldots,a_m,f), \tag{3}$$

where m is the number of intrinsic variables. The PBM and associated results can be further extended to a steady-state slug or plug flow through a tube by replacing the time t with the residence time τ, or to dynamic slug or plug flow through a tube by including both derivatives with respect to time and residence time,

$$\frac{\partial f(t,a_1,\ldots,a_m)}{\partial t} + \frac{\partial f(t,a_1,\ldots,a_m)}{\partial \tau} + \sum_{i=1}^{m} \frac{\partial (G_i f(t,a_1,\ldots,a_m))}{\partial a_i} = h(t,a_1,\ldots,a_m,f), \tag{4}$$

where $\tau = z/u$, with z being the axial position along the tube and u being the mean flow velocity.

A large literature derives finite difference and finite volume methods for the numerical solution of PBMs as well as for more general PDEs, with the focus being to minimize discretization errors (e.g., see Gunawan et al., 2004; Qamar et al., 2006; Woo et al., 2006; Gunawan et al., 2008; Mesbah et al., 2009 and citations therein). It is especially important that the numerical solution of this class of hyperbolic PDEs strive to reduce numerical diffusion

* Corresponding author.
E-mail address: braatz@mit.edu (R.D. Braatz).

https://doi.org/10.1016/j.compchemeng.2022.107808
0098-1354/

(i.e., smearing of the numerical solution compared to the true solution) and numerical dispersion (i.e., aphysical oscillations in the numerical solution compared to the true solution). The classic upwind scheme is most commonly used in pedagogical settings to introduce numerical methods (e.g., see LeVeque, 2002; Inguva et al., 2021), but is rarely used by practitioners because it is regarded as first-order accurate only and is prone to significant numerical diffusion. However, this article shows that, when applied in a very particular and unexpected way, the upwind scheme gives highly accurate solutions to a range of PBMs, while having very low computational cost. Along the way, the analysis dispels the misconception that a numerical method should not operate at the limit of numerical stability.

2. Theory and methods

To demonstrate the formulation of the upwind scheme to various classes of PBMs, this section is structured as a series of cases whereby each case outlines the numerical solution of a class of PBMs. Two other higher order schemes are also considered, the Lax–Wendroff and leapfrog schemes.

2.1. PBMs with constant growth rate

Consider the homogeneous population balance model,

$$\frac{\partial f(t,a)}{\partial t} + g\frac{\partial f(t,a)}{\partial a} = 0, \quad f(0,a) = f_0(a), \tag{5}$$

with a single intrinsic variable a in which $G = g$ and is constant, over the real domains $t, a \geq 0$. This PBM arises when the driving force for growth is constant, which occurs for example in crystallization where a is the crystal size and the supersaturation that appears in the growth rate is held constant by a control system to provide high purity while suppressing nucleation (e.g., see Abegg et al., 1968; Dalas et al., 1991; Schall et al., 1996; Grön et al., 2003; Westin and Rasmuson, 2005; Ward et al., 2006; Woo et al., 2011 and citations therein). The case of $g = 1$ is of particular relevance for cell populations where a is the cell age and is defined to have the same units as the time t (Kurtz et al., 1998).

The upwind method can be applied to (5) to yield

$$\frac{f_{i+1}^j - f_i^j}{\Delta t} + g\frac{f_i^j - f_i^{j-1}}{\Delta a} = 0, \tag{6}$$

where i is the time index and j is the index for the intrinsic variable. This equation can be further rearranged to obtain the update equation,

$$f_{i+1}^j = f_i^j - \frac{g\Delta t}{\Delta a}(f_i^j - f_i^{j-1}). \tag{7}$$

Similarly, the Lax–Wendroff and leapfrog schemes can be implemented as

Lax–Wendroff:

$$f_{i+1}^j = f_i^j - \frac{1}{2}\frac{g\Delta t}{\Delta a}\left(f_i^{j+1} - f_i^{j-1}\right) + \frac{1}{2}\left(\frac{g\Delta t}{\Delta a}\right)^2\left(f_i^{j+1} - 2f_i^j + f_i^{j-1}\right). \tag{8}$$

Leapfrog:

$$f_{i+1}^j = f_{i-1}^j - \frac{g\Delta t}{\Delta a}(f_i^{j+1} - f_i^{j-1}),$$
$$f_1^j = f_0^j - \frac{g\Delta t}{\Delta a}(f_0^j - f_0^{j-1}), \tag{9}$$

with the first timestep for the leapfrog scheme computed using the upwind scheme. By setting $\frac{g\Delta t}{\Delta a} = 1$, both the Lax–Wendroff and upwind schemes simplify to

$$f_{i+1}^j = f_i^{j-1}. \tag{10}$$

This difference equation is an analytical solution to (5). Expressing this equation in terms of the original variables gives

$$f(t_i + \Delta t, a) = f(t_i, a - g\Delta t). \tag{11}$$

Applying (11) for $t = 0$ and $i = 1$ gives

$$f(t_1, a) = f(t_0, a - gt_1) = f_0(a - gt_1), \tag{12}$$

which is equivalent to

$$f(t, a) = f_0(a - gt), \tag{13}$$

for $t = t_1$ which is the exact analytical solution to (5) and is exact for any value of Δt.[1] This result is also evident from considering the local truncation error of the upwind scheme;

$$\text{Error} = \sum_{n=2}^{\infty}\frac{1}{n!}\left((\Delta t)^{n-1} - \frac{(\Delta a)^{n-1}}{g^{n-1}}\right)\frac{\partial^n f}{\partial t^n}\bigg|_{t,a}. \tag{14}$$

As can be seen from (14), by setting $\frac{g\Delta t}{\Delta a} = 1.0$, the discretization error given by the local truncation error becomes zero. Imposing such a condition on either Δa or Δt is equivalent to solving the problem at the limit of numerical stability for a finite difference method solution to (5) given by the Courant-Friedrichs-Lewy (CFL) condition (Courant et al., 1928). This analysis applies independently of the boundary condition and holds for any physically meaningful boundary condition e.g., for the birth of cells at $a = 0$.

The implementation of this algorithm is especially efficient since only memory reallocation is required to obtain an exact solution.

2.2. PBMs with growth rate $G(t, a)$

2.2.1. PBMs with growth rate $G(a)$

Consider a homogeneous PBM with a growth rate expression dependent on the intrinsic variable expressed in conservative form,

$$\frac{\partial f(t,a)}{\partial t} + \frac{\partial (G(a)f(t,a))}{\partial a} = 0, \quad f(0,a) = f_0(a), \tag{15}$$

with $G(a)$ continuous in a, bounded, and $G(a) > 0, \forall a$ (similar results can be derived for the case where $G(a) < 0, \forall a$ with a small change in the discretization scheme). The expression $G(a)$ is typically used to model size-dependent growth which has been heavily studied in the literature on precipitation and crystallization (Abegg et al., 1968; Hounslow, 1990; Aamir et al., 2009; Iggland and Mazzotti, 2012; Colucci et al., 2020) and citations therein. When a is the age of a growing cell, then $G(a) \neq 0$ models age-dependent growth.

An exact scheme can be constructed in two steps. First, a new variable $\hat{f} = G(a)f(t,a)$ is defined, which transforms (15) to

$$\frac{\partial \hat{f}}{\partial t} + G(a)\frac{\partial \hat{f}}{\partial a} = 0, \quad \hat{f}(0,a) = \hat{f}_0(a). \tag{16}$$

The second step is to define a change of variable,

$$\tilde{a} = \int_0^a \frac{1}{G(a')}da'. \tag{17}$$

The function $\tilde{a}(a)$ computed using (17) is always invertible. The condition $G(a) > 0$ implies that $\tilde{a}(a)$ is strictly monotonically increasing and correspondingly is one-to-one (injective). In addition, $\tilde{a}(a)$ is surjective which follows from the intermediate value theorem by recognizing that $G(a)$ is bounded and $\lim_{a\to\infty}\tilde{a}(a) = \infty$. Since $\tilde{a}(a)$ is bijective, it is invertible. Reparameterizing $\hat{f}$ in terms of $\tilde{a}$,

$$\tilde{f}(t,\tilde{a}) = \hat{f}(t,a), \tag{18}$$

[1] This analytical solution can also be derived using the method of characteristics.

where $\tilde{f}$ denotes that $\hat{f}$ has been reparameterized, simplifies (16) to

$$\frac{\partial \tilde{f}}{\partial t} + \frac{\partial \tilde{f}}{\partial \tilde{a}} = 0, \quad \tilde{f}(0, \tilde{a}) = \tilde{f}_0(\tilde{a}). \tag{19}$$

This equation can be solved exactly as demonstrated in the first case. We consider four approaches for solving this class of PBMs:

1. Apply finite differences directly to (15) on a uniform mesh ("Con-Uniform"),
2. Apply finite differences directly to (15) on a non-uniform mesh to locally enforce CFL = 1 ("Con-Nonuniform"),
3. Apply finite differences to (16) on a non-uniform mesh to locally enforce CFL = 1 ("Trans-Nonuniform"),
4. Employ the exact method developed in this article ("Exact").

The upwind, Lax–Wendroff, and leapfrog schemes for the first approach can be written as

Upwind:

$$f_{i+1}^j = f_i^j - \frac{\Delta t}{\Delta a}(G(a^j) f_i^j - G(a^{j-1}) f_i^{j-1}), \tag{20}$$

Lax–Wendroff:

$$f_{i+1}^j = f_i^j + \left(\frac{(\Delta t)^2}{2}\frac{dG}{da}(a^j) - \Delta t\right)\frac{G(a^{j+1}) f_i^{j+1} - G(a^{j-1}) f_i^{j-1}}{2\Delta a} + \frac{(\Delta t)^2}{2} G(a^j) \frac{G(a^{j+1}) f_i^{j+1} - 2G(a^j) f_i^j + G(a^{j-1}) f_i^{j-1}}{\Delta a^2}, \tag{21}$$

Leapfrog:

$$f_{i+1}^j = f_{i-1}^j - \frac{\Delta t}{\Delta a}(G(a^{j+1}) f_i^{j+1} - G(a^{j-1}) f_i^{j-1}),$$
$$f_1^j = f_0^j - \frac{\Delta t}{\Delta a}(G(a^j) f_0^j - G(a^{j-1}) f_0^{j-1}). \tag{22}$$

The value of Δt is computed using the maximum value of $G(a)$ to enforce CFL ≤ 1,

$$\Delta t = \frac{\Delta a}{\max G(a)}. \tag{23}$$

For the second and third approaches, the non-uniform grid is constructed identically using the CFL condition,

$$a^{j-1} = a^j - G(a^j)\Delta t. \tag{24}$$

It is more convenient to start from the end of the mesh and compute backwards. Starting from the other direction would potentially require solving a nonlinear equation. The mesh generation step can be done offline prior to the main computation. The various schemes for the second approach can be implemented as

Upwind:

$$f_{i+1}^j = f_i^j - \Delta t \frac{G(a^j) f_i^j - G(a^{j-1}) f_i^{j-1}}{a^j - a^{j-1}}, \tag{25}$$

Lax–Wendroff:

$$f_{i+1}^j = f_i^j + \left(\frac{(\Delta t)^2}{2}\frac{dG}{da}(a^j) - \Delta t\right)\frac{G(a^{j+1}) f_i^{j+1} - G(a^{j-1}) f_i^{j-1}}{a^{j+1} - a^{j-1}} + \frac{(\Delta t)^2}{2} G(a^j) \frac{G(a^{j+1}) f_i^{j+1} - 2G(a^j) f_i^j + G(a^{j-1}) f_i^{j-1}}{(a^{j+1} - a^j)(a^j - a^{j-1})} \tag{26}$$

Leapfrog:

$$f_{i+1}^j = f_{i-1}^j - \frac{2\Delta t}{a^{j+1} - a^{j-1}}(G(a^{j+1}) f_i^{j+1} - G(a^j) f_i^{j+1})$$
$$f_1^j = f_0^j - \frac{\Delta t}{a^j - a^{j-1}}(G(a^j) f_0^j - G(a^{j-1}) f_0^{j-1}) \tag{27}$$

The third approach requires performing the variable transformation before and after the timestepping, i.e., compute $\hat{f}_0(a) = G(a) f_0(a)$ at the beginning and compute $f(t, a) = \hat{f}(t, a)/G(a)$ at the end. These steps can be performed offline. If the mesh is suitably constructed as described, the upwind scheme can be efficiently implemented as an algorithm with memory reallocation only. The various schemes for the third approach are implemented as

Upwind:

$$\hat{f}_{i+1}^j = \hat{f}_i^j - \frac{G(a^j)\Delta t}{a^j - a^{j-1}}(\hat{f}_i^j - \hat{f}_i^{j-1}), \tag{28}$$

Lax–Wendroff:

$$\hat{f}_{i+1}^j = \hat{f}_i^j + \left(\frac{(\Delta t)^2}{2} G(a^j)\frac{dG}{da}(a^j) - \Delta t\, G(a^j)\right)\frac{\hat{f}_i^{j+1} - \hat{f}_i^{j-1}}{a^{j+1} - a^{j-1}} + \frac{(\Delta t)^2}{2}(G(a^j))^2 \frac{\hat{f}_i^{j+1} - 2\hat{f}_i^j + \hat{f}_i^{j-1}}{(a^{j+1} - a^j)(a^j - a^{j-1})}, \tag{29}$$

Leapfrog:

$$\hat{f}_{i+1}^j = \hat{f}_{i-1}^j - \frac{2G(a^j)\Delta t}{a^{j+1} - a^{j-1}}(\hat{f}_i^{j+1} - \hat{f}_i^{j-1})$$
$$\hat{f}_1^j = \hat{f}_0^j - \frac{G(a^j)\Delta t}{a^j - a^{j-1}}(\hat{f}_0^j - \hat{f}_0^{j-1}) \tag{30}$$

The second and third approaches do not result in zero error even though CFL = 1 is enforced by virtue of the mesh construction. The local truncation error for the upwind scheme using the third approach is given by

$$\text{Error} = \sum_{n=2}^{\infty}(-1)^n (G(a^j))^{n-1}\frac{(\Delta t)^{n-1}}{n!}\left[\frac{1}{(G(a^j))^{n-2}}\frac{\partial^n \hat{f}}{\partial t^{n-1}\partial a} + (-1)^{n-1}\left(\frac{\partial^n \hat{f}}{\partial t \partial a^{n-1}} + \sum_{k=1}^{n-1} c_k^n \frac{d^k G}{da^k}\frac{\partial^{n-k}\hat{f}}{\partial a^{n-k}}\right)\right], \tag{31}$$

where c_n^k is the binomial coefficient,

$$c_k^n = \frac{(n-1)!}{(n-k-1)!\,k!}, \quad \forall k = 1, 2, \ldots, n-1. \tag{32}$$

As can be seen from (31), the discretization error is first order with the first-order term proportional to dG/da, which is consistent with the zero error observed for constant G. The error expression also demonstrates that the discretization error continuously transitions from first-order accurate to zero error as $d^n G/da^n$ decreases, which suggests that the third approach is well suited for problems where $G(a)$ is only weakly dependent on a.

The fourth approach is slightly more mathematically involved. After performing the first variable transformation used in the third approach, $\tilde{a}(a)$ is computed using (17). Subsequently, $\tilde{a}(a)$ can be inverted either analytically or numerically to construct $a(\tilde{a})$ which can be inserted into $\hat{f}_0(a)$ to yield $\tilde{f}_0(\tilde{a})$. One possible method of performing the inversion numerically is to plot $\tilde{a}(a)$ to obtain pairs $(\tilde{a}_i, a_i)$ at each point which can then be reordered to give $(a_i, \tilde{a}_i)$.[2]

2.2.2. PBMs with growth rate $G(t)$

Consider a homogeneous PBM with a time-varying growth rate in conservative form,

$$\frac{\partial f(t, a)}{\partial t} + \frac{\partial (G(t) f(t, a))}{\partial a} = 0, \quad f(0, a) = f_0(a). \tag{33}$$

[2] Solving (19) exactly using finite differencing on a uniform mesh in $\tilde{a}$ may result in some slight error since the value of Δt may not be a simple fraction that allows the user to obtain a solution at the final time of $t_{\text{end}} = 1.0$ exactly for example.

Being only a function of t, $G(t)$ can be pulled out of the partial derivative to give

$$\frac{\partial f(t,a)}{\partial t} + G(t)\frac{\partial f(t,a)}{\partial a} = 0, \quad f(0,a) = f_0(a), \tag{34}$$

over the real domains $t, a \geq 0$ with $G(t)$ continuous in t and $G(t)$ is positive and bounded from below, i.e., $G(t) \geq M > 0, \forall t$. This PBM is applicable to populations with time-varying driving forces such as growing crystals or precipitates under the condition of varying supersaturation (Gunawan et al., 2004) or growing cells in bioreactors with varying substrate concentrations (Zhu et al., 2000; Quedeville et al., 2018). Introducing the variable transformation

$$\tilde{t} = \int_0^t G(t')dt' \tag{35}$$

and reparameterization

$$\tilde{f}(\tilde{t}, a) = f(t, a) \tag{36}$$

simplifies (34) to

$$\frac{\partial \tilde{f}(\tilde{t},a)}{\partial \tilde{t}} + \frac{\partial \tilde{f}(\tilde{t},a)}{\partial a} = 0, \quad \tilde{f}(0,a) = \tilde{f}_0(a), \tag{37}$$

which again can be solved exactly as above. The function $\tilde{t}(t)$ is invertible following the same arguments outlined in Section 2.2.1. Similar to the case of $G = G(a)$, this model can be solved by three different approaches:

1. Apply finite differences directly on (34) with a fixed Δt.
2. Apply finite differences on (34) while either reconstructing the mesh (selecting a new Δa) or selecting a new value of Δt at each timestep to enforce CFL = 1.
3. Employ the exact method developed.

The first approach is the most straightforward, but can result in significant numerical error and numerical instability during the simulation. The second approach is also comparatively straightforward to implement if non-uniform timestepping is used, as the suitable value of Δt at each timestep can be computed using the CFL condition and applied accordingly.

For the second approach, the upwind scheme results in first-order accuracy, with local truncation error

$$\text{Error} = \sum_{n=2}^{\infty} -\frac{(\Delta t)^{n-1}}{n!}\left[\left(\sum_{k=0}^{n-1} c_k^n \frac{d^k G}{dt^k}\frac{\partial^{n-1} f}{\partial t^{n-k-1}\partial a}\right) + (-1)^{n-1}(G(t))^{n-1}\frac{\partial^n f}{\partial t \partial a^{n-1}}\right]. \tag{38}$$

2.2.3. PBMs with $G(t,a) = G_t(t)G_a(a)$

Consider a homogeneous PBM with a separable time- and size-dependent growth rate in conservative form,

$$\frac{\partial f(t,a)}{\partial t} = \frac{\partial (G_t(t)G_a(a)f(t,a))}{\partial a} = 0, \quad f(0,a) = f_0(a). \tag{39}$$

Many systems are described by this model, as the growth rate can typically be split into an "environmental" component which would be a function of t and a component that is only a function of the intrinsic variable a (Hulburt and Katz, 1964). This model can be solved exactly by employing the variable transformations developed in the previous two cases. By factoring out $G_t(t)$ from the partial derivative with respect to a and defining $\tilde{t} = \int_0^t G_t(t')dt'$ as in Section 2.2.2, (39) is first transformed to

$$\frac{\partial \tilde{f}(\tilde{t},a)}{\partial \tilde{t}} + \frac{\partial (G_a(a)\tilde{f}(\tilde{t},a))}{\partial a} = 0, \quad \tilde{f}(0,a) = \tilde{f}_0(a), \tag{40}$$

which is now identical in form to (15). The model equation can be further transformed by defining $\hat{f}(\tilde{t},a) = G_a(a)\tilde{f}(\tilde{t},a)$ and $\tilde{a} = \int_0^a G_a(a')da'$ as in Section 2.2.1 which gives

$$\frac{\partial \bar{f}(\tilde{t},\tilde{a})}{\partial \tilde{t}} + \frac{\partial \bar{f}(\tilde{t},\tilde{a})}{\partial \tilde{a}} = 0, \quad \bar{f}(0,\tilde{a}) = \bar{f}_0(\tilde{a}), \tag{41}$$

where $\bar{f}$ denotes that $\hat{f}$ has been reparametrized in terms of $\tilde{a}$. Eq. (41) now can easily be solved exactly. In principle, the approaches that can be employed for the above cases can similarly be used here. The error analysis is not derived here as the results look like the concatenation of the results from the previous two sections.

2.3. Nonhomogeneous PBMs with constant growth

Consider the nonhomogeneous PBM,

$$\frac{\partial f(t,a)}{\partial t} + g\frac{\partial f(t,a)}{\partial a} = h(t,a), \quad f(0,a) = f_0(a). \tag{42}$$

While it is not possible to generically employ a variable transformation to (42) to transform it into a form that can be numerically solved exactly, the numerical accuracy can be significantly improved by enforcing CFL = 1. The local truncation error for the upwind scheme is

$$\text{Error} = \sum_{n=2}^{\infty}\left[\frac{1}{n!}\left((\Delta t)^{n-1} - \left(\frac{\Delta a}{g}\right)^{n-1}\right)\frac{\partial^n f}{\partial t^n} + \frac{(\Delta a)^{n-1}}{n!}\left(\sum_{k=0}^{n-1}\frac{(-1)^k}{g^k}\frac{\partial^{n-1}h}{\partial t^k \partial a^{n-k-1}}\right)\right]. \tag{43}$$

By enforcing $\frac{g\Delta t}{\Delta a} = 1$, this error expression simplifies to

$$\text{Error} = \sum_{n=2}^{\infty}\frac{(\Delta a)^{n-1}}{n!}\left(\sum_{k=0}^{n-1}\frac{(-1)^k}{g^k}\frac{\partial^{n-1}h}{\partial t^k \partial a^{n-k-1}}\right). \tag{44}$$

The remaining terms depend on successively higher order derivatives of $h(t,a)$, which results in the error approaching zero as the dependency of h on t and a decreases, similarly as outlined in Section 2.2.1. The three schemes for this class of PBMs are

Upwind:

$$f_{i+1}^j = f_i^j - \frac{g\Delta a}{\Delta t}(f_i^j - f_i^{j-1}) + \Delta t\, h(t_i, a^j). \tag{45}$$

Lax–Wendroff:

$$f_{i+1}^j = f_i^j - \tfrac{1}{2}\tfrac{g\Delta t}{\Delta a}\left(f_i^{j+1} - f_i^{j-1}\right) + \tfrac{1}{2}\left(\tfrac{g\Delta t}{\Delta a}\right)^2\left(f_i^{j+1} - 2f_i^j + f_i^{j-1}\right) + \Delta t\, h\left(t_i, a^j\right) - \tfrac{(\Delta t)^2}{2}\left(g\tfrac{\partial h}{\partial t}\left(t_i, a^j\right) - \tfrac{\partial h}{\partial a}\left(t_i, a^j\right)\right). \tag{46}$$

Leapfrog:

$$f_{i+1}^j = f_{i-1}^j - \frac{g\Delta t}{\Delta a}(f_i^{j+1} - f_i^{j-1}) + 2\Delta t\, h(t_i, a^j),$$
$$f_1^j = f_0^j - \frac{g\Delta t}{\Delta a}(f_0^j - f_0^{j-1}) + \Delta t\, h(t_0, a^j). \tag{47}$$

When CFL = 1 is enforced, the upwind scheme can be implemented efficiently and at low cost, requiring one function evaluation and one multiplication in addition to the memory reallocation at each node:

$$f_{i+1}^j = f_i^{j-1} + \Delta t\, h(t_i, a^j). \tag{48}$$

2.4. PBMs with a linear nonhomogeneous term

A large number of PBMs have a linear nonhomogeneous term such as the von Foerster's equation (Trucco, 1965),

$$\frac{\partial f(t,a)}{\partial t} + \frac{\partial f(t,a)}{\partial a} = -\lambda(t,a)f(t,a), \quad f(0,a) = f_0(a). \tag{49}$$

Table 1
Summary of functional forms of μ.

Form of λ	Form of μ	Functional form of μ
Constant	$\mu(a) \vee \mu(t)$	$e^{\lambda a} \vee e^{\lambda t}$
$\lambda(a)$	$\mu(a)$	$\exp\left(\int_0^a \lambda(a')da'\right)$
$\lambda(t)$	$\mu(t)$	$\exp\left(\int_0^t \lambda(t')dt'\right)$
$\lambda(t,a)$	$\mu(t,a)$	$\exp\left(\int_0^t \lambda(t', t'+a_0)dt'\right)$, $a_0 = a - t$

We seek a variable transform of the form $\hat{f}(t,a) = \mu f(t,a)$ which transforms (49) into

$$\frac{\partial \hat{f}}{\partial t} + \frac{\partial \hat{f}}{\partial a} = 0, \tag{50}$$

which can be solved exactly. Eq. (50) can be expanded to yields a PDE for μ,

$$\frac{1}{\mu}\left(\frac{\partial \mu}{\partial t} + \frac{\partial \mu}{\partial a}\right) = \lambda. \tag{51}$$

A general expression for μ can be derived from (51) by simplifying to an ODE or by using the method of characteristics. A summary of the functional forms of μ can be found in Table 1. Additional details are in the Appendix (Section A5.1).

Three finite difference schemes for this class of PBMs (49) are

Upwind:

$$f_{i+1}^j = f_i^j - \frac{\Delta t}{\Delta a}(f_i^j - f_i^{j-1}) - \Delta t\, \lambda(t_i, a^j) f_i^j. \tag{52}$$

Lax–Wendroff:

$$\begin{aligned} f_{i+1}^j \quad = \quad & f_i^j - \tfrac{1}{2}\tfrac{\Delta t}{\Delta a}\left(f_i^{j+1} - f_i^{j-1}\right) - \Delta t\, \lambda\left(t_i, a^j\right) f_i^j \\ & + \tfrac{1}{2}\left(\tfrac{\Delta t}{\Delta a}\right)^2\left(f_i^{j+1} - 2f_i^j + f_i^{j-1}\right) \\ & + \tfrac{1}{2}(\Delta t)^2\left(\tfrac{\partial \lambda}{\partial a}\left(t_i, a^j\right) - \tfrac{\partial \lambda}{\partial t}\left(t_i, a^j\right)\right) f_i^j \\ & + \tfrac{1}{2}\tfrac{(\Delta t)^2}{\Delta a}\lambda\left(t_i, a^j\right)\left(f_i^{j+1} - f_i^{j-1}\right) \\ & + \tfrac{(\Delta t)^2}{2}\left(\lambda\left(t_i, a^j\right)\right)^2 f_i^j. \end{aligned} \tag{53}$$

Leapfrog:

$$\begin{aligned} f_{i+1}^j &= f_{i-1}^j - \frac{\Delta t}{\Delta a}(f_i^{j+1} - f_i^{j-1}) - 2\Delta t\, \lambda(t_i, a^j) f_i^j, \\ f_1^j &= f_0^j - \frac{\Delta t}{\Delta a}(f_0^j - f_0^{j-1}) - \Delta t\, \lambda(0, a^j) f_0^j. \end{aligned} \tag{54}$$

The exact scheme first computes $\hat{f}(0,a) = \mu(0,a) f_0(a)$, solves the transformed PDE exactly, and then recovers the true solution $f(t,a)$ by computing $f(t,a) = \hat{f}(t,a)/\mu(t,a)$. The first and last steps can be computed offline.

3. Results

This section is also structured as a series of cases whereby each case presents a numerical example for a class of PBMs. The error of various schemes is compared via the Root Mean Square Error,

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - y_{\text{analytical},i})^2}{n}}. \tag{55}$$

3.1. Case 1: PBMs with constant growth rate

Consider a PBM adapted from Gunawan et al. (2004),

$$\frac{\partial f}{\partial t} + 0.1\frac{\partial f}{\partial a} = 0, \quad f_0(a) = 100\exp\left(\frac{-a}{0.01}\right). \tag{56}$$

A no-flux boundary condition is applied at the right-end of the domain and a modified Dirichlet boundary condition at the left-end of the domain which assumes that values of f at the ghost nodes are zero (Gunawan et al., 2004). Exemplar simulation results and the convergence analysis can be seen in Fig. 1.

3.2. Case 2: PBMs with growth rate $G(t,a)$

Consider the PBM adapted from Gunawan et al. (2008),

$$\frac{\partial f}{\partial t} + \frac{\partial (G(a) f)}{\partial a} = 0, \quad f_0(a) = 50\exp\left(\frac{-(a-0.2)^2}{0.0005}\right), \tag{57}$$

with $G(a) = 0.434 + 0.2604a$.

Naive finite differencing ("Con-Uniform" in Fig. 2) results in the lowest accuracy, with the upwind scheme and leapfrog schemes demonstrating sub-first-order convergence. The RMSE from the leapfrog scheme plateaus as a result of numerical dispersion at the right end of the domain, which persists with increasing mesh resolution. It is interesting that enforcing $\frac{\partial (G(a)f)}{\partial a} = 0$ as the boundary condition (Novy et al., 1991) instead of $\frac{\partial f}{\partial a} = 0$ at the right end of the domain removes the numerical dispersion, resulting in the leapfrog scheme performing similarly to the Lax–Wendroff scheme (see Figure A3 in the Appendix).

The second and third approaches ("Con-Nonuniform" and "Trans-Nonuniform") result in nearly identical convergence properties for all the three schemes and is able to perform much better than the naive finite differencing. Typically, it is prudent to formulate and solve the model equation in a way that preserves the underlying physics of the equation instead of transforming the model equation (Ford Versypt and Braatz, 2014). For this particular problem, employing the variable transformation does not impact the accuracy. Interestingly, the Lax–Wendroff scheme performs comparably to the upwind scheme and demonstrates first-order convergence. These results are consistent with a recent study that showed that the Lax–Wendroff scheme for nonuniform grids has first-order or worse performance (Liska et al., 2021). The leapfrog scheme performs the best, displaying second-order convergence. Most notably, the last approach ("Exact") is able to solve the problem to machine precision, demonstrating the capability of the proposed scheme.

3.3. Case 3: Nonhomogeneous PBMs with constant growth

Consider the PBM with a nonhomogeneous term that only depends on a,

$$\frac{\partial f}{\partial t} + 0.1\frac{\partial f}{\partial a} = 1 + 0.1a + 0.1a^2, \quad f_0(a) = 100\exp\left(\frac{-a}{100}\right). \tag{58}$$

The higher accuracy obtained by enforcing $\text{CFL} = 1$ is shown in Fig. 3.

3.4. Case 4: PBMs with a linear nonhomogeneous term

Consider an example of the von Foerster equation,

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial a} = -af, \quad f_0(a) = 50\exp\left(\frac{-(a-0.2)^2}{0.0005}\right) \tag{59}$$

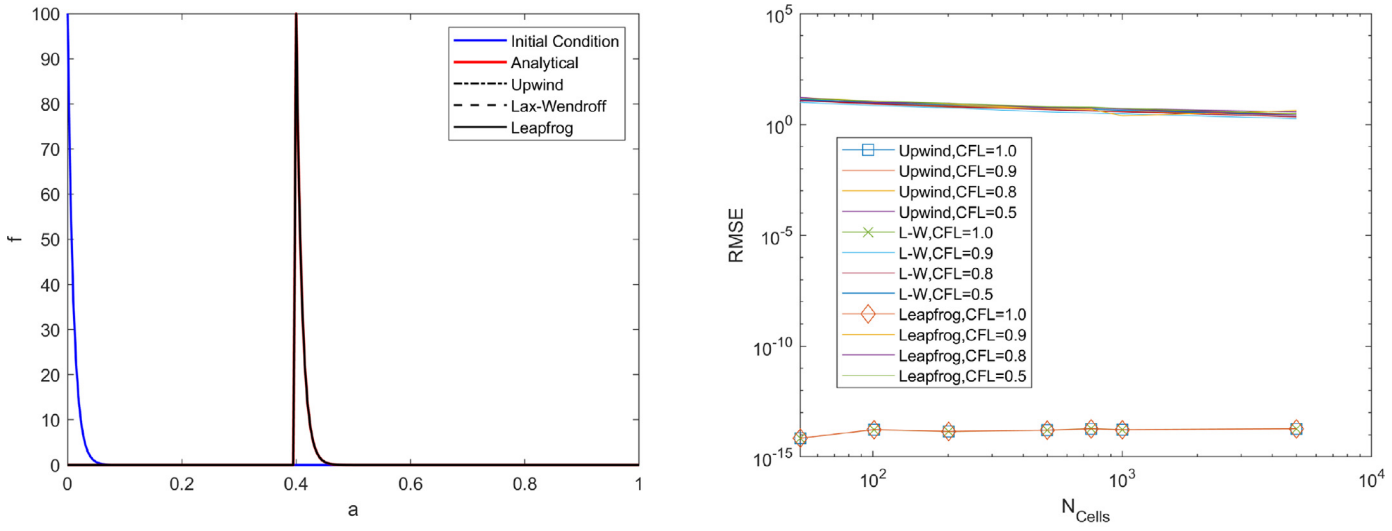
with the boundary condition

$$f(t, a=0) = \int_0^\infty \beta(a) f(t,a) da \equiv B(t), \tag{60}$$

where $\beta(a)$ can be understood as the "rate of birth." For the purposes of this example, $\beta(a) = 0$ is assumed. A Neumann boundary condition is enforced at the other end of the domain. The analytical solution is given by Chou and Greenman (2016)

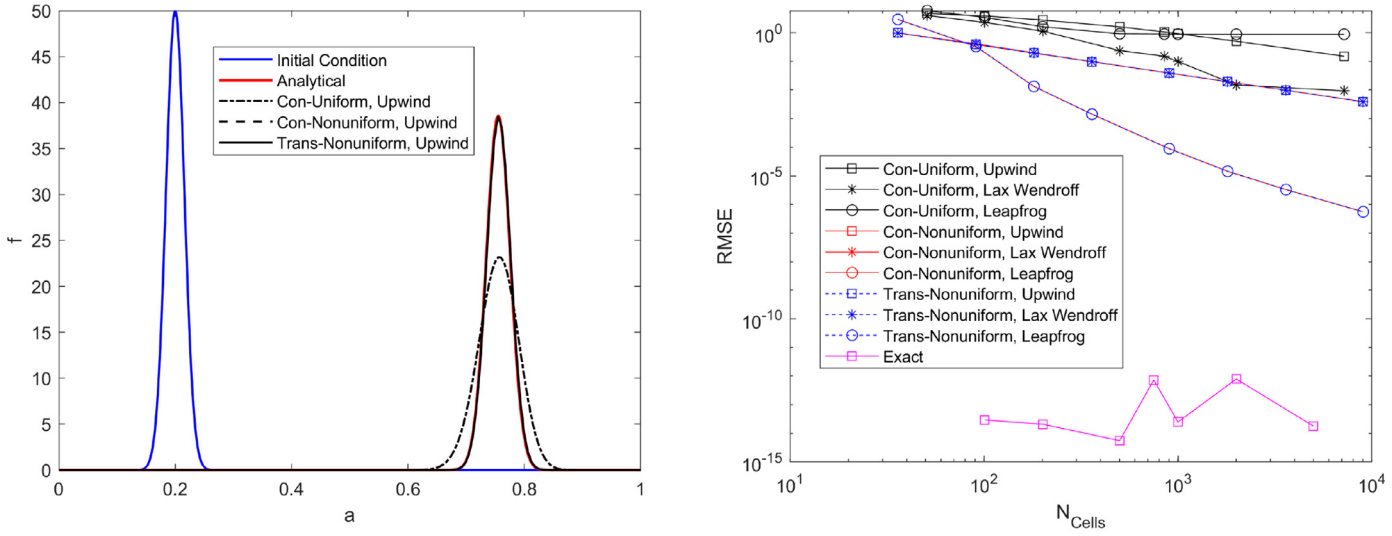
$$f(t,a) = \begin{cases} f_0(a-t)\exp(-\frac{2at-t^2}{2}), & a \geq t, \\ 0, & a < t. \end{cases} \tag{61}$$

The proposed numerical scheme solves the PBM to machine precision, whereas the other schemes do not (Fig. 4).



(a) Simulation (b) RMSE

Fig. 1. Simulation results for Case 1 with CFL = 1 and $N_{cells} = 200$ at time $t = 4.0$. The ability for all three schemes to solve the problem with CFL = 1 to machine precision is indicated by the RMSE plot. Any other CFL value results in significantly higher numerical errors. Numerical results at other CFL numbers can be found in the Appendix (Section A1.2).



(a) Simulation (b) RMSE

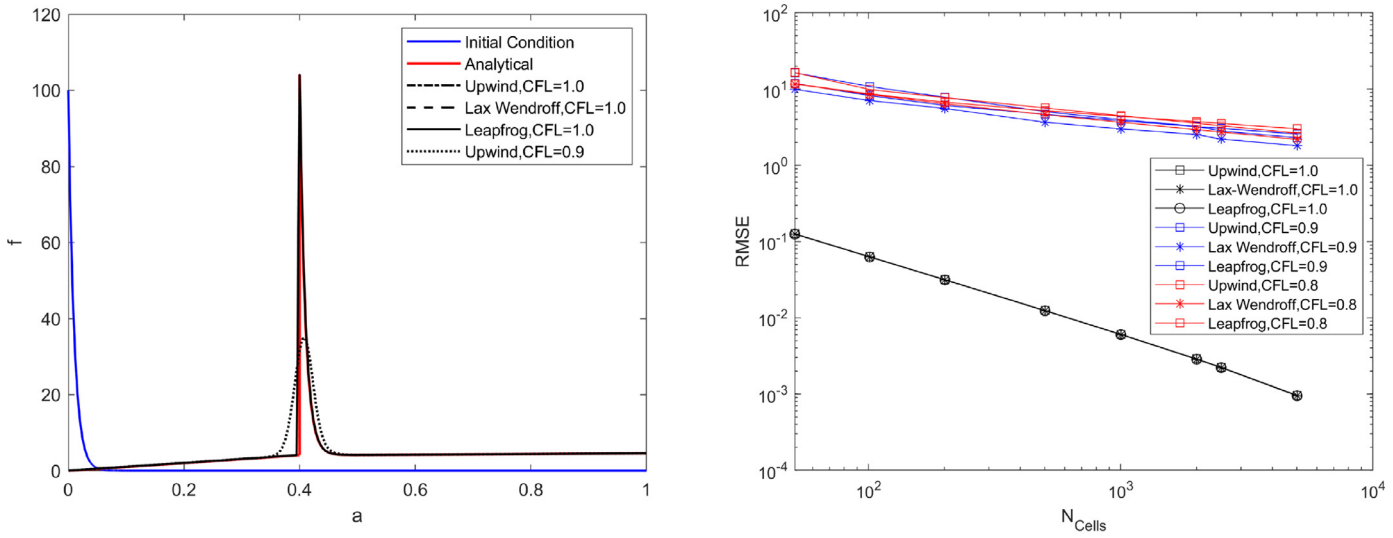
Fig. 2. Simulation results for Case 2. $N_{cells} = 200$ for "Con-Uniform and $N_{cells} = 181$ for "Con-Nonuniform" and "Trans-Nonuniform". The first three approaches result in significantly larger errors than the "Exact" scheme. However, when a non-uniform mesh is applied, even with a comparatively small number of cells, the numerical results have good qualitative agreement with the true solution. Additional results employing the different numerical schemes can be found in the Appendix (Section A2.2).

4. Discussion

The proposed approach of employing the upwind scheme with a combination of variable transformations and solving the equation at the limit of numerical stability (CFL = 1 for the 1D case with the upwind scheme) is able to effectively solve a few classes of PBMs to machine precision or significantly reduce numerical error in other cases. Even where the variable transformation step is avoided, by simply enforcing CFL = 1 as demonstrated by the various case studies, qualitatively satisfactory results can be obtained at low mesh resolutions such as in Case 3. This result is remarkable, as conventional wisdom advises choosing simulation parameters away from the stability limit, e.g., CFL < 1 LeVeque (2002).

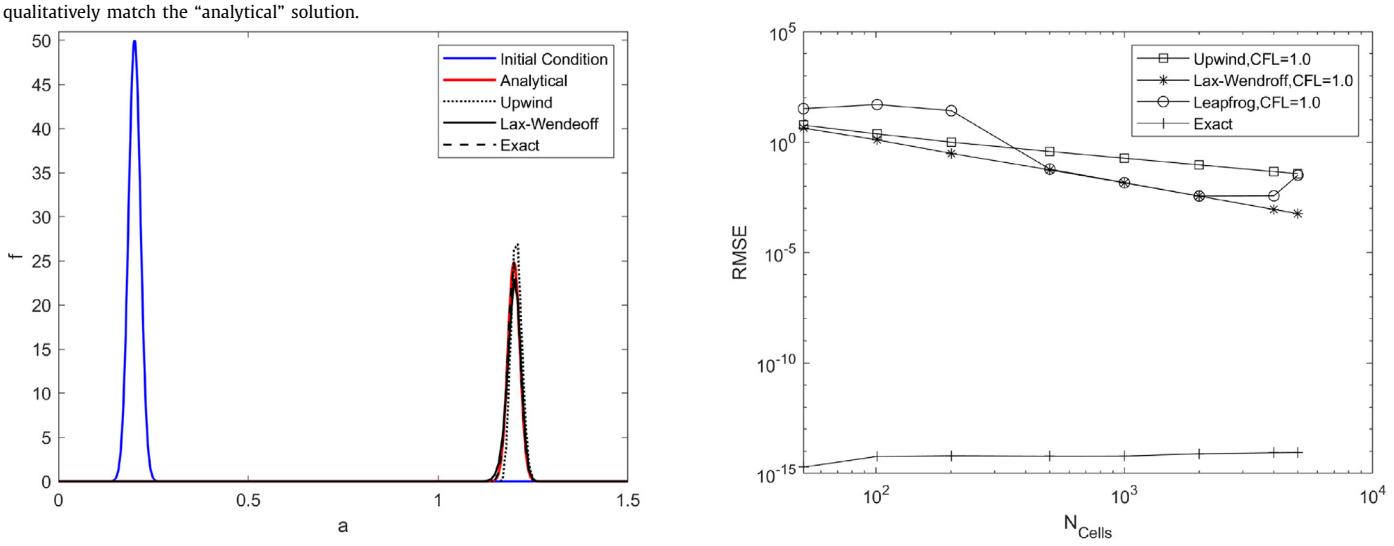
The proposed numerical scheme has two noteworthy benefits. Firstly, highly efficient numerical algorithms can be constructed easily as demonstrated in the Methods section. Where the PBM can be transformed into a form such as (19), the solution process simply requires memory reallocation as steps such as the mesh construction and variable transformation can be computed offline. In other cases, as in the third example, the algorithm would only involve other comparatively cheap steps such as a simple function evaluation and a multiplication. The second benefit is that the proposed scheme can be extended to higher dimensional problems. While not demonstrated in the present work, the construction of an appropriate n-dimensional mesh to solve the problem at CFL_{max} and the definitions for the variable transforms can be carried out using the strategy described in this article.

While not explicitly outlined, the numerical schemes can be extended to varying Δt such as in adaptive timestepping which is useful for the solution of models where the PBM is coupled with



(a) Simulation (b) RMSE

Fig. 3. Simulation results for Case 3. $N_{\text{cells}} = 200$ for all cases. For any fixed CFL value, all the schemes result in comparable numerical error, with all the schemes demonstrating first-order error at CFL = 1 and sub-first-order error for lower values. Even at a low mesh resolution of 200 cells, employing CFL = 1 enables all the schemes to qualitatively match the "analytical" solution.



(a) Simulation (b) RMSE

Fig. 4. Simulation results for Case 4. $N_{\text{cells}} = 200$ with CFL = 1. The upwind and Lax–Wendroff schemes demonstrate the expected first- and second-order convergence respectively and the proposed exact scheme is able to solve the problem to machine precision. The leapfrog scheme demonstrates severe numerical instability which results in non-monotonically decreasing error and is omitted in (a) for clarity. Additional results including results using the leapfrog scheme can be found in the Appendix (Section A5.2).

other equations. The exact schemes can be implemented as function calls to provide the solution to the PBM at any arbitrary time, making the integration of the exact scheme into a solution procedure straightforward. The non-exact schemes can be implemented with varying Δt. The adaptive timestepping should be set up to enforce the CFL condition at each time step to ensure numerical stability. Numerical diffusion and/or dispersion can be reduced by regenerating the mesh[3] at each timestep to keep CFL = 1. The strategies can also be extended to finite volumes. For some classes of population balance models, the error is zero, that is, the cell-averaged values are exact.

In practice, PBM model predictions deviate from experimental data (e.g., see Liu et al., 2020; Lapidot et al., 2019; Zhan et al., 2017). While discretization error from the solution of the PBM can contribute to the overall error, it is often overshadowed by other factors such as parameter uncertainty and inherent limitations of the PBM to fully describe the physics of a system. As such, even in the case where the proposed scheme is unable to solve the model equation to machine precision, the numerical accuracy even with a comparatively low mesh resolution can be satisfactory for the purposes of employing PBMs to describe the process of interest.

5. Conclusions

The upwind scheme, when applied suitably, can be used to solve a variety of PBMs either to machine precision or with suf-

[3] Which may involve interpolation.

ficiently high accuracy even at low mesh resolutions for most applications. The use of variable transformations and specially constructed meshes in conjunction with solving the problem at the limit of numerical stability is what enables the efficacy of the proposed numerical scheme. Even where only parts of the proposed scheme are used such as constructing a non-uniform mesh to locally enforce CFL = 1, significant improvements in the convergence properties of the numerical solution can be observed. Conveniently, the proposed scheme can be implemented very efficiently, requiring memory reallocation and in some cases, a minimal number of floating point operations and function calls only.

Availability of Code

A MATLAB implementation of the various schemes discussed in this work can be found at https://github.com/pavaninguva/PBM_Schemes.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Pavan K. Inguva: Methodology, Software, Validation, Writing – original draft, Writing – review & editing, Visualization. **Kaylee C. Schickel:** Methodology, Writing – original draft. **Richard D. Braatz:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgments

Financial support is acknowledged from the U.S. Food and Drug Administration (75F40121C00090) and the Agency for Science, Technology and Research (A*STAR), Singapore.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compchemeng.2022.107808.

References

Aamir, E., Nagy, Z.K., Rielly, C.D., Kleinert, T., Judat, B., 2009. Combined quadrature method of moments and method of characteristics approach for efficient solution of population balance models for dynamic modeling and crystal size distribution control of crystallization processes. Ind. Eng. Chem. Res. 48 (18), 8575–8584. doi:10.1021/ie900430t.

Abegg, C.F., Stevens, J.D., Larson, M.A., 1968. Crystal size distributions in continuous crystallizers when growth rate is size dependent. AIChE J. 14 (1), 118–122. doi:10.1002/aic.690140121.

Arosio, P., Rima, S., Lattuada, M., Morbidelli, M., 2012. Population balance modeling of antibodies aggregation kinetics. J. Phys. Chem. B 116 (24), 7066–7075. doi:10.1021/jp301091n.

Chou, T., Greenman, C.D., 2016. A hierarchical kinetic theory of birth, death and fission in age-structured interacting populations. J. Stat. Phys. 164 (1), 49–76. doi:10.1007/s10955-016-1524-x.

Colucci, D., Fissore, D., Barresi, A.A., Braatz, R.D., 2020. A new mathematical model for monitoring the temporal evolution of the ice crystal size distribution during freezing in pharmaceutical solutions. Eur. J. Pharm. Biopharm. 148, 148–159. doi:10.1016/j.ejpb.2020.01.004.

Courant, R., Friedrichs, K., Lewy, H., 1928. Uber die partiellen Differenzengleichungen der mathematischen Physik. Mathematische Annalen 100 (1), 32–74. doi:10.1007/BF01448839.

Dalas, E., Kallitsis, J.K., Koutsoukos, P.G., 1991. Crystallization of hydroxyapatite on polymers. Langmuir 7 (8), 1822–1826. doi:10.1021/la00056a040.

Ford Versypt, A.N., Braatz, R.D., 2014. Analysis of finite difference discretization schemes for diffusion in spheres with variable diffusivity. Comput. Chem. Eng. 71, 241–252. doi:10.1016/j.compchemeng.2014.05.022.

Fredrickson, A., Ramkrishna, D., Tsuchiya, H., 1967. Statistics and dynamics of procaryotic cell populations. Math. Biosci. 1 (3), 327–374. doi:10.1016/0025-5564(67)90008-9.

Grön, H., Borissova, A., Roberts, K.J., 2003. In-process ATR-FTIR spectroscopy for closed-loop supersaturation control of a batch crystallizer producing monosodium glutamate crystals of defined size. Ind. Eng. Chem. Res. 42 (1), 198–206. doi:10.1021/ie020346d.

Gunawan, R., Fusman, I., Braatz, R.D., 2004. High resolution algorithms for multidimensional population balance equations. AIChE J. 50 (11), 2738–2749. doi:10.1002/aic.10228.

Gunawan, R., Fusman, I., Braatz, R.D., 2008. Parallel high-resolution finite volume simulation of particulate processes. AIChE J. 54 (6), 1449–1458. doi:10.1002/aic.11484.

Hounslow, M.J., 1990. A discretized population balance for continuous systems at steady state. AIChE J. 36 (1), 106–116. doi:10.1002/aic.690360113.

Hulburt, H., Katz, S., 1964. Some problems in particle technology. Chem. Eng. Sci. 19 (8), 555–574. doi:10.1016/0009-2509(64)85047-8.

Iggland, M., Mazzotti, M., 2012. Population balance modeling with size-dependent solubility: Ostwald ripening. Cryst. Growth Des. 12 (3), 1489–1500. doi:10.1021/cg201571n.

Inguva, P., Bhute, V.J., Cheng, T.N., Walker, P.J., 2021. Introducing students to research codes: a short course on solving partial differential equations in Python. Educ. Chem. Eng. 36, 1–11. doi:10.1016/j.ece.2021.01.011.

Jeldres, R.I., Fawell, P.D., Florio, B.J., 2018. Population balance modelling to describe the particle aggregation process: a review. Powder Technol. 326, 190–207. doi:10.1016/j.powtec.2017.12.033.

Kurtz, M.J., Zhu, G.-Y., Zamamiri, A., Henson, M.A., Hjortsø, M.A., 1998. Control of oscillating microbial cultures described by population balance models. Ind. Eng. Chem. Res. 37 (10), 4059–4070. doi:10.1021/ie9708945.

Lapidot, T., Matar, O.K., Heng, J.Y., 2019. Calcium sulphate crystallisation in the presence of mesoporous silica particles: experiments and population balance modelling. Chem. Eng. Sci. 202, 238–249. doi:10.1016/j.ces.2019.02.045.

LeVeque, R.J., 2002. Finite Volume Methods for Hyperbolic Problems. Cambridge University Press, Cambridge,UK doi:10.1017/CBO9780511791253.

Liska, R., Váchal, P., Wendroff, B., 2021. Lax-Wendroff methods on highly non-uniform meshes. dedicated to the memory of Blair Swartz (1932–2019). Appl. Numer. Math. 163, 167–181. doi:10.1016/j.apnum.2021.01.014.

Liu, Y.C., Acevedo, D., Yang, X., Naimi, S., Wu, W.-L., Pavurala, N., Nagy, Z.K., O'Connor, T.F., 2020. Population balance model development verification and validation of cooling crystallization of carbamazepine. Cryst. Growth Des. 20 (8), 5235–5250. doi:10.1021/acs.cgd.0c00434.

Mesbah, A., Kramer, H.J., Huesman, A.E., Van den Hof, P.M., 2009. A control oriented study on the numerical solution of the population balance equation for crystallization processes. Chem. Eng. Sci. 64 (20), 4262–4277. doi:10.1016/j.ces.2009.06.060.

Novy, A.R., Davis, H., Scriven, L., 1991. A comparison of synthetic boundary conditions for continuous-flow systems. Chem. Eng. Sci. 46 (1), 57–68. doi:10.1016/0009-2509(91)80116-G.

Qamar, S., Elsner, M., Angelov, I., Warnecke, G., Seidel-Morgenstern, A., 2006. A comparative study of high resolution schemes for solving population balances in crystallization. Comput. Chem. Eng. 30 (6–7), 1119–1131. doi:10.1016/j.compchemeng.2006.02.012.

Quedeville, V., Ouazaite, H., Polizzi, B., Fox, R., Villedieu, P., Fede, P., Létisse, F., Morchain, J., 2018. A two-dimensional population balance model for cell growth including multiple uptake systems. Chem. Eng. Res. Des. 132, 966–981. doi:10.1016/j.cherd.2018.02.025.

Schall, C.A., Riley, J.S., Li, E., Arnold, E., Wiencek, J.M., 1996. Application of temperature control strategies to the growth of hen egg-white lysozyme crystals. J. Cryst. Growth 165 (3), 299–307. doi:10.1016/0022-0248(96)00181-9.

Schmidt, S.A., Simon, M., Attarakih, M.M., Lagar, G.L., Bart, H.-J., 2006. Droplet population balance modelling–hydrodynamics and mass transfer. Chem. Eng. Sci. 61 (1), 246–256. doi:10.1016/j.ces.2005.02.075.

Sidorenko, Y., Schulze-Horsel, J., Voigt, A., Reichl, U., Kienle, A., 2008. Stochastic population balance modeling of influenza virus replication in vaccine production processes. Chem. Eng. Sci. 63 (1), 157–169. doi:10.1016/j.ces.2007.09.014.

Szilágyi, B., Muntean, N., Barabás, R., Ponta, O., Lakatos, B.G., 2015. Reaction precipitation of amorphous calcium phosphate: population balance modelling and kinetics. Chem. Eng. Res. Des. 93, 278–286. doi:10.1016/j.cherd.2014.04.003.

Trucco, E., 1965. Mathematical models for cellular systems the von Foerster equation. Part I. Bull. Math. Biophys. 27 (3), 285–304. doi:10.1007/BF02478406.

Wang, T., Wang, J., Jin, Y., 2005. Population balance model for gas-liquid flows: influence of bubble coalescence and breakup models. Ind. Eng. Chem. Res. 44 (19), 7540–7549. doi:10.1021/ie0489002.

Ward, J.D., Mellichamp, D.A., Doherty, M.F., 2006. Choosing an operating policy for seeded batch crystallization. AIChE J. 52 (6), 2046–2054. doi:10.1002/aic.10808.

Westin, K.-J., Rasmuson, Å., 2005. Crystal growth of aragonite and calcite in presence of citric acid, DTPA, EDTA and pyromellitic acid. J. Colloid Interface Sci. 282 (2), 359–369. doi:10.1016/j.jcis.2004.03.029.

Woo, X.Y., Tan, R.B.H., Braatz, R.D., 2011. Precise tailoring of the crystal size distribution by controlled growth and continuous seeding from impinging jet crystallizers. CrystEngComm 13 (6), 2006. doi:10.1039/c0ce00637h.
Woo, X.Y., Tan, R.B.H., Chow, P.S., Braatz, R.D., 2006. Simulation of mixing effects in antisolvent crystallization using a coupled CFD-PDF-PBE approach. Cryst. Growth Des. 6 (6), 1291–1303. doi:10.1021/cg0503090.
Zhan, S., Wang, Z., Yang, J., Zhao, R., Li, C., Wang, J., Zhou, J., 2017. 3D numerical simulations of gas-liquid two-phase flows in aluminum electrolysis cells with the coupled model of computational fluid dynamics-population balance model. Ind. Eng. Chem. Res. 56 (30), 8649–8662. doi:10.1021/acs.iecr.7b01765.
Zhu, G.-Y., Zamamiri, A., Henson, M.A., Hjortsø, M.A., 2000. Model predictive control of continuous yeast bioreactors using cell population balance models. Chem. Eng. Sci. 55 (24), 6155–6167. doi:10.1016/S0009-2509(00)00208-6.