Efficient Numerical Schemes for Nucleation-Aggregation Models: Early Steps

H.T. Banks [*] M. Doumic[†] C. Kruse[‡]

March 20, 2014

Abstract

In the formation of large clusters out of small particles, the initializing step is called the *nucleation*, and consists in the spontaneous reaction of agents which aggregate into small and stable polymers called *nuclei*. After this early step, the polymers are involved into a bunch of reactions such as polymerization, fragmentation and coalescence. Since there may be several orders of magnitude between the size of a particle and the size of an aggregate, building efficient numerical schemes to capture accurately the kinetics of the reaction is a delicate step of key importance. In this article, we propose a conservative scheme, based on finite volume methods on an adaptive grid, which is able to render out the early steps of the reaction as well as the later chain reactions.

Keywords: polymerization, aggregation-fragmentation models, finite volume schemes, adaptive grid.

Introduction

In the formation of large clusters out of small particles, the initializing step is called the *nucleation*, and consists in the spontaneous reaction of agents which aggregate into small and stable polymers called *nuclei*. After this early step, the

[*] Center for Research in Scientific Computation [Raleigh] (CRSC), North Carolina State University; htbanks@ncsu.edu

[†] INRIA Rocquencourt; Laboratoire Jacques-Louis Lions (LJLL), Universit Pierre et Marie Curie (UPMC) - Paris VI; marie.doumic@inria.fr

[‡] INRIA Rocquencourt; Laboratoire Jacques-Louis Lions (LJLL), Universit Pierre et Marie Curie (UPMC) - Paris VI; carola.kruse@inria.fr

polymers are involved into a wide range of possible reactions, such as polymerization, fragmentation and coalescence. These reactions vary from one species to the other, and even from one application field (microtubule or protein polymerization in general) to the other (phase condensation or crystallization).

To model such nucleation and polymerization processes, deterministic models consist in huge systems of ordinary differential equations, where there may be several orders of magnitude between the size of a single agent and the size of an aggregate. In these systems, the concentration of polymers made-up of i monomers, $i \in \mathbb{N}$, is described by a time-dependent variable $c_i(t)$. Its kinetics is given by a first-order differential equation, coupled with the equations of possibly all the other species. Among such models, we can quote the Bekker-Döring system [4], discrete growth-fragmentation models [16] or discrete coagulation-fragmentation models [14].

Continuous coagulation-fragmentation models have then been developed, and proved to be the (weak) limit of the discrete models when an appropriate rescaling is carried out [4, 14, 7, 19]. In such models, the discrete concentration $c_i(t)$ is replaced by a continuous concentration $c(t,x)$ of polymers at time t of size x, whereas the concentration $c_1(t)$ of monomers is treated separately. The limit is taken for a vanishing parameter $\varepsilon := \frac{1}{i_M}$ where i_M is the average size of a polymer. The concentration $c(t,x)$ of polymers is then the solution of a one-dimensional nonlinear first-order integro-partial differential equation on a space $[0,T] \times [x_0, x_M]$ with $0 \leq x_0 < x_M \leq \infty$, and coupled with the equation satisfied by the concentration $c_1(t)$ of monomers.

In these asymptotic results however, the integro-PDE satisfied by $c(t,x)$ requires a boundary condition at $x = x_0 \geq 0$ for the problem to be well-posed. Such a boundary condition is formally derived for different models in [4, 7, 19]. Complete proofs (in a weak formulation) are also provided in [4, 7], but with some extra assumptions either on the parameters (the polymerization rate needs to vanish as one approaches zero, so that a boundary condition is no longer necessary) or on x_0 ($x_0 > 0$ is required). Unfortunately, these restrictive cases are often not physically relevant: $x_0 > 0$ would mean a very large minimal size of stable polymers, since it has to be in the same order of magnitude as the average size $i_M \gg 1$ by assumption, and if we assume a vanishing polymerization rate near 0 then spontaneous formation of polymers from monomers is impossible.

In [19] we proposed a general model to take into account both these large scale phenomena, modeled by a PDE approximation, and the nucleation step, which becomes a boundary condition for the smallest polymers. This boundary condition was formally derived but not theoretically proved, since it fails to satisfy these restrictive assumptions which were used in the previous studies [4, 7]. We also showed on some specific examples the accuracy of this new model. However, it remains to determine how it is possible to capture numerically the specificity of the nucleation step, because its scale is of infinitesimal size compared to the scale where continuous models are valid. This early step is of key importance because it influences the overall dynamics: as shown below, it provides the so-called *lag time*, which is the time needed for the polymerization chain reaction to ignite when initially the solution contains only monomers.

In considering numerical schemes for coagulation-fragmentation models, many successful studies have already been carried out, for the continuous equations (see *e.g.,* [3, 12] for the Lifshitz-Slyozov equation including even a space variable, and [1, 11, 10, 9, 8, 13]) as well as for discrete cases [21, 6]. Our purpose here is not to elaborate on these studies, but rather to focus on the treatment of the nucleation step, which, to the best of our knowledge, has not yet been treated when combined with large chain reactions.

In a first section, we will recall the general model proposed in [19], both in its original ODE version and its approximation by an integro-PDE system. We then write a simplified version of this system, which is the basis of this article: since our point is the treatment of the nucleation step, for the sake of clarity we neglect all the reactions which are of secondary importance when this early step dominates. In a second section, we detail our numerical strategy: the choice of an adapted grid, and convenient finite volume schemes. In a third section we present problems we chose to test our methods - one of them having the main advantage of possessing an analytical solution, which allows quantitative error estimates. We then give our numerical results. Finally, we discuss our results and how to adapt our method to more general situations where secondary pathways need to be considered.

1 Model

1.1 Framework Model

In this preliminary section, we recall the general ODE model that we wish to simulate. This model has been designed to be as general as possible so that any type of reaction is represented. In the remainder of the article, we will call a *monomer* a single particle (or dust or atom or molecule) which is the basic unit agent in the aggregation chain reactions. Its concentration is denoted by $c_1(t)$, whereas a concentration of polymers of size i (assumed here, for the sake of simplicity, to belong to a unique species) is denoted $c_i(t)$. We consider the following reactions.

- Activation scheme. An inert monomer may spontaneously convert into an active conformer, whose concentration is denoted c_1^* with reaction

$$c_1 \underset{k_I^-}{\overset{k_I^+}{\rightleftharpoons}} c_1^*.$$

- Nucleation step. There exist a wide variety of nucleation types - homogeneous or heterogeneous, progressive or not. Here we chose the type of reactions proposed by Oosawa and co-authors [17] in the case of many protein polymerization processes. A nucleus here denotes the smallest stable size of polymers: smaller ones are highly unstable and too transitory to be observed. We call i_0 the size of the nucleus, whose concentration is c_{i_0}. Instead of modelling a sequential addition (represented by $c_1 \to c_2 \to c_3 \to \cdots \to c_{i_0}$), the nucleus formation may be equivalently represented by an i_0 order kinetic reaction, *i.e.* $i_0 c_1^* \to c_{i_0}$. The nucleus size i_0, of unknown value, can be equal to 1, 2, 3 or even more, with reaction given by

$$\underbrace{c_1^* + \cdots + c_1^*}_{i_0} \underset{k_{off}^N}{\overset{k_{on}^N}{\rightleftharpoons}} c_{i_0}.$$

- Chain reaction of polymerization. Polymers of size i quickly polymerize into polymers of size $i+1$ by addition of a monomer at a reaction rate k_{on}^i, and may also depolymerize with a rate k_{dep}^i. This is modeled as

$$c_i + c_1 \underset{k_{dep}^{i+1}}{\overset{k_{on}^i}{\rightleftharpoons}} c_{i+1}.$$

- Coalescence and fragmentation. Polymers can coalesce with one another or break into two smaller polymers. We neglect the breakage into 3 or more pieces, which is generally much more hazardeous, as well as higher order coalescence of 3 or more polymers for the same reason. We denote $k_{col}^{i,j}$ the coagulation rate of two polymers of respective size i and j, and $k_{off}^{i,j}$ the fragmentation rate of a polymer of size i giving rise to smaller polymers of size j and $i-j$, with $2 \leq j \leq i_0$. Thus

$$c_i + c_j \underset{k_{off}^{i+j,i}}{\overset{k_{col}^{i,j}}{\rightleftharpoons}} c_{i+j}.$$

  We define $K_{off}^j = \sum\limits_{i=2}^{j-2} k_{off}^{i,j}$. This represents the total rate with which a polymer of size j can break to give smaller polymers. By symmetry we have that $k_{off}^{i,j} = k_{off}^{j-i,j}$ and $k_{col}^{i,j} = k_{col}^{j,i}$.

- Degradation and monomer addition. Each polymer, conformer or monomer may degrade with a degradation rate k_m^i, k_m^{1*} and k_m^1 respectively, and monomers may be added to the system with a rate $\lambda(t)$.

With these assumptions, the ordinary differential model is given by the sum of the laws of mass action for each of these reactions, namely

$$\frac{dc_1}{dt} = -k_I^+ c_1 + k_I^- c_1^* - k_m^1 c_1 + \lambda(t), \tag{1}$$

$$\begin{aligned} \frac{dc_1^*}{dt} &= k_I^+ c_1 - k_I^- c_1^* - i_0\, k_{on}^N\, (c_1^*)^{i_0} + i_0\, k_{off}^N\, c_{i_0} - k_m^{1*} c_1^* \\ &\quad - c_1^* \sum_{i \geq i_0} k_{on}^i\, c_i + \sum_{j=i_0}^{\infty} k_{dep}^j c_j + 2 \sum_{i=2}^{i_0-1} \sum_{j=i_0}^{\infty} i\, k_{off}^{i,j}\, c_j, \end{aligned} \tag{2}$$

$$\begin{aligned} \frac{dc_{i_0}}{dt} &= k_{on}^N\, (c_1^*)^{i_0} - k_{off}^N\, c_{i_0} - k_{on}^{i_0}\, c_{i_0}\, c_1^* + k_{dep}^{i_0+1} c_{i_0+1} - k_m^{i_0} c_{i_0} \\ &\quad + 2 \sum_{j=i_0+2}^{\infty} k_{off}^{i_0,j}\, c_j - K_{off}^{i_0} c_{i_0} - \sum_{j \geq i_0} k_{col}^{i_0,j}\, c_{i_0}\, c_j, \end{aligned} \tag{3}$$

$$\begin{aligned} \frac{dc_i}{dt} &= c_1^*(k_{on}^{i-1}\, c_{i-1} - k_{on}^i\, c_i) - (k_{dep}^i c_i - k_{dep}^{i+1} c_{i+1}) - k_m^i c_i \\ &\quad + 2 \sum_{j=i+2}^{\infty} k_{off}^{i,j}\, c_j - K_{off}^i c_i + \frac{1}{2} \sum_{i_0 \leq j \leq i-2} k_{col}^{j,i-j} c_j\, c_{i-j} - \sum_{j \geq i_0} k_{col}^{i,j} c_i\, c_j. \end{aligned} \tag{4}$$

When the early steps of nucleation and conformation are absent, this is a classical system of coagulation-fragmentation reactions, which turns out to be the Becker-Döring system if we do not consider either fragmentation and coalescence but only polymerization and depolymerization. This system has a positivity property, and satisfies a mass balance equation of the form

$$\frac{d}{dt}\left(c_1(t) + c_1^*(t) + \sum_{i_0}^{\infty} i c_i(t) \right) = \lambda(t) - k_m^1 c_1(t) - k_m^{1*} c_1^*(t) - \sum_{i_0}^{\infty} i k_m^i c_i(t). \tag{5}$$

1.2 Aim of the Article

The primary objective of this article is to find a fast and accurate numerical scheme to simulate this system. Efficiency is necessary since intensive simulations may be necessary, for instance, to estimate parameters from experimental measurements. Such inverse problem methods and parameter identification algorithms generally require a nontrivial number of simulations. It is also required if we embed this model into a more complex one: for instance if we need a space variable [2, 12], or if we want to model the distribution of polymers in a proliferating cell population [20].

The main difficulty is that we expect i to take values up to 10^6 or even more (for instance in the case of Becker-Döring equation, part of the mass goes to infinity in the super-critical case [18]). This makes an explicit scheme in which each differential equation for c_i must be solved very time-consuming. That is one of the reasons for the interest in a continuous approximation of c_i as was carried out in [19], following previous studies [4, 7, 14].

1.3 Continuous Approximation and Numerical Strategy

The continuous version of this model, formally derived in [19], is summarized next. The notation for c_1 and c_1^* is unchanged, $c(t,x)$ represents the concentration of polymers of size $x \geq x_0 \geq 0$ at time t, and the parameter functions are defined similarly. The continuous variable x replaces the discrete one i. Assumptions that coefficients must satisfy are detailed in [19, Supplementary Data 1]. The system is given by

$$\frac{dc_1}{dt} = -k_I^+ c_1 + k_I^- c_1^* - k_m^1 c_1, \tag{6}$$

$$\begin{aligned}\frac{dc_1^*}{dt} &= k_I^+ c_1 - k_I^- c_1^* - \frac{i_0\, k_{on}^N\, (c_1^*)^{i_0+1} k_{on}^{i_0}}{k_{off}^N + k_{on}^{i_0} c_1^*} \\ &- k_m^{1*} c_1^* - c_1^* \int_{x_0}^{\infty} k_{on}(x)\, c(t,x)dx + \int_{x_0}^{\infty} k_{dep}(x) c(t,x) dx,\end{aligned} \tag{7}$$

$$\begin{aligned}\frac{\partial c(t,x)}{\partial t} &= -c_1^* \frac{\partial}{\partial x}\big(k_{on}(x) c(t,x)\big) + \frac{\partial}{\partial x}\big(k_{dep}(x) c(t,x)\big) \\ &+ 2\int_{x}^{\infty} k_{off}(x,y)\, c(y) dy - K_{off}(x) c(t,x) - k_m(x) c(t,x) \\ &+ \frac{1}{2}\int_{x_0}^{x} k_{col}(y, x-y) c(t,y) c(t,x-y) dy - \int_{x_0}^{\infty} k_{col}(x,y) c(t,x)\, c(t,y) dy, \quad x \geq x_0,\end{aligned} \tag{8}$$

$$k_{on}(x_0) c(t,x_0) = k_{on}(x_0) \frac{k_{on}^N\, (c_1^*)^{i_0}}{k_{off}^N + k_{on}(x_0) c_1}. \tag{9}$$

As soon as the average polymer size i_M is large, this system is expected to be a good approximation of System (1)–(4). A second-order approximation for the polymerizing-depolymerizing terms has been proposed by S. Hariz and J.F. Collet in [5], and is expected to be second-order accurate as shown by the formal calculation carried out in [4] with respect to $\varepsilon = \frac{1}{i_M}$. The problem is that if $i_0 \ll i_M$, what is most often the case, there is *a priori* no reason for this approximation (and even the second order one) to be accurate, since the main assumption, which is the large size of i, fails to be satisfied. Our numerical strategy is thus the following.

Let us set ε as the typical precision that we want to achieve.

- For sizes $i \leq N_0 = 1 + \lfloor \frac{1}{\varepsilon} \rfloor$, we solve the original ODE system described by (1)–(4) by an accurate scheme of the desired order.

- For sizes larger than N_0, we solve the PDE given by (8) by an appropriate finite volume scheme, as described in Section 2, and define a proper approximation of the polymerised mass $\int\limits_{N_0}^{\infty} xc(t,x)dx$. For this step, it is also possible to take advantage of existing schemes, such as developed in [1, 11, 10, 9, 8, 3, 12] for instance.

- We define c_1 by its equation and c_1^* by the mass conservation relation.

This corresponds to solving a mixed ODE and PDE system that we write below in the simplified case on which we will focus.

In order to keep the physical meaning and orders of magnitude, let us note that we did not carry out any dimensionless reformulation of the equations. This leads to large values of x in Equation (8) and to not small values for our space step $\Delta x \geq 1$. The expected precision is not linked to a small Δx but rather to a small ratio $\frac{\Delta x}{x}$, assumed to be in the order of ε. In this case, our PDE approximation is perfectly valid under the same kind of assumptions as in the previous studies [4, 7, 19], for example under the assumption $k_{on}(x) = K_{on}(\varepsilon x)$ with a function $K_{on} \in \mathcal{C}_b^1$ independent of ε. This also means that the larger x, the more we neglect small variations of the coefficients. This is at least correct while nucleation and small polymers dominate the reactions.

1.4 Simplified Model

Since our interest here is to study the nucleation step and how one can build adaptive numerical schemes, for the sake of simplicity we describe our method on a simplified case. This then is meant to be combined with existing numerical schemes for coagulation-fragmentation or the Lifshitz-Slyozov-Wagner equation.

We focus on the case when fragmentation, coalescence, depolymerization and death are not present, and we apply the previously described strategy. The ODE system is then the following

$$\frac{dc_1}{dt} = -k_I^+ c_1 + k_I^- c_1^*, \tag{10}$$

$$\frac{dc_1^*}{dt} = k_I^+ c_1 - k_I^- c_1^* - i_0\, k_{on}^N\, (c_1^*)^{i_0} + i_0\, k_{off}^N\, c_{i_0} - c_1^* \sum_{i \geq i_0} k_{on}^i\, c_i, \tag{11}$$

$$\frac{dc_{i_0}}{dt} = k_{on}^N\, (c_1^*)^{i_0} - k_{off}^N\, c_{i_0} - k_{on}^{i_0}\, c_{i_0}\, c_1^*, \tag{12}$$

$$\frac{dc_i}{dt} = c_1^* (k_{on}^{i-1}\, c_{i-1} - k_{on}^i\, c_i), \tag{13}$$

with the initial conditions

$$c_i(t=0) = c_1^*(t=0) = c_{i_0}(t=0) = 0, \qquad c_1(t=0) = c_0 \in \mathbb{R}. \tag{14}$$

The mass conservation, which may replace either Equation (10) or (11), becomes

$$\frac{d}{dt}\left(c_1(t) + c_1^*(t) + \sum_{i_0}^{\infty} ic_i(t)\right) = 0. \tag{15}$$

2 Numerical Scheme

The domain size in the order of up to a million (shown in experiments as a maximal size for protein polymers, but still larger for other applications like cluster formation) presents a challenge in the computations. Our simplified model begins with an initial concentration of only monomers. After the nucleation step, the polymers bind one monomer at a time. After the usually relevant observation times, smaller polymers are thus found at a higher concentration than larger polymers. A uniform grid, which ideally should not contain a large amount of elements in order to remain computationally tractable, does not capture these peaks at the left-hand side of the polymer distribution efficiently.

As explained in Section 1.3, we approximate (13) by solving the ODE system as long as $i \leq N_0$ and by a PDE for $x \geq N_0$. The system of equations is thus given as

$$\frac{dc_1}{dt} = -k_I^+ c_1 + k_I^- c_1^*, \tag{16}$$

$$\begin{aligned}\frac{dc_1^*}{dt} &= k_I^+ c_1 - k_I^- c_1^* - i_0\, k_{on}^N\, (c_1^*)^{i_0} + i_0\, k_{off}^N\, c_{i_0} \\ &\quad - c_1^* \Bigg(\sum_{i_0}^{N_0} k_{on}^i\, c_i + \int_{N_0}^{\infty} k_{on}(x) c(t,x) dx \Bigg),\end{aligned} \tag{17}$$

$$\frac{dc_{i_0}}{dt} = k_{on}^N\, (c_1^*)^{i_0} - k_{off}^N\, c_{i_0} - k_{on}^{i_0}\, c_{i_0}\, c_1^*, \tag{18}$$

$$\frac{dc_i}{dt} = c_1^* (k_{on}^{i-1} c_{i-1} - k_{on}^i c_i), \qquad i \leq N_0, \tag{19}$$

$$\frac{\partial c}{\partial t} = -c_1^* \partial_x (k_{on}(x) c(x,t)), \qquad x > N_0. \tag{20}$$

with the initial and boundary conditions

$$c_1^*(t=0) = c_{i_0}(t=0) = c_i(t=0) = 0, \quad c_1(t=0) = c_0 \in \mathbb{R}, \tag{21}$$

$$c(t=0,x) = 0, \quad c(t, x = N_0) = c_{N_0}(t). \tag{22}$$

The mass conservation equation becomes

$$\frac{d}{dt} \Bigg(c_1(t) + c_1^*(t) + \sum_{i_0}^{N_0} i c_i + \int_{N_0}^{\infty} x c(t,x) dx \Bigg) = 0. \tag{23}$$

We will now discuss different types of finite volume approximations for the PDE.

2.1 Finite Volume Approximation

We use a finite volume scheme to approximate Equation (20). Let the mesh be defined by $N_0 = x_{1/2} < x_{3/2} < ... < x_{N-1/2} = i_{max}$ with $I_i = [x_{i-1/2},\, x_{i+1/2}]$ and $h_i = x_{i+1/2} - x_{i-1/2}$, not necessarily uniform. We define the cell average on the interval I_i as

$$Q_i^k := \frac{1}{h_i}\int_{x_{i-1/2}}^{x_{i+1/2}} c(x,t_k)\,dx = \frac{1}{h_i}\int_{I_i} c(x,t_k)\,dx. \tag{24}$$

The integral form of (20) on the interval $[t_k,\, t_{k+1}]$ is given by

$$\frac{d}{dt}\int_{I_i} c(x,t)\,dx = f_{i-1/2}(c,c_1^*,t) - f_{i+1/2}(c,c_1^*,t) \tag{25}$$

with $f_{i-1/2}(c,c_1^*,t) = c_1^*(t)k_{on}(x_{i-1/2})c(x_{i-1/2},t)$. By integration, we obtain the time stepping scheme

$$Q_i^{k+1} = Q_i^k - \frac{\Delta t}{h_i}(F_{i+1/2}^k - F_{i-1/2}^k) \tag{26}$$

with

$$F_{i-1/2}^k \approx \frac{1}{\Delta t}\int_{t_k}^{t_{k+1}} f_{i-1/2}(c,c_1^*,t)\,dt \tag{27}$$

being an approximation to the average flux on the interval $[t_k,\, t_{k+1}]$. Choosing $F_{i-1/2}^k = c_1^*(t_k)k_{on}^{i-1/2}Q_{i-1}^k$ with $k_{on}^{i-1/2} := k_{on}(x_{i-1/2})$, we have the simple upwind method

$$Q_i^{k+1} = Q_i^k - \frac{\Delta t}{h_i}c_1^{*,k}(k_{on}^{i+1/2}Q_i^k - k_{on}^{i-1/2}Q_{i-1}^k). \tag{28}$$

This scheme is of first order. To increase the accuracy of the numerical simulations, we add a second order correction term and employ a Flux Limiter method on a non-uniform mesh [15, Chapter 6.17.1]. We have

$$Q_i^{k+1} = Q_i^k - \frac{\Delta t}{h_i}c_1^{*,k}(k_{on}^{i+1/2}Q_i^k - k_{on}^{i-1/2}Q_{i-1}^k) - \frac{\Delta t}{h_i}(\tilde{F}_{i+1/2}^k - \tilde{F}_{i-1/2}^k), \tag{29}$$

where we approximate the correction term by

$$\tilde{F}_{i-1/2}^k = \frac{c_1^{*,k}}{2}\left(h_{i-1} - c_1^{*,k}k_{on}^{i-1/2}\Delta t\right)k_{on}^{i-1/2}\left(\frac{Q_i^k - Q_{i-1}^k}{\frac{1}{2}(h_{i-1}+h_i)}\right)\Phi(\lambda_{i-1}^k) \tag{30}$$

with

$$\lambda_{i-1}^k = \begin{cases} \frac{Q_{i-1}^k - Q_{i-2}^k}{Q_i^k - Q_{i-1}^k}, & Q_i^k \neq Q_{i-1}^k \\ 0, & \text{else} \end{cases}. \tag{31}$$

For $\Phi(\lambda_i^k) = 1$, we obtain the Lax-Wendroff (LW) method which is a classical second order scheme. However, it often leads to oscillations if sharp fronts are present in the solution.
As second choice we use the Van Leer (VL) Limiter given by

$$\Phi(\lambda) = \frac{|\lambda| + \lambda}{1 + |\lambda|} = \left\{ \begin{array}{cc} 0, & \lambda < 0 \\ \frac{2|\lambda|}{1+|\lambda|}, & \lambda > 0 \end{array} \right. . \tag{32}$$

Last, we will use a combination of Beam-Warming and Lax-Wendroff (BWLW) defined through

$$\Phi(\lambda) = \left\{ \begin{array}{cc} 0, & \lambda \leq 0 \\ \lambda, & 0 \leq \lambda \leq 1 \\ 1, & 1 \leq \lambda \end{array} \right. . \tag{33}$$

2.2 Implementation of Boundary conditions

To advance the overall algorithm by one time step, we first compute $c_{i_0}(t_{k+1})$ and solve the finite ODE system c_i^{k+1} for $i = 1, .., N_0$. The computation of Q_1^{n+1} in (26) requires the flux $F_{-1/2}^k$ which is outside the defined problem domain. One approach would be to employ a special formula for the first cell and to compute the flux (27) by numerical integration. This in our case is not possible as $c_1^{*,k+1}$ is necessary, but still unknown (we will discuss the exact algorithm in Section 2.3). As alternative, we use a ghost cell approach as defined in [15]. The main idea is to make use of the solution of the ODE at time t_n. As we have a hyperbolic equation, all information is transported along the streamlines through the domain. We define $x_{-1/2} := N_0 - 1$, $x_{1/2} = N_0$ and define the linear function

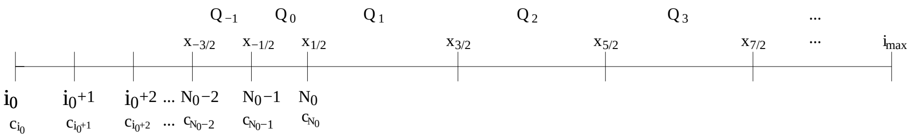


Figure 1: Mesh interpretation for ghost cell approach

$$g(x) = c_{N_0-1} + (x - N_0 + 1)\,(c_{N_0} - c_{N_0-1}) \tag{34}$$

for $x \in [N_0 - 1, N_0]$. We then define

$$Q_0^k := \int_{x_{-1/2}}^{x_{1/2}} g(x)\,dx \tag{35}$$

and consequently obtain (28).

2.3 The algorithm

To obtain $c_1^{*,k}$, we need to compute the total polymerized mass M^k. In case of our ODE-PDE approximation, the total polymerized mass is given by $M^k = M_{ode}^k + M_{pde}^k$ with

$$M_{ode}^k = \sum_{i=i_0}^{N_0} i c_i^k, \qquad M_{pde}^k = \int_{N_0}^{\infty} x c(x, t_k)\, dx.$$

Let now x_i be the midpoint of I_i, i.e. $x_i = \frac{1}{2}(x_{i-1/2} + x_{i+1/2})$. As in [12], we use the approximation

$$M_{pde}(t_k) = \sum_{i=1}^{N} \int_{I_i} x c(x, t_k)\, dx \approx \sum_{i=1}^{N} x_i \int_{I_i} c(x, t_k)\, dx = \sum_{i=1}^{N} x_i h_i Q_i^k.$$

The computational algorithm is thus given by

1. Given $c_1^*(t_k)$, compute $c_{i_0}(t_{k+1})$.
2. Given $c_{i_0}(t_k)$, solve the finite ODE system for c_i^{k+1}, $i = 1, .., N_0$.
3. Given $c_{N_0-1}^k, c_{N_0}^k$, compute a ghost cell average Q_0^k. (Analogously for Q_{-1}^k, Q_{-2}^k in case of the flux limiter methods).
4. Solve the PDE using one of the methods defined in section 2.1 and obtain $c(x_i, t_{k+1})$ for $i = N_0, .., N$
5. Compute $M(t_{k+1})$ and update $c_1^*(t_{k+1})$ with the mass balance equation (23), i.e. $c_1^*(t_{k+1}) = c_1(0) - M_{ode}^{k+1} - M_{pde}^{k+1} - c_1(t_{k+1})$.

Remark 1. *Time discretization.*
In our numerical implementation, the ODE systen (38) for i $\leq N_0$ is solved with the forward Euler method. This scheme is explicit and of first order. To make use of the full higher order convergence that the Lax-Wendroff as well as Flux Limiter methods provide, it is necessary to also employ a second order scheme in time. The difficulty herein lies in the algorithm given above. A classical Runge-Kutta scheme can not be used, as it requires in step 2 the evaluation of $c_1^*(t_{k+1})$ which is still unknown. A remedy is provided by the Adams-Bashforth multi-step method, as it depends only on previous time steps. In our discussion below, we will keep however the backward Euler method. The CFL condition dictates a rather small time step for stability, such that the measured error is mainly spatial and convergence rates become clearly visible.

Remark 2. *Properties of our scheme.*
By replacing (37) with $c_1^*(t) = c_1(0) - M(t) - c_1(t) = c_1(0) - M_{ode}(t) - M_{pde}(t) -$

$c_1(t)$, our scheme is conservative for the mass balance by construction. In case of discretizing the PDE by the upwind scheme, we also have a positive method. On the uniform mesh, the Lax-Wendroff method as defined in (29) is consistent and of second order [15, Chapter 9].

3 Numerical Experiments

3.1 Description of numerical examples

3.1.1 Example 1

As a first example, we neglect the conformation step and choose $c_1^* = a \in \mathbb{R}$, use a constant polymerization rate $k_{on} \in \mathbb{R}$ and set $k_{off} = 0$. The equations (10)-(13) then become

$$\begin{aligned} c_1^* &= a && (36)\\ \frac{dc_{i_0}}{dt} &= k_{on}^N (c_1^*)^{i_0} - k_{on} c_{i_0} c_1^*, && (37)\\ \frac{dc_i}{dt} &= c_1^* k_{on}(c_{i-1} - c_i), && (38)\\ c_i(0) &= 0, \qquad i = i_0, ... \end{aligned}$$

A solution in closed form to this simplified model can be found.

Lemma 1. *For $c_1^* = a \in \mathbb{R}$, $k_{on} \in \mathbb{R}$ and $k_{off} = 0$, we have*

$$c_{i_0} = -\frac{k_{on}^N a^{i_0-1}}{k_{on}} e^{-k_{on}at} + \frac{k_{on}^N a^{i_0-1}}{k_{on}}, \tag{39}$$

For $i > i_0$, the polymer distribution is given by

$$c_{i+1}(t) = c_i(t) - (k_{on}a)^{i-i_0} \frac{k_{on}^N a^{i_0}}{(i+1-i_0)!} t^{i+1-i_0} e^{-k_{on}at}. \tag{40}$$

Having the exact solution provides the possibility to determine a discretization error for the distribution c of our method. However, we chose here to use a representative parameter set, for which the simulation of (40) becomes numerically unstable. When comparing the discretization error in the following section, we will therefore use a numerically computed distribution, obtained by (36), (39) and an explicit very accurate scheme for (38).

The inverse problem uses the total polymerized mass in the cost function, as this is measured in the experiments. In the following, we derive explicit solutions for the total polymerized mass to (36)-(38). Let therefore $P = \sum_{i \geq i_0} c_i$. We add up equations (37) and (38) and use the telescoping sum

$$\begin{aligned}\frac{dP}{dt} &= \frac{dc_{i_0}}{dt} + \sum_{i>i_0} \frac{dc_i}{dt} \\ &= k_{on}^N (c_1^*)^{i_0} - k_{on} c_{i_0} c_1^* + \sum_{i>i_0} c_1^* k_{on} (c_{i-1} - c_i) \\ &= k_{on}^N (c_1^*)^{i_0}.\end{aligned}$$

With $P(0) = 0$, we obtain

$$P(t) = \int_0^t k_{on}^N (c_1^*)^{i_0} \, dt = k_{on}^N a^{i_0} t. \tag{41}$$

Similarly, we obtain the first moment (or total polymerized mass) $M = \sum_{i \geq i_0} i c_i$ by multiplying equations (37) and (38) by i and summing over i

$$\begin{aligned}\frac{dM}{dt} &= i_0 \frac{dc_{i_0}}{dt} + \sum_{i>i_0} k \frac{dc_i}{dt} \\ &= i_0 k_{on}^N (c_1^*)^{i_0} - i_0 k_{on} c_{i_0} c_1^* + \sum_{i>i_0} c_1^* k_{on} ((i-1) c_{i-1} - c_i + c_{i-1}) \\ &= i_0 k_{on}^N (c_1^*)^{i_0} + P c_1^* k_{on}.\end{aligned}$$

Since $M(0) = 0$, we find

$$M(t) = \int_0^t i_0 k_{on}^N (c_1^*)^{i_0} + k_{on} c_1^* P \, ds = i_0 k_{on}^N a^{i_0} t + \frac{k_{on} k_{on}^N a^{i_0+1}}{2} t^2. \tag{42}$$

3.1.2 Example 2

In a second example, we again allow a conformation step and choose the polymerization function k_{on}^i to be linear in i, i.e.,

$$k_{on}^i = k_{on}^{(1)} + i k_{on}^{(2)}$$

for some constants $k_{on}^{(1)}, k_{on}^{(2)} \in \mathbb{R}$. With this choice for k_{on}^i, the setting is a variation of the typical nucleation-aggregation model given by

$$\frac{dc_1}{dt} = -k_I^+ c_1 + k_I^- c_1^*, \tag{43}$$

$$\frac{dc_1^*}{dt} = k_I^+ c_1 - k_I^- c_1^* - i_0 k_{on}^N (c_1^*)^{i_0} + i_0 k_{off}^N c_{i_0} - c_1^* \sum_{i \geq i_0} k_{on}^i c_i, \tag{44}$$

$$\frac{dc_{i_0}}{dt} = k_{on}^N \, (c_1^*)^{i_0} - k_{off}^N \, c_{i_0} - k_{on}^{i_0} \, c_{i_0} \, c_1^*, \tag{45}$$

$$\frac{dc_i}{dt} = c_1^* (k_{on}^{i-1} \, c_{i-1} - k_{on}^i \, c_i). \tag{46}$$

with initial conditions

$$c_i(t=0) = c_1^*(t=0) = c_{i_0}(t=0) = 0, \quad c_1(t=0) = c_0. \tag{47}$$

A solution in closed form cannot be found, but we derive an aggregated version of the model. We follow [19, Supplementary Data 1] and in an analogous manner as above obtain

$$\frac{dP}{dt} = k_{on}^N (c_1^*)^{i_0} - k_{off}^N c_{i_0} \tag{48}$$

and

$$\frac{dM}{dt} = c_1^* k_{on}^{(1)} P + c_1^* k_{on}^{(2)} M + i_0 k_{on}^N (c_1^*)^{i_0} - i_0 k_{off}^N c_{i_0}. \tag{49}$$

Equations (43)-(45) and (48)-(49) form a (finite) system of ODEs which are easily solved at a high precision using an explicit scheme. This numerical solution is then used in Section 3.2 to compute an error for the numerical approximation.

3.2 Numerical Results

We now present numerical results for the finite volume schemes applied to the two examples of the previous section. We make two choices for the mesh required in the discretization of the PDE. First, we use a simple uniform mesh defined as

$$x_i = N_0 + i \cdot h, \; i = 0, .., N$$

with $h = \frac{i_{max} - N_0}{N}$. Second, we use a progressive mesh that is defined such that a ratio between the spatial step size and the corresponding mesh element is kept constant, i.e., $\frac{\Delta x_i}{x_i} = q < 1$. This results in the formula

$$x_i = \frac{1}{1-q} x_{i-1}. \tag{50}$$

Remark 3. *The progressive mesh is a quasi uniform mesh in the sense that* $\frac{\Delta x_{i-1}}{\Delta x_i} = 1 - q = 1 + O(h)$*. With this property, it can be shown that the upwind and Lax-Wendroff methods are consistent on the progressive mesh.*

Based on the parameter estimation problem, where the cost-function uses the total polymerized mass M for the minimization process, we compute the relative $L^2[0,T]$-discretization error e_M as

$$e_M = \frac{\|M^h - M\|_{L^2[0,T]}}{\|M\|_{L^2[0,T]}}$$

with M^h being the solution of the discretized problem. It should be emphasized here that the exact solution M corresponds to the infinite ODE setting, i.e., $M = \sum_{i \geq i_0} i c_i$. We thus compare the numerical solution M^h (discretized by the ODE-PDE scheme) to the ODE solution. The obtained error is therefore always influenced by the quality of the continuous (PDE) approximation of the ODE system.

In case of Example 1, we also compute a relative L^2 error for the distribution c at the final time t_N as

$$e_c(t_N) = \frac{\left(\sum_{i \geq i_0} (\tilde{c}_i(t_N) - c^h(x = i, t_N))^2\right)^{1/2}}{\left(\sum_{i \geq i_0} \tilde{c}_i(t_N)^2\right)^{1/2}}. \tag{51}$$

The approximation to the exact solution c is obtained, as described above, by solving (36), (39) and (38) with $\Delta t_{max} = 1e - 3$. The integer steps of $c^h(x = i, t_N)$ are obtained by linear interpolation between two grid points x_k. Again, we compare the solution of the ODE-PDE scheme to the (numerical) ODE solution. The observed error does not obey any convergence results known from the literature for an approximation of a PDE. However it will give a rough estimate of the convergence properties of our scheme.

To measure the computational efficiency of the schemes, we also include the computation times of each method. These are measured using the Matlab tic-toc command on an Intel Core i7 processor.

3.3 Example 1

We begin by investigating a uniform mesh for the PDE with N elements and compare the flux limiter methods. The parameters are chosen as

$$\begin{aligned} c_0 &= 285 \cdot 10^{-6}, \\ k_{on}^N &= 5.5079 \cdot 10^3, \\ k_{on} &= 2.1691 \cdot 10^6, \\ i_{max} &= 3.2907 \cdot 10^5, \\ i_0 &= 3. \end{aligned}$$

These parameters are derived from a previous paper [19], where similar values were found as best-fit parameters to the polymerization problem and thus represent some typical values. We choose the maximal time step size Δt=1e-3. The smallest time step size is determined by the CFL condition (and thus is unknown a priori). Consequently, we have a fine time stepping and expect to see (for a

Table 1: Ex.1: Error e_M and convergence rates λ_i for the uniform mesh with Δt_{max}=1e-3 and $N_0 = 100$.

N	Upw.	λ_1	LW	λ_2	VL	λ_3	BWLW	λ_4
10	1.7063		1.2790		1.1047		1.2803	
20	0.8531	1.00	0.4753	1.43	0.4384	1.33	0.5379	1.25
40	0.4265	1.00	0.1348	1.82	0.1688	1.38	0.2246	1.26
80	0.2132	1.00	0.0312	2.11	0.0638	1.40	0.0933	1.27
160	0.1065	1.00	0.0077	2.01	0.0238	1.42	0.0385	1.28
320	0.0532	1.00	0.0019	2.04	0.0088	1.43	0.0157	1.29
640	0.0265	1.00	0.0004	2.15	0.0032	1.45	0.0064	1.30
1280	0.0132	1.01	0.0001	2.62	0.0012	1.47	0.0026	1.32
2560	0.0065	1.01	0.0000	0.52	0.0004	1.54	0.0010	1.35
5120	0.0032	1.03	0.0001	-0.44	0.0001	1.77	0.0004	1.43

rather coarse spatial mesh) mainly a spatial error. We compute the simulations up to $T = 40h$. The discretization error e_M is found in Table 1 together with the corresponding convergence rates. These are computed in the usual way as $\lambda_i^k = \frac{\log(e_M(N_k)/e_M(N_{k-1}))}{\log(N_k/N_{k-1})}$ indicating the slope of the error curves in Figure 2. The computation times of each method with corresponding discretization error are presented in Table 2 and the polymer distributions are given in Figure 4.

The simple upwind method converges with a rate of 1, while the Lax-Wendroff method converges with a rate of 2. The Flux-Limiter method rates are between 1 and 2 with the Van-Leer Limiter exhibiting a somewhat better convergence. The uniform mesh does not take the high concentration of smaller polymer sizes into account, but gives each spatial interval equal importance. The exact polymer distribution c_i to Example 1 contains a sharp front. It is thus expected and confirmed in Figure 4 that this feature will not be captured properly. The upwind method on a uniform grid smooths out the sharp front. While the Lax-Wendroff method converges the fastest for M, it leads to large oscillations for the distribution c_i. The flux limiter methods avoid oscillations and approximate the sharp front better than the upwind method. On the other hand, they have a slightly larger error e_M than Lax-Wendroff.

The first experiment for the progressive mesh keeps the ratio q fixed and changes N_0. The error does not change significantly. Due to the constant inflow of conformers and as seen in Figure 3, the distribution c_i is constant for smaller polymer sizes. Any of the four proposed discretization methods approximates a constant accurately. The number of N_0 is thus, for this particular example, not

Table 2: Ex.1: Error e_M for the uniform mesh and computation time ct_i, Δt_{max}=1e-3, $N_0 = 100$

N	Upw.	ct_1	LW	ct_2	VL	ct_3	BWLW	ct_4
10	1.7063	1.58	1.2790	2.93	1.1047	3.50	1.2803	3.69
20	0.8531	1.60	0.4753	3.04	0.4384	3.85	0.5379	3.77
40	0.4265	1.62	0.1348	3.11	0.1688	4.80	0.2246	4.01
80	0.2132	1.74	0.0312	3.23	0.0638	4.21	0.0933	4.29
160	0.1065	2.03	0.0077	3.70	0.0238	4.59	0.0385	4.53
320	0.0532	2.21	0.0019	4.21	0.0088	4.87	0.0157	5.34
640	0.0265	2.73	0.0004	5.02	0.0032	6.18	0.0064	6.72
1280	0.0132	3.92	0.0001	7.37	0.0012	9.00	0.0026	9.85
2560	0.0065	7.74	0.0000	15.04	0.0004	17.91	0.0010	19.99
5120	0.0032	12.74	0.0001	25.54	0.0001	29.79	0.0004	32.75

important. In the following simulations, we set $N_0 = 100$.

The second experiment for the progressive mesh focuses on the convergence of the error in q (or in the corresponding number of elements N). All methods converge satisfactorily (Table 4 and 5 and Figure 3). In terms of the error e_M, the Lax-Wendroff method is best but again exhibits oscillations in the distribution c. In Figure 6, we present the relative error $e_{rel}(t) = \frac{|M^h(t)-M(t)|}{|M|}$. At the beginning, all methods have a large relative error. The model uses an instantaneous inflow of conformers which could be compared to a Dirac-delta function. Since all methods start with solving (38) up to $i = N_0$, this peak is the same for all cases. For $t \geq 1$, the relative error is about constant and thus the approximation M^h is found in a fan-shaped environment around M. It is clearly shown that Lax-Wendroff gives the best approximation to M, while the upwind method performs the worst.

Comparing the two meshes, the error using the progressive mesh is smaller than that using the uniform mesh. In conclusion, a choice for practical simulations would be a flux limiter method to avoid the oscillations in the approximation of c_i in combination with a progressive mesh to make use of the smaller error.

3.4 Example 2

In the second numerical example, we approximate (43)-(46) using (16)-(20). The polymerization function is chosen as $k^i_{on} = k^1_{on} + ik^2_{on}$ for the ODE, and as $k_{on}(x) = k^1_{on} + xk^2_{on}$ in the continuous case. We use the following parameter values

Table 3: Ex.1: Error e_M for the progressive mesh with $\Delta t_{max} = 10^{-3}$

q	N_0	N	BWLW	Upwind	LW	VL
0.10	10	97	0.0554	0.00549	0.0126	0.0199
0.10	100	77	0.0554	0.00549	0.0127	0.0199
0.10	1000	55	0.0553	0.00549	0.0127	0.0199
0.01	10	1009	0.0049	0.00005	0.0003	0.0008
0.01	100	803	0.0049	0.00003	0.0003	0.0008
0.01	1000	577	0.0049	0.00002	0.0003	0.0008

Table 4: Ex.1: Error e_M and convergence rates λ_i for the progressive mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 100$

q	N	Upwind	λ_1	LW	λ_2	VL	λ_3	BWLW	λ_4
1/2	12	0.4995		0.2525		0.2780		0.3329	
1/4	29	0.1665	1.2451	0.0416	2.04	0.0614	1.71	0.0830	1.57
1/8	61	0.0713	1.1409	0.0089	2.08	0.0182	1.63	0.0277	1.48
1/16	126	0.0332	1.0536	0.0020	2.04	0.0060	1.53	0.0102	1.38
1/32	255	0.0160	1.0357	0.0004	2.17	0.0020	1.53	0.0039	1.36
1/64	513	0.0078	1.0266	0.0001	2.91	0.0007	1.56	0.0015	1.37
1/128	1029	0.0038	1.0312	0.0000	0.34	0.0002	1.76	0.0005	1.44
1/256	2062	0.0018	1.0413	0.0001	-0.86	0.0000	3.00	0.0002	1.75

Table 5: Ex.1: Error e_M and computation time ct_i for the progressive mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 100$

q	N	Upwind	ct_1	LW	ct_2	VL	ct_3	BWLW	ct_4
1/2	12	0.4995	1.49	0.2525	2.78	0.2780	3.47	0.3329	3.61
1/4	29	0.1665	1.5157	0.0416	2.90	0.0614	3.50	0.0830	3.68
1/8	61	0.0713	1.8885	0.0089	3.12	0.0182	3.76	0.0277	3.86
1/16	126	0.0332	2.4272	0.0020	4.27	0.0060	5.24	0.0102	5.52
1/32	255	0.0160	1.9706	0.0004	3.64	0.0020	5.05	0.0039	5.99
1/64	513	0.0078	2.4639	0.0001	4.50	0.0007	5.64	0.0015	6.14
1/128	1029	0.0038	3.5800	0.0000	6.82	0.0002	8.48	0.0005	9.51
1/256	2062	0.0018	11.8933	0.0001	22.70	0.0000	27.55	0.0002	31.11

Figure 2: Example 1: Convergence plots for error e_M (uniform mesh)

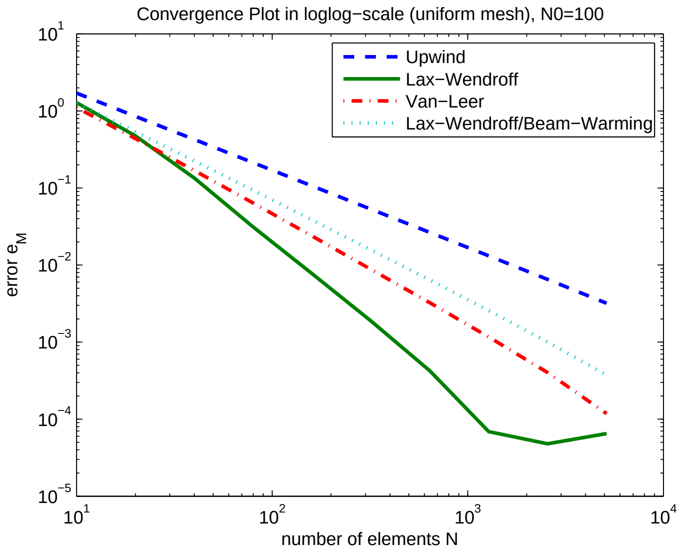


Figure 3: Example 1: Convergence plots for error e_M (progressive mesh)

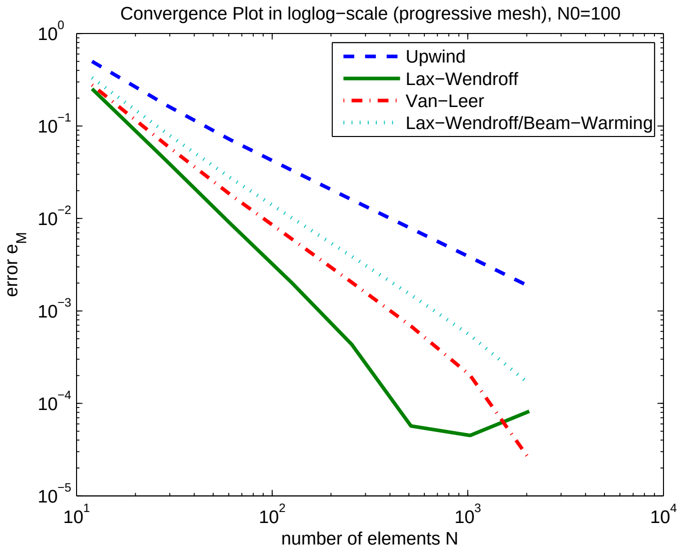


Figure 4: Example 1: Polymer Distribution (uniform mesh)

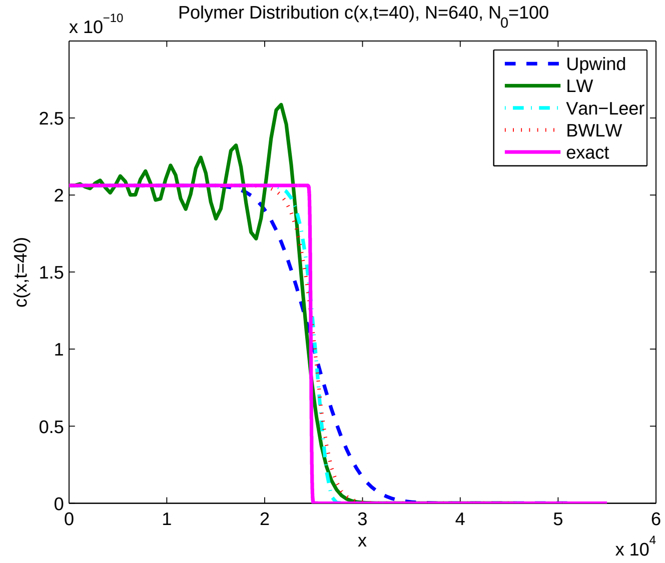


Figure 5: Example 1: Polymer Distribution (progressive mesh)

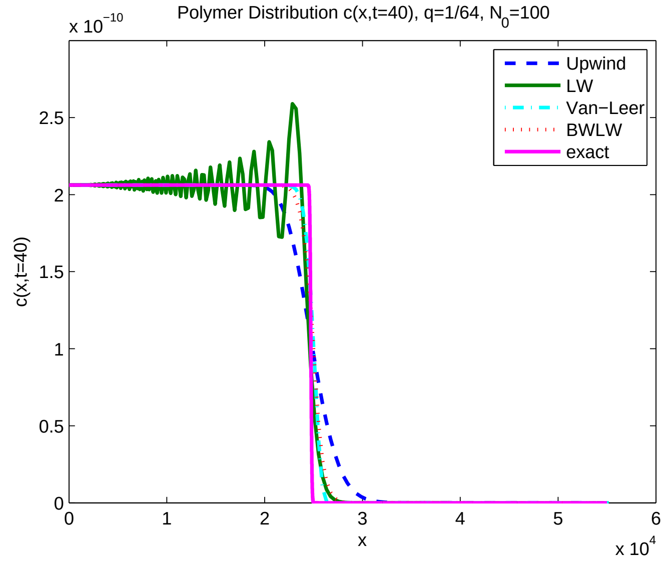


Figure 6: Example 1: Development of error e_{rel} in time (progressive mesh) for q=1/32, $N_0 = 100$ and $\Delta t_{max} = 10^{-3}$

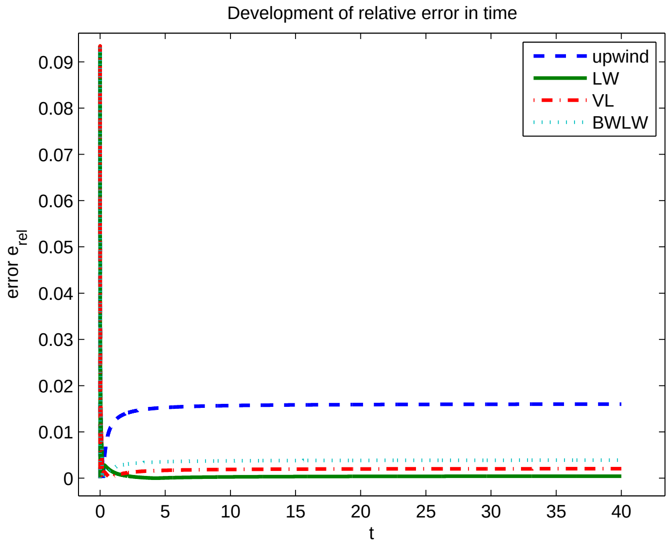


Table 6: Ex.1: Error e_c and convergence rates λ_i for the uniform mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 100$

N	Upwind	λ_1	LW	λ_2	VL	λ_3	BWLW	λ_4
80	3.11e-01		2.35e-01		2.52e-01		2.59e-01	
160	2.58e-01	0.27	1.94e-01	0.27	1.89e-01	0.42	1.98e-01	0.39
320	2.14e-01	0.27	1.62e-01	0.26	1.41e-01	0.42	1.52e-01	0.38
640	1.78e-01	0.27	1.33e-01	0.28	1.04e-01	0.43	1.16e-01	0.39
1280	1.47e-01	0.27	1.08e-01	0.31	7.55e-02	0.47	8.71e-02	0.41
2560	1.21e-01	0.28	8.50e-02	0.34	5.24e-02	0.53	6.36e-02	0.45
5120	9.88e-02	0.29	6.48e-02	0.39	3.32e-02	0.66	4.40e-02	0.53

Table 7: Ex.1: Error e_c and convergence rates λ_i for the progressive mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 100$

q	N	Upwind	λ_1	LW	λ_2	VL	λ_3	BWLW	λ_4
1/8	61	0.2521		0.1931		0.2113		0.2153	
1/16	126	0.2039	0.2927	0.1500	0.35	0.1520	0.45	0.1586	0.42
1/32	255	0.1670	0.2831	0.1204	0.31	0.1093	0.47	0.1176	0.42
1/64	513	0.1371	0.2821	0.0965	0.32	0.0775	0.49	0.0865	0.44
1/128	1029	0.1121	0.2890	0.0758	0.35	0.0528	0.55	0.0620	0.48
1/256	2062	0.0910	0.3010	0.0582	0.38	0.0329	0.68	0.0421	0.56
1/512	4128	0.0727	0.3227	0.0434	0.42	0.0170	0.95	0.0255	0.72

$$\begin{aligned} c_0 &= 285 \cdot 10^{-6}, \\ k_I^+ &= 5.7428 \cdot 10^{-1}, \\ k_I^- &= 1 \cdot 10^{-2}, \\ k_{on}^N &= 5.5079 \cdot 10^3, \\ k_{on}^1 &= 8.276610^3, \\ k_{on}^2 &= 6.5916 \cdot 10^3, \\ i_{max} &= 3.2907 \cdot 10^5, \\ i_0 &= 3. \end{aligned}$$

The simulated curve for the total polymerized mass has a typical shape (Figure 11) for the polymerization-aggregation model. After having a lag-phase at the beginning where the conforming step takes place, it grows steeply (polymerization) and damps out in the end when all monomers are bound to a polymer.

We again discuss the convergence of the different suggested schemes in terms of the total polymerized mass M. This example does not permit an exact solution. We solve the (finite) system of ODEs (43)-(45) and (48)-(49) numerically, which gives a good approximation $\tilde{M}$ to the exact solution. The approximation error is then computed as

$$e_{\tilde{M}} = \frac{\|\tilde{M} - M^h\|_{L^2[0,T]}}{\|\tilde{M}\|_{L^2[0,T]}} \tag{52}$$

We define a maximum time step size to solve the ODE system for $\tilde{M}$, while the minimum time step size is determined through the CFL condition. In the following numerical computations, we use $\Delta t_{max} = 10^{-4}$. For the uniform mesh, we distinguish two different N_0. The errors $e_{\tilde{M}}$ vs. the convergence rates as well as

Table 8: Ex.2: Error $e_{\tilde{M}}$ and convergence rates λ_i for the uniform mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 100$

N	Upwind	λ_1	LW	λ_2	VL	λ_3	BWLW	λ_4
10	0.5400		0.5234		0.4761		0.5234	
20	0.5065	0.09	0.4808	0.12	0.3944	0.27	0.4808	0.12
40	0.4642	0.13	0.4277	0.17	0.2859	0.46	0.4277	0.17
80	0.4130	0.17	0.3649	0.23	0.1796	0.67	0.3650	0.23
160	0.3537	0.22	0.2948	0.31	0.1071	0.75	0.2948	0.31
320	0.2882	0.30	0.2213	0.41	0.0651	0.72	0.2214	0.41
640	0.2207	0.38	0.1504	0.56	0.0389	0.74	0.1505	0.56
1280	0.1568	0.49	0.0888	0.76	0.0209	0.90	0.0891	0.76
2560	0.1023	0.62	0.0432	1.04	0.0094	1.15	0.0439	1.02
5120	0.0615	0.74	0.0164	1.40	0.0034	1.46	0.0173	1.35
10240	0.0345	0.83	0.0044	1.89	0.0010	1.82	0.0057	1.60

computation times for the four different methods using the uniform mesh with $N_0 = 100$ and $N_0 = 500$ are given in Tables 8 - 11. For the progressive mesh, the errors $e_{\tilde{M}}$ and convergence rate are given in Tables 12 - 16. We distinguish between $\Delta t_{max} = 0.5 \cdot 10^{-3}$ and $\Delta t_{max} = 10^{-3}$, as well as $N_0 = 100$ and $N_0 = 500$.

For the uniform mesh and $N_0 = 100$ and $N_0 = 500$, the numerical method has not reached the asymptotic range yet for N, as the convergence rates are still changing. A convergence rate of 1 for the upwind method and 2 for the LW method are likely. We stopped the computations at this point and did not further investigate the convergence rates, as the computation times in combination with the size of the error had already reached an impractical size for our application. It should be noted that the error $e_{\tilde{M}}$ is smaller for $N_0 = 500$ throughout all methods and mesh sizes. This confirms the theory; the longer we use the actual ODE model for smaller polymers, the better is the approximation.

In case of the progressive mesh, the numerical scheme ceases to converge at a certain mesh size. To explain this, we first lowered the maximum time step size, to exclude that the error is caused by the temporal approximation. Having a smaller time step however gives an error in roughly the same range as before. We may conclude that the stagnating error is not caused by the temporal approximation. Recall that the PDE is only an approximation to the infinite ODE system, but the error $e_{\tilde{M}}$ is computed with respect to the (numerical) ODE solution. Increasing N_0 to 500 shows a smaller error and the stagnated error is smaller than in the case of $N_0 = 100$. We thus conclude that we ob-

Table 9: Ex.2: Error $e_{\tilde{M}}$ and computation time ct_i for the uniform mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 100$

N	Upwind	ct_1	LW	ct_2	VL	ct_3	BWLW	ct_4
10	0.5400	0.61	0.5234	0.90	0.4761	1.14	0.5234	1.19
20	0.5065	0.62	0.4808	0.90	0.3944	1.16	0.4808	1.22
40	0.4642	0.62	0.4277	0.93	0.2859	1.18	0.4277	1.23
80	0.4130	0.66	0.3649	1.08	0.1796	1.24	0.3650	1.30
160	0.3537	0.76	0.2948	1.44	0.1071	1.63	0.2948	1.71
320	0.2882	0.89	0.2213	1.78	0.0651	2.06	0.2214	1.76
640	0.2207	1.10	0.1504	1.82	0.0389	2.42	0.1505	2.63
1280	0.1568	2.13	0.0888	4.11	0.0209	5.44	0.0891	5.48
2560	0.1023	8.10	0.0432	16.21	0.0094	21.10	0.0439	21.34
5120	0.0615	24.07	0.0164	50.11	0.0034	62.94	0.0173	73.25
10240	0.0345	104.97	0.0044	198.91	0.0010	234.69	0.0057	267.99

Table 10: Ex.2: Error $e_{\tilde{M}}$ and convergence rates λ_i for the uniform mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 500$

N	Upwind	λ_1	LW	λ_2	VL	λ_3	BWLW	λ_4
10	0.3900		0.3698		0.3178		0.3697	
20	0.3489	0.16	0.3188	0.21	0.2367	0.42	0.3188	0.21
40	0.2992	0.22	0.2586	0.30	0.1515	0.64	0.2586	0.30
80	0.2428	0.30	0.1931	0.42	0.0853	0.83	0.1932	0.42
160	0.1839	0.40	0.1288	0.58	0.0440	0.95	0.1288	0.58
320	0.1283	0.52	0.0736	0.81	0.0205	1.10	0.0736	0.81
640	0.0820	0.65	0.0344	1.10	0.0083	1.32	0.0344	1.10
1280	0.0484	0.76	0.0127	1.44	0.0028	1.55	0.0128	1.42
2560	0.0268	0.85	0.0037	1.76	0.0009	1.71	0.0043	1.58
5120	0.0142	0.92	0.0009	2.09	0.0003	1.66	0.0014	1.61

Table 11: Ex.2: Error $e_{\tilde{M}}$ and computation times ct_i for the uniform mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 500$

N	Upwind	ct_1	LW	ct_2	VL	ct_3	BWLW	ct_4
10	0.3900	0.77	0.3698	1.09	0.3178	1.31	0.3697	1.35
20	0.3489	0.78	0.3188	1.07	0.2367	1.34	0.3188	1.42
40	0.2992	0.79	0.2586	1.13	0.1515	1.37	0.2586	1.43
80	0.2428	0.81	0.1931	1.16	0.0853	1.44	0.1932	1.49
160	0.1839	0.88	0.1288	1.29	0.0440	1.55	0.1288	1.62
320	0.1283	0.94	0.0736	1.43	0.0205	2.02	0.0736	1.90
640	0.0820	1.33	0.0344	1.96	0.0083	2.49	0.0344	2.62
1280	0.0484	2.64	0.0127	4.53	0.0028	6.08	0.0128	7.45
2560	0.0268	9.15	0.0037	17.39	0.0009	20.89	0.0043	22.81
5120	0.0142	27.48	0.0009	52.64	0.0003	64.11	0.0014	72.69

Table 12: Ex.2: Error $e_{\tilde{M}}$ and convergence rates λ_i for the progressive mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 100$

q	N	Upwind	λ_1	LW	λ_2	VL	λ_3	BWLW	λ_4
1/2	12	0.2857		0.2018		0.1253		0.1990	
1/4	29	0.1480	0.75	0.0475	1.64	0.0159	2.34	0.0488	1.59
1/8	61	0.0726	0.96	0.0102	2.07	0.0023	2.58	0.0114	1.95
1/16	126	0.0354	0.99	0.0018	2.38	0.0002	3.24	0.0027	1.99
1/32	255	0.0172	1.02	0.0003	2.54	0.0004	-0.73	0.0005	2.48
1/64	513	0.0083	1.04	0.0007	-1.11	0.0007	-0.82	0.0004	0.12
1/128	1029	0.0039	1.09	0.0008	-0.21	0.0009	-0.39	0.0008	-0.83
1/256	2062	0.0017	1.19	0.0008	-0.05	0.0010	-0.28	0.0008	-0.05

tained a converged solution for the chosen N_0 for which the error to the ODE model can only be diminished by choosing a larger N_0 or a better continuous approximation to the ODE system (e.g., a second order PDE).

The goal of our study was to find an efficient scheme for the nucleation step in terms of accuracy and computation time. Fixing an acceptable computation time of 1.5s, for example, and interpreting $\tilde{M}$ as good approximation to the exact solution, we can make several conclusions from Tables 13 and 16. For $N_0 = 100$ and for the allowed upper temporal bound, the Upwind method achieves a minimal error of $4 \cdot 10^{-3}$, while LW and VL fall below an error of $3 \cdot 10^{-4}$. The VL method performs best with an error of 0.02% in 1.38s. A similar outcome is observed for $N_0 = 500$ in Table 16. Again for a maximal computation time of 1.5 seconds, the VL method obtains the smallest error of

Table 13: Ex.2: Error $e_{\tilde{M}}$ and computational times ct_i for the progressive mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 100$

q	N	Upwind	ct_1	LW	ct_2	VL	ct_3	BWLW	ct_4
1/2	12	0.2857	0.63	0.2018	0.93	0.1253	1.21	0.1990	1.19
1/4	29	0.1480	0.62	0.0475	0.93	0.0159	1.19	0.0488	1.23
1/8	61	0.0726	0.64	0.0102	0.96	0.0023	1.22	0.0114	1.29
1/16	126	0.0354	0.71	0.0018	1.10	0.0002	1.38	0.0027	1.42
1/32	255	0.0172	0.76	0.0003	1.25	0.0004	1.51	0.0005	1.60
1/64	513	0.0083	0.97	0.0007	1.74	0.0007	1.91	0.0004	2.00
1/128	1029	0.0039	1.16	0.0008	2.14	0.0009	2.82	0.0008	3.11
1/256	2062	0.0017	2.41	0.0008	3.71	0.0010	4.66	0.0008	4.99

Table 14: Ex.2: Error $e_{\tilde{M}}$ and convergence rates λ_i for the progressive mesh with $\Delta t_{max} = 0.5 \cdot 10^{-3}$ and $N_0 = 100$

q	N	Upwind	λ_1	LW	λ_2	VL	λ_3	BWLW	λ_4
1/2	12	0.2857		0.2017		0.1252		0.1990	
1/4	29	0.1480	0.75	0.0474	1.64	0.0157	2.35	0.0487	1.59
1/8	61	0.0726	0.96	0.0100	2.09	0.0022	2.67	0.0113	1.96
1/16	126	0.0354	0.99	0.0016	2.49	0.0001	4.41	0.0026	2.04
1/32	255	0.0172	1.02	0.0004	1.96	0.0005	-2.53	0.0003	2.98
1/64	513	0.0083	1.04	0.0008	-1.01	0.0008	-0.67	0.0006	-0.86
1/128	1029	0.0039	1.09	0.0010	-0.18	0.0011	-0.33	0.0010	-0.71
1/256	2062	0.0017	1.19	0.0010	-0.04	0.0012	-0.24	0.0010	-0.04

0.03% in 1.45s (with q=1/16), which is about 8x smaller than the corresponding Upwind error of 0.23% in 1.34s (with q=1/128).

Since no limiter functions have to be computed, the LW method is faster for roughly the same small error. However, due to possible oscillations, the VL method is preferable.

Table 15: Ex.2: Error $e_{\tilde{M}}$ and convergence rates λ_i for the progressive mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 500$

q	N	Upwind	λ_1	LW	λ_2	VL	λ_3	BWLW	λ_4
1/2	10	0.2022		0.1429		0.0901		0.1406	
1/4	23	0.0992	0.85	0.0329	1.76	0.0119	2.43	0.0334	1.73
1/8	49	0.0474	0.98	0.0074	1.98	0.0018	2.50	0.0078	1.92
1/16	101	0.0228	1.01	0.0017	2.05	0.0003	2.45	0.0019	1.97
1/32	205	0.0110	1.03	0.0004	2.17	0.0001	1.32	0.0005	1.99
1/64	412	0.0053	1.06	0.0001	1.37	0.0001	-0.09	0.0001	1.61
1/128	827	0.0024	1.12	0.0002	-0.16	0.0001	-0.15	0.0001	0.21
1/256	1657	0.0010	1.26	0.0002	-0.09	0.0002	-0.14	0.0001	-0.18

Table 16: Ex.2: Error $e_{\tilde{M}}$ and computational times ct_i for the progressive mesh with $\Delta t_{max} = 10^{-3}$ and $N_0 = 500$

q	N	Upwind	ct_1	LW	ct_2	VL	ct_3	BWLW	ct_4
1/2	10	0.2022	0.82	0.1429	1.18	0.0901	1.42	0.1406	1.41
1/4	23	0.0992	0.81	0.0329	1.11	0.0119	1.38	0.0334	1.43
1/8	49	0.0474	0.81	0.0074	1.13	0.0018	1.43	0.0078	1.50
1/16	101	0.0228	0.88	0.0017	1.22	0.0003	1.47	0.0019	1.52
1/32	205	0.0110	0.88	0.0004	1.33	0.0001	1.67	0.0005	1.76
1/64	412	0.0053	1.06	0.0001	1.58	0.0001	2.00	0.0001	2.12
1/128	827	0.0024	1.34	0.0002	2.00	0.0001	2.62	0.0001	2.79
1/256	1657	0.0010	1.78	0.0002	3.41	0.0002	4.13	0.0001	4.23

Figure 7: Example 2: Convergence plots of error $e_{\tilde{M}}$ (uniform mesh), $\Delta t_{max} = 10^{-3}$ and $N_0 = 100$

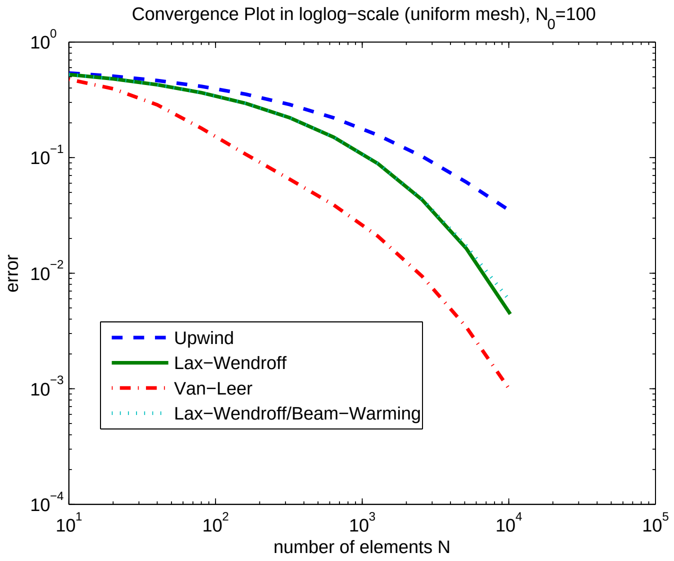


Figure 8: Example 2: Convergence plots of error $e_{\tilde{M}}$ (uniform mesh), $\Delta t_{max} = 10^{-3}$ and $N_0 = 500$

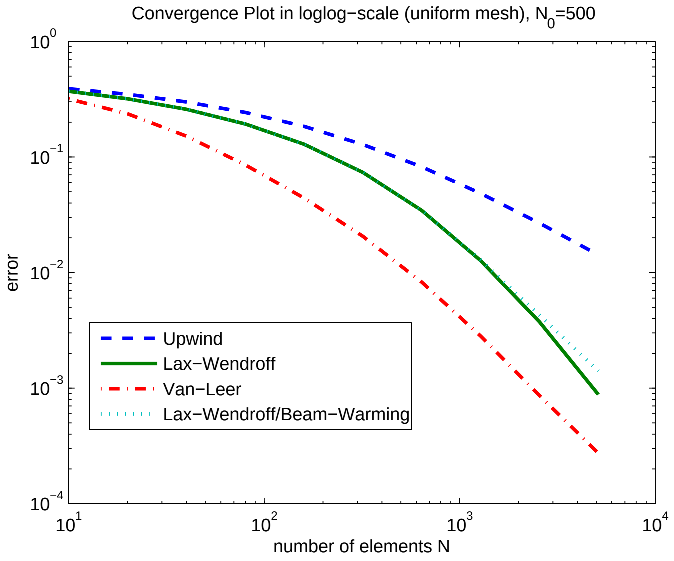


Figure 9: Example 2: Convergence plots of error $e_{\tilde{M}}$ (progressive mesh), $\Delta t_{max} = 10^{-3}$ and $N_0 = 100$

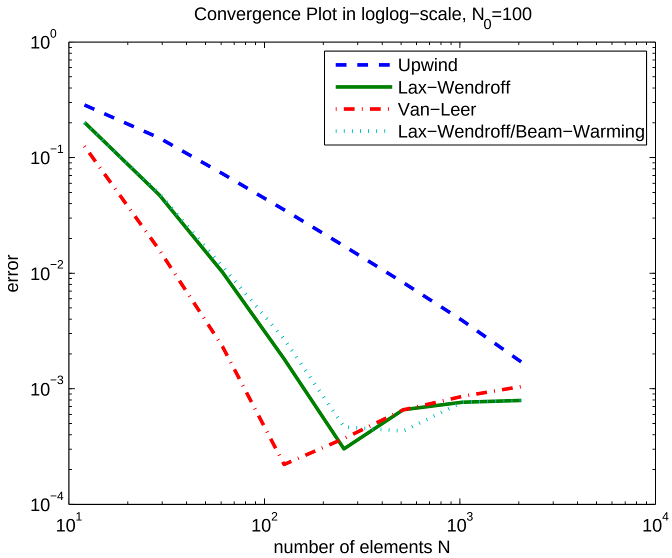


Figure 10: Example 2: Convergence plots of error $e_{\tilde{M}}$ (progressive mesh), $\Delta t_{max} = 10^{-3}$ and $N_0 = 500$

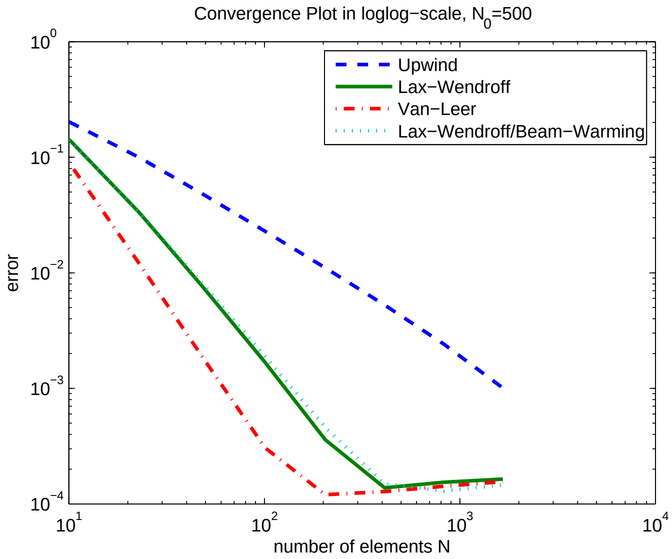
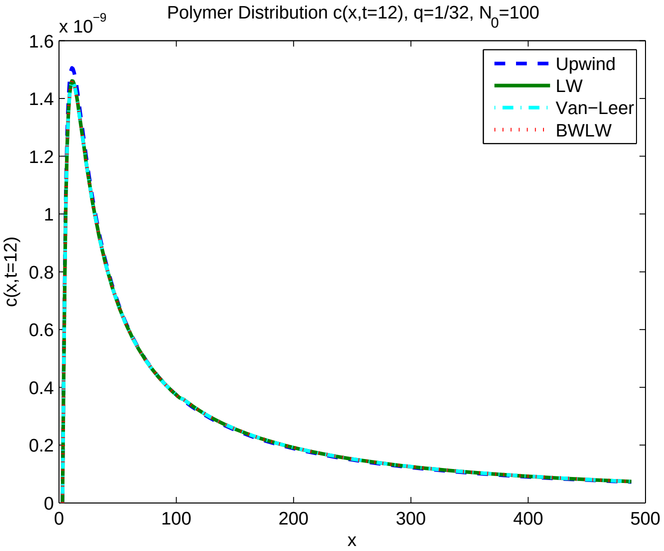


Figure 11: Example 2: Distribution of polymers at $t = 12$



Conclusion

This article proposes a method to deal numerically with both small sizes, predominant during early reaction phases for instance, and with very large aggregates. It is based on a mixed ODE-PDE approach, which keeps the original ODE system for small sizes and uses an approximate PDE, on a progressive grid, for larger sizes. Tested on simplified cases for which explicit solutions are available, the method proved to be accurate, especially when using a flux limiter method in combination with a progressive mesh.

For the PDE component, we used finite volume methods which were accurate for the simplified case we were investigating. The methods presented in this paper neglect the possibility of depolymerization. However, they could be applied equally, by defining the flux limiter method according to [15, Chapter 9.5]

$$F_{i-1/2} = k_{on}^{i-1/2} Q_{i-1} + k_{dep}^{i-1/2} Q_i + \tilde{F}_{i-1/2} \tag{53}$$

where $\tilde{F}_{i-1/2}$ is defined in [15, Chapter 9.3.1, (9.19)].

To adapt our ideas to more general equations or to capture long-time asymptotic behaviors, containing for instance fragmentation or coagulation terms, any other method could be used as soon as it proved efficient to deal with the continuous equation considered on a non-uniform mesh. For instance, the very recent and accurate method developed by T. Goudon, F. Lagoutière and L.M. Tine in [12] for the Lifshitz-Slyozov equation, even if derived on uniform meshes, can be adapted on a non-uniform mesh, and others like [10] are already written on non-uniform grids.

References

1. J.P. Bourgade and F. Filbet. Convergence of a finite volume scheme for coagulation-fragmentation equations. *Math. Comp.*, 77(262):851–882, 2008.

2. J. A. Caizo, L. Desvillettes, and K. Fellner. Regularity and mass conservation for discrete coagulation-fragmentation equations with diffusion. *Annales de lInstitut Henri Poincare. Annales: Analyse Non Lineaire/Nonlinear Analysis*, 27(2):639–654, 2010.

3. J. A. Carrillo and T. Goudon. A numerical study on large-time asymptotics of the Lifshitz-Slyozov system. *J. Sci. Comput.*, 20(1):69–113, 2004.

4. J.F. Collet, T. Goudon, F. Poupaud, and A. Vasseur. The Becker–Döring system and its Lifshitz–Slyozov limit. *SIAM J. on Appl. Math.*, 62(5):1488–1500, 2002.

5. J.F. Collet and S. Hariz. A modified version of the Lifshitz-Slyozov model. *Applied Mathematics Letters*, 12:81–95, 1999.

6. P Deuflhard, W Huisinga, T Jahnke, and M Wulkow. Adaptive discrete Galerkin methods applied to the chemical master equation. *Technology*, 30(6):2990–3011, 2008.

7. M. Doumic, T. Goudon, and T. Lepoutre. Scaling limit of a discrete prion dynamics model. *Comm. in Math. Sc.*, 7(4):839–865, 2009.

8. F. Filbet and P. Laurençot. Numerical approximation of the Lifshitz-Slyozov-Wagner equation. *SIAM J. Numer. Anal.*, 41(2):563–588 (electronic), 2003.

9. F. Filbet and P. Laurençot. Mass-conserving solutions and non-conservative approximation to the Smoluchowski coagulation equation. *Arch. Math. (Basel)*, 83(6):558–567, 2004.

10. F. Filbet and P. Laurençot. Numerical simulation of the Smoluchowski coagulation equation. *SIAM J. Sci. Comput.*, 25(6):2004–2028 (electronic), 2004.

11. Francis Filbet. An asymptotically stable scheme for diffusive coagulation-fragmentation models. *Commun. Math. Sci.*, 6(2):257–280, 2008.

12. T. Goudon, F. Lagoutière, and L.M. Tine. The Lifschitz-Slyozov equation with space-diffusion of monomers. *Kinetic and Related Models*, 5(2):325–355, 06 2012.

13. J. Kumar, M. Peglow, G. Warnecke, S. Heinrich, and L. Mrl. Improved accuracy and convergence of discretized population balance for aggregation: The cell average technique. *Chemical Engineering Science*, 61(10):3327 – 3342, 2006.

14. Laurençot, P., and S. Mischler. From the discrete to the continuous coagulation-fragmentation equations. *Proc. Roy. Soc. Edinburgh Sect. A*, 132(5):1219–1248, 2002.

15. Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2002.

16. J. Masel, V.A.A. Jansen, and M.A. Nowak. Quantifying the kinetic parameters of prion replication. *Biophysical Chemistry*, 77(2-3):139 – 152, 1999.

17. F. Oosawa and S. Asakura. *Thermodynamics of the Polymerization of Protein*. Academic Press, 1975.

18. Oliver Penrose. Metastable states for the becker-dring cluster equations. *Communications in Mathematical Physics*, 124(4):515–541, 1989.

19. S. Prigent, A. Ballesta, F. Charles, N. Lenuzza, P. Gabriel, L.M. Tine, H. Rezaei, and M. Doumic. An efficient kinetic model for assemblies of amyloid fibrils and its application to polyglutamine aggregation. *PLoS ONE*, 7(11):e43273, 11 2012.

20. Suzanne S. Sindi and Tricia R. Serio. Prion dynamics and the quest for the genetic determinant in protein-only inheritance. *Current Opinion in Microbiology*, 12(6):623–630, 2009.

21. M. Wulkow. The simulation of molecular weight distributions in polyreaction kinetics by discrete Galerkin methods. *Macromol. Theory Simul.*, 5:396–416, 1996.