

MULTISCALE REACTION-DIFFUSION ALGORITHMS: PDE-ASSISTED BROWNIAN DYNAMICS*

BENJAMIN FRANZ[†], MARK B. FLEGG[†], S. JONATHAN CHAPMAN[†], AND
RADEK ERBAN[‡]

Abstract. Two algorithms that combine Brownian dynamics (BD) simulations with mean-field partial differential equations (PDEs) are presented. This PDE-assisted Brownian dynamics (PBD) methodology provides exact particle tracking data in parts of the domain, whilst making use of a mean-field reaction-diffusion PDE description elsewhere. The first PBD algorithm couples BD simulations with PDEs by randomly creating new particles close to the interface, which partitions the domain, and by reincorporating particles into the continuum PDE-description when they cross the interface. The second PBD algorithm introduces an overlap region, where both descriptions exist in parallel. It is shown that the overlap region is required to accurately compute variances using PBD simulations. Advantages of both PBD approaches are discussed and illustrative numerical examples are presented.

Key words. reaction-diffusion systems, Brownian dynamics, multiscale simulation

AMS subject classifications. 35R60, 60J65, 92C40

DOI. 10.1137/120882469

1. Introduction. Spatial reaction-diffusion models have been widely used for the description of biological systems [25]. Often continuum approaches, written in the form of reaction-diffusion partial differential equations (PDEs), are used due to their simplicity and the availability of ready-to-use numerical solvers. However, many biological effects cannot be fully described by deterministic PDE-based models. This is because a deterministic model requires large copy numbers of molecules to minimize the relative fluctuation of the spatial concentration. If low copy numbers are present in a biological system [22, 26], then stochastic models, such as mesoscopic compartment-based algorithms [19, 6] or trajectory tracking (Brownian dynamics) methods, should be deployed [2, 29].

In many situations individual trajectories are important only in certain parts of the domain, whilst in the remainder of the domain a coarser, less detailed, method can be used [15]. This is the case, for example, in the modeling of ion-channels [24]. Ions pass through a channel in single file and an individual-based model has to be used to accurately compute the discrete, stochastic, current in the channel [4]. The positions of individual ions are less important away from the channel where copy numbers may be very large (rendering a detailed Brownian dynamics (BD) description infeasible) [5]. Another example is the stochastic reaction-diffusion modeling of filopodia, which

*Received by the editors June 26, 2012; accepted for publication (in revised form) March 6, 2013; published electronically June 19, 2013. The research leading to these results has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement 239870. This publication was based on work supported in part by Award KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST).

<http://www.siam.org/journals/siap/73-3/88246.html>

[†]Mathematical Institute, University of Oxford, Oxford OX1 3LB, UK (franz@maths.ox.ac.uk, flegg@maths.ox.ac.uk, chapman@maths.ox.ac.uk).

[‡]Corresponding author. Mathematical Institute, University of Oxford, Oxford OX1 3LB, UK (erban@maths.ox.ac.uk). This author's work was supported by a Royal Society University Research Fellowship; by a Nicholas Kurti Junior Fellowship of Brasenose College, University of Oxford; and by a Philip Leverhulme Prize awarded by the Leverhulme Trust.

are dynamic finger-like protrusions used by eukaryotic motile cells to probe their environment and help guide cell motility [33]. These relatively small protrusions are connected to a larger cytosol compartment. If a modeler is interested in understanding the dynamics of filopodia, then there is a potential to decrease the computational cost of simulations by using a coarser model in the cytosol. In both examples, it is important to understand how models with a different level of detail can be used in different parts of the computational domain [15].

In this paper, we develop algorithms that calculate BD paths in a desired part of the domain, whilst using a continuum PDE-based model in the remainder. This PDE-assisted Brownian dynamics (PBD) methodology has the advantage that efficient methods for solving PDEs can be used for large parts of the modeled domain, whilst BD data is available in other areas where required. The main goal of the PBD methodology is to get the same statistics (means and variances) in the BD subdomain as we would get if we were able to use BD simulations in the whole domain. In particular, the correct coupling between the two parts of the domain is of vital importance for the accuracy of a PBD algorithm.

The paper is organized as follows. Section 2 states, in mathematical terms, the requirements for the developed algorithms and introduces the notation used throughout the remainder of the paper. We then introduce the first PBD algorithm for a pure diffusion system in section 3, where we also explore the complications of this algorithm. In section 4 we present the second PBD algorithm which provides more accurate computations. In section 5 we investigate issues relating to the introduction of reactions into the system and present several computational examples.

2. Problem formulation. Consider a general Brownian dynamics reaction-diffusion simulation with M chemical species in the (open) domain $\Omega \subset \mathbb{R}^3$. Let $N_j(t)$, $j = 1, 2, \dots, M$, be the number of molecules of the j th chemical species in the system at time t . Let us denote by $\mathbf{x}_i^{(j)}(t)$, $i = 1, 2, \dots, N_j(t)$, the positions of molecules of the j th chemical species at time t . Brownian dynamics can be used for every subset $A \subset \Omega$ to evaluate the random variable

$$\varphi_j(A, t) = \left| \left\{ \mathbf{x}_i^{(j)}(t) \in A, i = 1, 2, \dots, N_j(t) \right\} \right|,$$

which is the number of molecules of the j th chemical species inside the subset A at time t . We denote by $n_j(\mathbf{x}, t)$, $j = 1, 2, \dots, M$, the expected spatio-temporal concentration of the j th chemical species at the position \mathbf{x} and time t over our domain Ω . Then we have

$$\mathbb{E}[\varphi_j(A, t)] = \int_A n_j(\mathbf{x}, t) dx.$$

We also define the expected spatio-temporal standard deviation of the number of molecules of the j th chemical species inside $A \subset \Omega$ by

$$(2.1) \quad \sigma_j(A, t) = \sqrt{\text{var}[\varphi_j(A, t)]} = \sqrt{\mathbb{E}[(\varphi_j(A, t))^2] - \left(\int_A n_j(\mathbf{x}, t) dx\right)^2}.$$

The approximate mean-field reaction-diffusion PDEs for the time evolution of concentrations can be written as follows:

$$(2.2) \quad \frac{\partial p_j}{\partial t} = D_j \Delta p_j + R_j(p_1, p_2, \dots, p_M), \quad j = 1, 2, \dots, M,$$

where $p_j \equiv p_j(\mathbf{x}, t) : \Omega \times [0, \infty) \rightarrow [0, \infty)$ is the mean-field approximation of n_j , D_j is the diffusion constant of the j th chemical species, and $R_j : [0, \infty)^M \rightarrow \mathbb{R}$ represents the reaction terms.

The goal of PBD algorithms is to couple the macroscopic description (2.2) in the open subdomain $\Omega_P \subset \Omega$ with a stochastic BD simulation in the open subdomain $\Omega_B \subset \Omega$, where the closures of Ω_B and Ω_P cover Ω , i.e.,

$$(2.3) \quad \Omega \subset \overline{\Omega_B} \cup \overline{\Omega_P}.$$

In Ω_B , we will consider BD trajectories of individual molecules, i.e., the state of the microscopic subdomain Ω_B is defined by the number $N_B^{(j)}(t)$ of molecules of the j th chemical species at time t and their positions $\mathbf{x}_i^{(j)}(t) \in \Omega_B$, $i = 1, 2, \dots, N_B^{(j)}(t)$, $j = 1, 2, \dots, M$. We will investigate two cases:

[A] Ω_B and Ω_P do not overlap, i.e., $\Omega_B \cap \Omega_P = \emptyset$, with interface between the subdomains Ω_B and Ω_P denoted by

$$(2.4) \quad I = \overline{\Omega_B} \cap \overline{\Omega_P};$$

[B] there exists an overlap region O where the PDE description and BD simulations exist in parallel, i.e., $O = \Omega_B \cap \Omega_P \neq \emptyset$.

Case [A] will lead to the PBD algorithm (A1)–(A5) presented in Table 3.1. Case [B] is implemented in the second PBD algorithm (B1)–(B5), which is presented in Table 4.1. We will start our discussion with case [A] because it is less technical to implement than case [B].

To simplify our presentation, we will consider that Ω is a “narrow” three-dimensional domain, and hence consider only the process mapped onto an effective one-dimensional domain $\Omega \subset \mathbb{R}$ by assuming that the system is well mixed in the other two dimensions. In particular, we have $\Omega_B \subset \Omega \subset \mathbb{R}$ and the state of the BD subdomain Ω_B will be described by the x -coordinates of molecules which we will denote as $x_i^{(j)}(t) \in \Omega_B$, $i = 1, 2, \dots, N_B^{(j)}(t)$, $j = 1, 2, \dots, M$. We simulate the system using finite time step $\Delta t > 0$, in which particles in Ω_B change their position according to the discretized version of the overdamped Langevin equation

$$(2.5) \quad x_i^{(j)}(t + \Delta t) = x_i^{(j)}(t) + \sqrt{2D_j \Delta t} \xi,$$

where ξ is a normally distributed random variable with zero mean and unit variance. Figure 2.1 shows a sketch of the described system for one chemical species along with the notation used in the case [A]. Our goal is to construct PBD algorithms which will satisfy the following two conditions.

Condition (C.1). We require that the expected distribution of molecules in $\Omega_B \setminus \Omega_P$ match that of the expected distribution $n_j(x, t)$, i.e., the distribution which we would obtain if we used detailed BD simulations in the whole domain Ω . In particular, this means that for every set $A \subset \Omega_B \setminus \Omega_P$, the expected number of particles in A at time $t > 0$ has to satisfy

$$(2.6) \quad \mathbb{E} \left[\left| \left\{ x_i^{(j)}(t) \in A, i = 1, \dots, N_B^{(j)}(t) \right\} \right| \right] = \int_A n_j(x, t) dx \quad \forall A \subset \Omega_B \setminus \Omega_P.$$

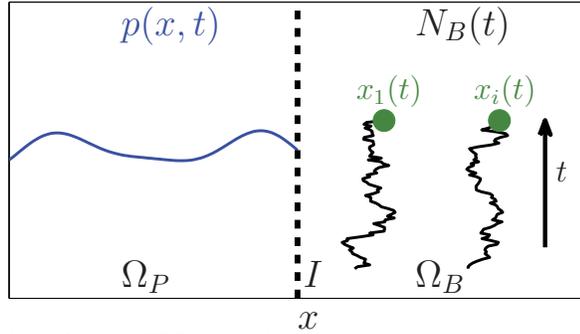


FIG. 2.1. Sketch of the first PBD algorithm and the notation related to it. In Ω_P , molecules are described by their density distribution $p(x, t)$ and in the microscopic domain Ω_B are described by the number $N_B(t)$ of molecules and their positions $x_i(t)$, $i = 1, 2, \dots, N_B(t)$. The interface between these domains is denoted I .

The choice of an arbitrarily small but finite interval $A = [x, x + dx)$ for $x \in \Omega_B \setminus \Omega_P$ leads to an alternative formulation of this condition,

$$(2.7) \quad \mathbb{E} \left[\left| \left\{ x_i^{(j)}(t) \in [x, x + dx), i = 1, \dots, N_B^{(j)}(t) \right\} \right| \right] = n_j(x, t) dx \quad \forall x \in \Omega_B \setminus \Omega_P.$$

Condition (C.2). Whilst we aim to match the expected outcome of the stochastic simulation to $n(x, t)$, we also want the variances of molecule distribution in $\Omega_B \setminus \Omega_P$ to match that which would be expected if a BD simulation were to be performed over the entire domain Ω , which means

$$(2.8) \quad \text{var} \left[\left| \{x_i(t) \in A, i = 1, \dots, N_B(t)\} \right| \right] = \sigma_j^2(A, t)$$

for all $A \subset \Omega_B$ and $t > 0$. Here, $\sigma_j(A, t)$ was defined by (2.1).

In $\Omega_P \setminus \Omega_B$, the system is described by the concentration vector $p_j, j = 1, 2, \dots, M$, which evolves according to the PDE (2.2). This distribution, whilst continuous, is not strictly deterministic since it is coupled with the stochastic outcomes of the BD subdomain Ω_B . If the stochastic reaction-diffusion model only includes zero-order or first-order reactions, then the mean-field PDE (2.2) describes the expected behavior of stochastic models [12]. In this case it is reasonable to require the following additional condition.

Condition (C.3). We require

$$(2.9) \quad \mathbb{E} [p_j(x, t)] = n_j(x, t) \quad \forall x \in \Omega_P \setminus \Omega_B, t > 0.$$

In case [A], Conditions (C.1)–(C.3) can be simplified by observing that $\Omega_B = \Omega_B \setminus \Omega_P$ and $\Omega_P = \Omega_P \setminus \Omega_B$. In case [B], we will also require that the PBD algorithm gives the correct mean distribution of molecules in the overlap region $O = \Omega_B \cap \Omega_P$, i.e.,

$$(2.10) \quad \mathbb{E} [p_j(x, t)] dx + \mathbb{E} \left[\left| \left\{ x_i^{(j)}(t) \in [x, x + dx), i = 1, \dots, N_B^{(j)}(t) \right\} \right| \right] = n_j(x, t) dx$$

for all $x \in O = \Omega_B \cap \Omega_P$ and $t > 0$.

Note that we did not make any explicit assumptions for the solutions at the interface I . Instead, we formulate these assumptions implicitly using Conditions (C.1)–(C.3). Conditions (C.1) and (C.3) imply that PBD algorithms guarantee the continuity of the expected spatio-temporal concentration across the interface I .

3. PBD simulation of diffusion. In this section, we explain our case [A] PBD algorithm (i.e., $\Omega_B \cap \Omega_P = \emptyset$) using a system of diffusing noninteracting molecules of a single chemical species. Therefore, for all further discussions in this and the following section we will drop the index j representing the species. Then the macroscopic PDE (2.2) for this system becomes the diffusion equation

$$(3.1) \quad \frac{\partial n}{\partial t} = D \frac{\partial^2 n}{\partial x^2},$$

where $n \equiv n(x, t) : \Omega \times [0, \infty) \rightarrow [0, \infty)$ and D is the diffusion constant. We will consider the infinite domain $\Omega = \mathbb{R}$ for simplicity. Without loss of generality, we assume that $\Omega_P = (-\infty, 0)$ and $\Omega_B = (0, \infty)$, i.e., the internal boundary $I = \{0\}$ is situated at the origin $x = 0$. In the case of diffusion only, the total number of molecules N in the system is conserved, i.e.,

$$(3.2) \quad N = \int_{\Omega} n(x, t) dx \quad \forall t \geq 0.$$

Hence, for the PBD algorithm the conservation of mass condition takes the form

$$(3.3) \quad N = \int_{\Omega_P} p(x, t) dx + N_B(t) \quad \forall t \geq 0.$$

Since the diffusing molecules are noninteracting, we can evaluate $\sigma(A, t)$ given in (2.1) by assuming that every particle has the (identical) probability $p_A = \int_A n(x, t) dx / N$ of being in the set A at time t . Consequently, $\sigma^2(A, t)$ is equal to $N p_A (1 - p_A)$. Substituting for p_A and using (3.2), equation (2.8) takes the following form:

$$(3.4) \quad \text{var} [\{x_i(t) \in A, i = 1, \dots, N_B(t)\}] = \int_A n(x, t) dx \left(1 - \frac{\int_A n(x, t) dx}{\int_{\Omega} n(x, t) dx} \right)$$

for all $A \subset \Omega_B$ and $t > 0$. Using again $A = [x, x + dx)$, we obtain the alternative formulation

$$(3.5) \quad \text{var} [\{x_i(t) \in [x, x + dx), i = 1, \dots, N_B(t)\}] = n(x, t) dx.$$

In sections 3.1 and 3.2, we present steps for updating the continuum and the particle-based simulations, respectively, before the full PBD algorithm (A1)–(A5) is formulated in section 3.3. In section 3.4, we will discuss the accuracy of the algorithm with respect to Conditions (C.1)–(C.3).

3.1. Updating the PDE regime in Ω_P . At time t , we have the concentration $p(x, t)$ for $x \in \Omega_P$ and are aiming to calculate the concentration $p(x, t + \Delta t)$ that corresponds to a realization of one time step Δt of the diffusion process (3.1). We, therefore, define the exact outcome of a diffusion step in the full domain Ω given initial data $p(x, t)$:

$$(3.6) \quad \tilde{p}(x, t + \Delta t) = \int_{\Omega_P} K(x - x', \Delta t) p(x', t) dx',$$

where $K(x - x', \Delta t)$ is the diffusion kernel

$$(3.7) \quad K(\xi, \Delta t) = \frac{1}{\sqrt{4\pi D \Delta t}} \exp\left(-\frac{\xi^2}{4D \Delta t}\right)$$

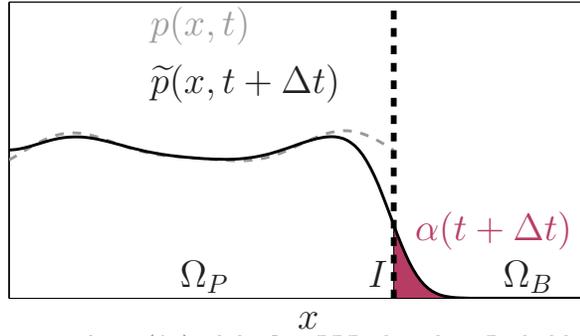


FIG. 3.1. Illustration of step (A1) of the first PBD algorithm. Dashed line: $p(x, t)$; solid line: $\tilde{p}(x, t + \Delta t)$ as defined in (3.6); shaded area: $\alpha(t + \Delta t)$ as defined in (3.8).

and the function $\tilde{p}(x, t)$ has support Ω . Using this process, a certain proportion of the concentration distribution, namely

$$(3.8) \quad \alpha(t + \Delta t) \equiv \int_{\Omega_B} \tilde{p}(x, t + \Delta t) dx,$$

would have crossed the interface I in the time interval $[t, t + \Delta t)$ (see Figure 3.1). This value $\alpha(t + \Delta t)$ represents the number of molecules expected to cross the interface from Ω_P to Ω_B in the time interval $[t, t + \Delta t)$. Since all molecules in Ω_P are identical (of the same chemical species and have the same probability distribution proportional to $\tilde{p}(x, t + \Delta t)$), the number of molecules that cross the interface from Ω_P to Ω_B in the time interval $[t, t + \Delta t)$ is Poisson distributed with an average of $\alpha(t + \Delta t)$. Let us assume that Δt has been chosen small enough to ensure that $\alpha(t + \Delta t) \ll 1$. In this case the probability of more than one particle crossing the interface is negligible and we need to consider two cases:

- (i) one particle gets created in Ω_B with the probability $\alpha(t + \Delta t)$;
- (ii) no particle is created with the probability $1 - \alpha(t + \Delta t)$.

For both cases we need to calculate the updated concentration $p(x, t + \Delta t)$ where $x \in \Omega_P$.

We consider the concentration $p(x, t)$ as the distribution of $N - N_B(t)$ identically distributed particles at time t . Therefore each of these particles has at time t the probability distribution $p(x, t)/(N - N_B(t))$. After one time step each of these particles can be found in an infinitesimal interval $[x, x + dx)$ with probability $\tilde{p}(x, t + \Delta t) dx$. For each particle, its probability distribution, given that it did not leave Ω_P , can be calculated as

$$p_1(x, t + \Delta t) = \frac{\tilde{p}(x, t + \Delta t)}{N - N_B(t) - \alpha(t + \Delta t)} \quad \text{for } x \in \Omega_P.$$

On the other hand, if a particle does leave the domain Ω_P , then its distribution function becomes zero for $x \in \Omega_P$. Instead the particle is introduced into Ω_B at time $t + \Delta t$ at a location x with the probability distribution given by

$$(3.9) \quad p_2(x, t + \Delta t) = \frac{\tilde{p}(x, t + \Delta t)}{\alpha(t + \Delta t)} \quad \text{for } x \in \Omega_B.$$

This updating process is rather like collapsing a wavefunction in quantum mechanics [30]. We look to see whether a particle has crossed the boundary into Ω_B : if it is

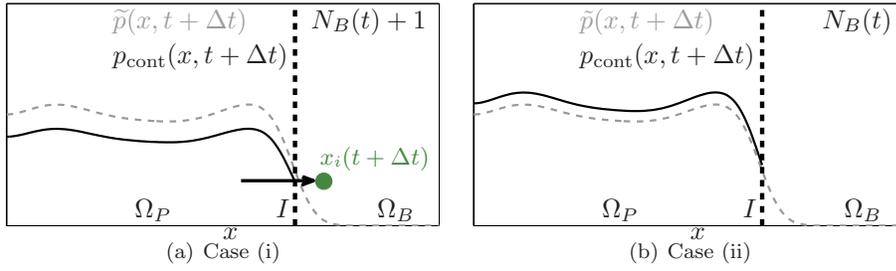


FIG. 3.2. Illustration of steps (A2) and (A3) of the first PBD algorithm. Dashed line: $\tilde{p}(x, t + \Delta t)$; solid line: $p_{\text{cont}}(x, t + \Delta t)$; (green in the electronic version) circle: created particle.

there on the other side, then its distribution function collapses to a δ function at its new position, while if it is not, then the distribution function collapses to zero in Ω_B with corresponding rescaling in Ω_P .

Combining these arguments for all the particles, we see that the last update step is a simple rescaling of the probability distribution so that the updated distribution satisfies conservation of mass according to (3.3). We, therefore, have

$$(3.10) \quad p_{\text{cont}}(x, t + \Delta t) = \begin{cases} \beta_{(i)} \tilde{p}(x, t + \Delta t) & \text{in case (i),} \\ \beta_{(ii)} \tilde{p}(x, t + \Delta t) & \text{in case (ii)} \end{cases} \quad \text{for } x \in \Omega_P,$$

with the rescaling constants $\beta_{(i)}$, $\beta_{(ii)}$ given by

$$(3.11) \quad \beta_{(i)} = \frac{N - N_B(t) - 1}{N - N_B(t) - \alpha(t + \Delta t)}, \quad \beta_{(ii)} = \frac{N - N_B(t)}{N - N_B(t) - \alpha(t + \Delta t)}.$$

Note that the updating step for the continuum regime satisfies conservation of mass (3.3). An illustration of the two cases can be seen in Figure 3.2.

3.2. Updating the BD regime in Ω_B . We use the discretized version of the overdamped Langevin equation introduced in (2.5) to update the positions of particles in Ω_B . If the position of the i th molecule, computed by (2.5), is in Ω_B at the end of the time step, then we will continue representing it as a particle. Note that a particle that crossed the boundary I and came back into Ω_B during the time step $[t, t + \Delta t)$ is also captured by this case. We have to be more careful whenever the position $x_i(t + \Delta t)$, computed by (2.5), is inside the PDE subdomain Ω_P at time $t + \Delta t$, i.e., $x_i(t + \Delta t) \in \Omega_P$. In this case, the random walk of the i th molecule crossed the interface I without crossing back at the end of the time step. This event needs to be taken into account for the update of the final concentration $p(x, t + \Delta t)$ in Ω_P . As we know the exact position of this particle i at time $t + \Delta t$, we add a Dirac δ function at the position $x_i(t + \Delta t) \in \Omega_P$. Therefore, we compute $p(x, t + \Delta t)$ in Ω_P by

$$p(x, t + \Delta t) = p_{\text{cont}}(x, t + \Delta t) + \sum_{x_i(t + \Delta t) \in \Omega_P} \delta(x - x_i(t + \Delta t)),$$

where $p_{\text{cont}}(x, t + \Delta t)$ is given by (3.10).

3.3. The first PBD algorithm. This algorithm computes the concentration $p(x, t)$ for $x \in \Omega_P$, and the number $N_B(t)$ and positions of BD particles $x_i(t) \in \Omega_B$,

TABLE 3.1

One time step of the first PBD algorithm for a system of diffusing molecules.

<p>(A1) Calculate $\tilde{p}(x, t + \Delta t)$ using (3.6) and $\alpha(t + \Delta t)$ using (3.8).</p> <p>(A2) Generate uniformly distributed random number r in $(0, 1)$.</p> <p style="padding-left: 2em;">(i) If $r < \alpha(t + \Delta t)$, then create a new particle in Ω_B according to the probability density $p_2(x, t + \Delta t)$ defined in (3.9). Set $\beta = \beta_{(i)}$, where $\beta_{(i)}$ is given by (3.11). Set $N_{B,1} = 1$.</p> <p style="padding-left: 2em;">(ii) If $r \geq \alpha(t + \Delta t)$, then set $\beta = \beta_{(ii)}$, where $\beta_{(ii)}$ is given by (3.11). Set $N_{B,1} = 0$.</p> <p>(A3) Compute positions $x_i(t + \Delta t)$, $i = 1, 2, \dots, N_B(t)$, of BD particles according to (2.5).</p> <p>(A4) Compute new concentration in Ω_P by</p> $p(x, t + \Delta t) = \beta \tilde{p}(x, t + \Delta t) + \sum_{x_i(t+\Delta t) \in \Omega_P} \delta(x - x_i(t + \Delta t)) \quad \text{for } x \in \Omega_P.$ <p>(A5) Update the number of BD particles by</p> $N_B(t + \Delta t) = N_B(t) + N_{B,1} - \{x_i(t + \Delta t) \in \Omega_P, i = 1, \dots, N_B(t)\} .$ <p>Terminate computation of trajectories of BD molecules which landed in Ω_P (i.e., the BD particles which satisfy $x_i(t + \Delta t) \in \Omega_P$).</p> <p>Then continue with step (A1) for time $t + \Delta t$.</p>
--

$i = 1, 2, \dots, N_B(t)$. One time step of the first PBD algorithm is presented in Table 3.1 as algorithm (A1)–(A5). In order to simplify the presentation of this algorithm, we consider that the time step Δt is chosen so small that $\alpha(t + \Delta t) \ll 1$. In particular, we need only implement cases (i)–(ii) presented in section 3.1, because the probability that two or more molecules are initiated in Ω_B during one time step is negligible.

The auxiliary distribution $\tilde{p}(x, t + \Delta t)$ in step (A1) is, in practice, calculated using a numerical approximation algorithm. To calculate $\alpha(t + \Delta t)$ we can either use this numerical approximation of $\tilde{p}(x, t + \Delta t)$, which requires an additional time step $\tilde{\Delta t} \ll \Delta t$ to be used to ensure the accuracy of $\alpha(t + \Delta t)$, or (more efficiently) we can approximate $\alpha(t + \Delta t)$ analytically using a boundary layer expansion in the vicinity of the interface I .

By construction, the algorithm (A1)–(A5) satisfies the conservation of mass condition (3.3) and the concentration $p(x, t)$ satisfies nonnegativity. We will now show that this algorithm also guarantees the correct expected outcome and, therefore, satisfies Conditions (C.1) and (C.3).

THEOREM 3.1. *Consider the BD simulation of N diffusing molecules in the computational domain Ω , which is divided into subdomains $\Omega_B \subset \Omega$ and $\Omega_P \subset \Omega$ satisfying (2.3) and case [A]. Suppose that the $N_B(0)$ particles are initially distributed in Ω_B at positions $x_i(0)$, $i = 1, 2, \dots, N_B(0)$, and that the remaining $N - N_B(0)$ particles are randomly distributed according to $p(x, 0)$ in Ω_P such that $p(x, 0) = n(x, 0)$ for $x \in \Omega_P$. Then the expected outcome of the PBD algorithm (A1)–(A5) (presented in Table 3.1) satisfies Conditions (C.1) and (C.3) for arbitrary $\Delta t > 0$.*

Proof. We will show that the identities (2.6) and (2.9) hold during one iteration (A1)–(A5) presented in Table 3.1. It will then follow by induction that they hold for all times $k\Delta t$, $k = 0, 1, 2, \dots$.

Let us assume that the probability distribution of particles in Ω_B at time t is given through $n(x, t)$, $x \in \Omega_B$. Given the probability distribution $p(x, t)$, $x \in \Omega_P$, at

time t , the conditional expected value of $p(x, t + \Delta t)$ for $x \in \Omega_P$ is given by

$$\begin{aligned} \mathbb{E} [p(x, t + \Delta t) | p(x, t)] &= \alpha(t + \Delta t) \beta_{(i)} \tilde{p}(x, t + \Delta t) \\ &\quad + (1 - \alpha(t + \Delta t)) \beta_{(ii)} \tilde{p}(x, t + \Delta t) \\ &\quad + \int_{\Omega_B} K(x - x', \Delta t) n(x', t) \, dx', \end{aligned}$$

where $\beta_{(i)}$ and $\beta_{(ii)}$ are given by (3.11), $K(x - x', \Delta t)$ is the diffusion kernel given in (3.7), and the last term represents particles that moved across the interface I , defined by (2.4), during the time step $[t, t + \Delta t)$. Using (3.11), we obtain

$$\mathbb{E} [p(x, t + \Delta t) | p(x, t)] = \tilde{p}(x, t + \Delta t) + \int_{\Omega_B} K(x - x', \Delta t) n(x', t) \, dx'.$$

Using (3.6) and the law of total expectation (law of iterated expectations), we get

$$\mathbb{E} [p(x, t + \Delta t)] = \int_{\Omega_P} K(x - x', \Delta t) \mathbb{E}[p(x', t)] \, dx' + \int_{\Omega_B} K(x - x', \Delta t) n(x', t) \, dx'.$$

Using the induction assumption that $\mathbb{E}[p(x, t)] = n(x, t)$, we obtain

$$\mathbb{E} [p(x, t + \Delta t)] = \int_{\Omega} n(x', t) K(x - x', \Delta t) \, dx' = n(x, t + \Delta t) \quad \text{for } x \in \Omega_P,$$

i.e., we have derived (2.9).

Let us consider a set $A \subset \Omega_B$. Given the probability distribution $p(x, t)$, $x \in \Omega_P$, at time t , the conditional expected number of particles in A at time $t + \Delta t$ is

$$\begin{aligned} \mathbb{E} [|\{x_i(t) \in A, i = 1, \dots, N_B(t)\}| | p(x, t)] &= \int_A \alpha(t + \Delta t) p_2(x, t + \Delta t) \, dx \\ &\quad + \int_A \int_{\Omega_B} K(x - x', \Delta t) n(x', t) \, dx' \, dx, \end{aligned}$$

where the first term represents newly created particles from the PDE regime Ω_P and the second term represents the movement of particles inside Ω_B . Using (3.6), (3.9), and the law of total expectation, we obtain

$$\begin{aligned} \mathbb{E} [|\{x_i(t) \in A, i = 1, \dots, N_B(t)\}|] &= \int_A \int_{\Omega} K(x - x', \Delta t) n(x', t) \, dx' \, dx \\ &= \int_A n(x, t + \Delta t) \, dx, \end{aligned}$$

which is the condition (2.6). Thus we have showed that Conditions (C.1) and (C.3) are both satisfied. This concludes the proof. \square

Theorem 3.1 also holds if the algorithm (A1)–(A5) is extended to the creation of more than one new particle per time step as long as the expected value of the number of created particles is equal to $\alpha(t + \Delta t)$ and the rescaling is done accordingly. In all simulations throughout this paper, we indeed make use of this extension and allow for more than one particle to be created in Ω_B per timestep. The algorithm (A1)–(A5) and the proof can also be easily extended for a finite domain $\Omega = [0, L]$ with no flux boundary conditions by redefining the kernel $K(\xi, \Delta t)$ accordingly.

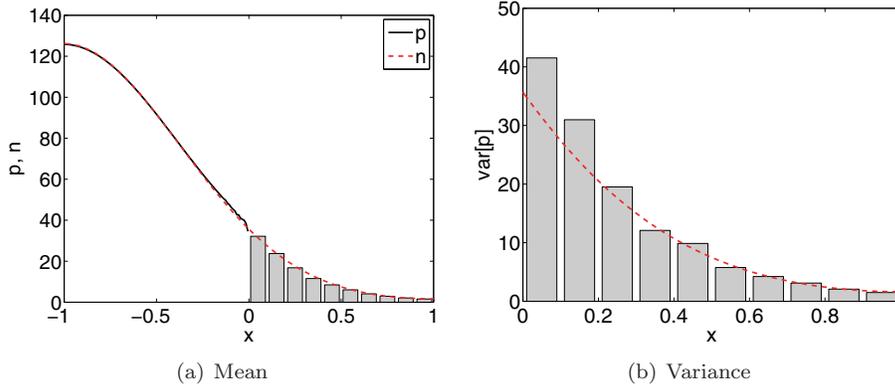


FIG. 3.3. Simulation results of diffusion of 100 particles in $(-1, 1)$ with no flux boundary conditions initialized at $x = -0.95$, i.e., $n(x, 0) = 100 \delta(x + 0.95)$. Results averaged over 1000 realizations. Dashed (red in the electronic version) line: expected outcome. (a) Solid line: mean value in Ω_P ; gray bars: particle concentrations in Ω_B . (b) Gray bars: measured variances in particle concentrations. Parameter values as shown in the text.

3.4. Discussion of the PBD algorithm (A1)–(A5). In Theorem 3.1 we showed that the PBD algorithm (A1)–(A5) satisfies Conditions (C.1) and (C.3). However, we still need to check whether Condition (C.2) on the variances is also satisfied.

To investigate the variances created by this algorithm, we show the outcome of an illustrative numerical example in Figure 3.3. We simulate the diffusion of 100 molecules in the domain $\Omega = [-1, 1]$ which are initialized at the same location $x = -0.95$. We use no flux boundary conditions. We test the algorithm (A1)–(A5) where $\Omega_P = (-1, 0)$, $\Omega_B = (0, 1)$, and $I = \{0\}$. To calculate $\tilde{p}(x, \Delta t)$ we use an implicit Euler-scheme with $\Delta x = 0.01$ and a numerical time step of $\tilde{\Delta t} = 10^{-6}$. The time step Δt used by the algorithm (A1)–(A5) is $\Delta t = 10^{-3}$ and we simulate the system until $T_{\text{final}} = 0.2$. The time step $\tilde{\Delta t} \ll \Delta t$ for the approximation of $\tilde{p}(x, t + \Delta t)$ was chosen in order to minimize numerical artefacts. We run 1000 realizations of this process and measure the number of particles in 10 intervals (“bins”) of the size 0.1 in Ω_B . Averaging over 1000 realizations, we calculate the mean value and the variance of the particle number for each of the bins at time T_{final} . The results are presented in Figure 3.3 as gray histograms. In this example, it is easy to calculate the correct distribution $n(x, t)$ which the algorithm (A1)–(A5) tries to approximate. It can be obtained by solving the diffusion equation (3.1) in the domain $\Omega = (-1, 1)$ with $n(x, 0) = 100 \delta(x + 0.95)$ and no flux boundary conditions. The expected values for both means and variances are plotted as (red in the electronic version) dashed lines in Figure 3.3.

In Figure 3.3(a), we see that the mean value of the simulation results matches well with the solution of the diffusion process, with only small fluctuations close to the internal boundary I at $x = 0$ due to stochastic effects and inaccuracies caused by the numerical approximation of $\tilde{p}(x, t + \Delta t)$. However, in Figure 3.3(b) it becomes clear that the variance between different realizations is higher than the desired value, in particular close to the internal interface. These kinds of numerical artifacts were always to be expected when coupling an inherently stochastic BD simulation (with a single particle resolution) with a deterministic PDE system (that considers only concentrations). In particular, we identified one such effect and will explain it using

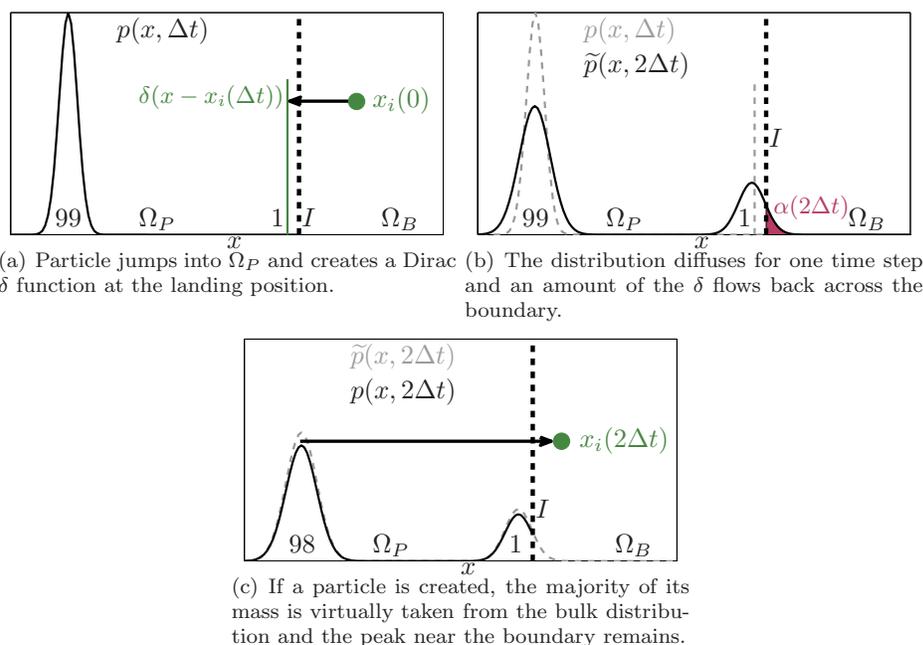


FIG. 3.4. Thought experiment that leads to errors in the variance.

a thought experiment.

Let us consider a situation where $p(x, 0)$ is 0 close to the internal boundary I and has a peak of mass 99 arbitrarily far away from I . Additionally, we assume that one particle is situated in Ω_B close to the interface I . Assuming the particle crosses the interface in the first simulation step, a Dirac δ function is created in Ω_P close to the interface, as illustrated in Figure 3.4(a). This δ function has a large impact on the region in Ω_P that is close to the interface, as the distribution $p(x, \Delta t)$ is negligible in this region. Immediately after incorporating the particle into $p(x, \Delta t)$, all its information is lost and we are forced to assume 100 independent particles with the probability distribution $p(x, \Delta t)$ in Ω_P . In particular, this implies that every particle has a 1% chance of being at the position of the δ and 99% chance of being in the bulk distribution far away from the boundary. This is indeed not the case, since we know that there is exactly one molecule near the interface and 99 molecules away from the interface, but the nature of the continuum distribution means that this information must be lost or else we should necessarily demand a separate distribution for all molecules in Ω_P .

In the second diffusion step, we calculate $\tilde{p}(x, 2\Delta t)$ according to (3.6), and some “mass” $\alpha(2\Delta t)$ may have drifted across the interface I (see Figure 3.4(b)). Because the bulk distribution is far away from the boundary, almost all of this mass $\alpha(2\Delta t)$ comes from the δ function close to I . Let us imagine that a new particle is now created in Ω_B (according to the probability $\alpha(2\Delta t)$), in which case the whole distribution needs to be rescaled, as shown in Figure 3.4(c). Because 99% of the mass in $p(x, \Delta t)$ is situated in the bulk far away from the boundary, the majority of the rescaling happens in this region such that effectively the mass needed to create the particle is almost entirely taken from the bulk, rather than from the region close to the interface. This also implies that most of the mass close to the boundary will stay, and it is therefore

possible to create another particle from this mass in further time steps. This is in contradiction to the result that would be expected if information was not lost in the first time step. That is, the distribution close to the interface should disappear and the bulk far from the interface left alone. This effect is the main reason a higher than expected variance can be measured in Ω_B . Note, however, that this does not effect the expected values, as shown in Theorem 3.1.

Of course, our thought example is an extreme case: We would expect that in practical cases there would be a significant density of particles throughout the continuum region (otherwise individual particle tracking would be preferable). Nevertheless the fact that all information about an individual particle is lost as soon as it crosses the interface does generate an error in the variance of particle numbers near the interface, and the effect becomes more pronounced when the concentrations in Ω_P close to I are low.

3.4.1. Dependence of the variance on the system parameters. We want to quantify the error in the variance as a function of the system parameters, which are the size of the domain $[-L, L]$, the diffusion constant D , the simulated time T_{final} , and the total mass N . After a nondimensionalization the macroscopic PDE (3.1) can be written in the form

$$\frac{\partial n}{\partial t} = \frac{\partial^2 n}{\partial x^2}, \quad x \in [-1, 1],$$

where the simulation is run until

$$T_{\text{final}}^* = T_{\text{final}} \frac{L^2}{D}.$$

Hence, the system only has two parameters that need to be investigated: The final simulation time T_{final}^* and the total number of molecules N .

As in Figure 3.3, we use PBD algorithm (A.1)–(A.5) with $\Omega_P = (-1, 0)$, $\Omega_B = (0, 1)$, and $I = \{0\}$. For each parameter (T_{final}^* and N), we simulate the system 1000 times for different values of this parameter and measure in each case the number of particles in Ω_B at the end of the simulation, i.e., the value $N_B(T_{\text{final}}^*)$. In Figure 3.5,

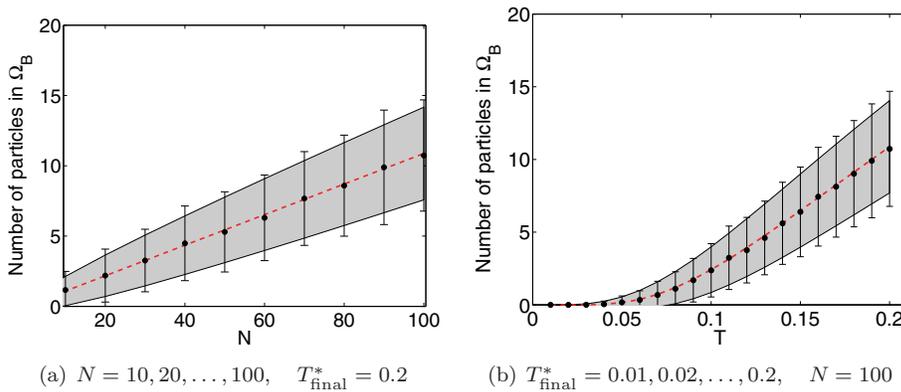


FIG. 3.5. Mean values and standard deviations of the number of $N_B(T_{\text{final}}^*)$ depending on T_{final}^* and N . Dashed line: expected number of particles; shaded area: expected standard deviation; dots: measured mean values; error bars: measured standard deviations. Other parameters are chosen as for Figure 3.3.

we see that the measured mean values match well with the expected outcomes. For the standard deviations, however, we see that the measured outcomes are higher than expected for all values of T_{final}^* and N . This is an undesired effect and the next section will discuss a way to reduce this artifact.

4. A PBD algorithm with an overlap region. In section 3.4 we saw that the immediate return of particles from Ω_P into Ω_B in combination with relatively low concentrations close to the interface can lead to errors in the variance of particle concentrations in Ω_B . One way to overcome this problem is the introduction of an overlap region where BD simulation and a continuum description exist in parallel, i.e., we will consider case [B] defined in section 2 by $\Omega_B \cap \Omega_P \neq \emptyset$. A sketch of this new set up can be seen in Figure 4.1 with the overlap region denoted as $O = \Omega_B \cap \Omega_P$. We also denote the interfaces I_1 and I_2 by

$$I_1 = \partial\Omega_B \cap \Omega_P, \quad I_2 = \Omega_B \cap \partial\Omega_P.$$

In the case of diffusion only, this new setup requires only subtle changes in the algorithm. Molecules are now incorporated into the concentration when they cross the boundary I_1 . The definition of \tilde{p} is still equal to (3.6), but we have to redefine $\alpha(t + \Delta t)$ and $p_2(x, t + \Delta t)$ as follows:

$$(4.1) \quad \alpha(t + \Delta t) = \int_{\Omega_B \setminus \Omega_P} \tilde{p}(x, t + \Delta t) dx,$$

$$(4.2) \quad p_2(x, t + \Delta t) = \frac{\tilde{p}(x, t + \Delta t)}{\alpha(t + \Delta t)} \quad \text{for } x \in \Omega_B \setminus \Omega_P.$$

The introduction of the overlap region prevents undesired effects generated by molecules crossing over and coming back straight away, as discussed in the thought experiment in section 3.4. In particular, a molecule initialized as a Dirac δ function in $\Omega_P \setminus \Omega_B$ initially contributes very little to the overall probability density near the interface I_2 ; by the time it has a significant probability of crossing I_2 its distribution has become sufficiently spread that it is “lost” in the subdomain Ω_P as is required for the continuous distribution. One time step of the second PBD algorithm is presented in Table 4.1. As before, we consider that the time step Δt is chosen so small that $\alpha(t + \Delta t) \ll 1$. Therefore, we need only implement cases (i)–(ii) in step (B2), because

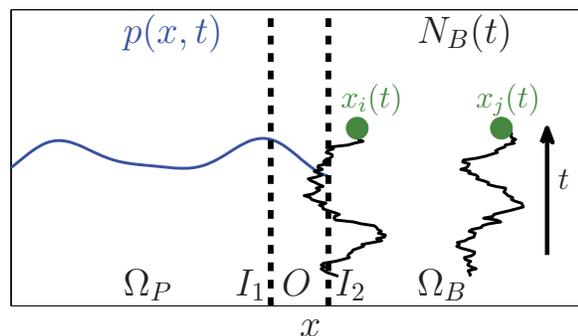


FIG. 4.1. Sketch of the PBD algorithm with overlap region and the notation related to it. In Ω_P , molecules are described by their density distribution $p(x, t)$ in the microscopic domain Ω_B described by the number $N_B(t)$ of molecules and their positions $x_i(t)$, $i = 1, \dots, N_B(t)$. In the overlap region O , both descriptions exist in parallel. The interfaces between the various subdomains are denoted I_1 and I_2 .

TABLE 4.1

One time step of the PBD algorithm with overlap region for the system of diffusing molecules.

<p>(B1) Calculate $\tilde{p}(x, t + \Delta t)$ using (3.6) and $\alpha(t + \Delta t)$ using (4.1).</p> <p>(B2) Generate uniformly distributed random number r in $(0, 1)$.</p> <p>(i) If $r < \alpha(t + \Delta t)$, then create new particle in $\Omega_B \setminus \Omega_P$ according to the probability density $p_2(x, t + \Delta t)$ defined in (4.2). Set $\beta = \beta_{(i)}$, where $\beta_{(i)}$ is given by (3.11). Set $N_{B,1} = 1$.</p> <p>(ii) If $r \geq \alpha(t + \Delta t)$, then set $\beta = \beta_{(ii)}$, where $\beta_{(ii)}$ is given by (3.11). Set $N_{B,1} = 0$.</p> <p>(B3) Compute positions $x_i(t + \Delta t)$, $i = 1, 2, \dots, N_B(t)$, of BD particles according to (2.5).</p> <p>(B4) Compute new concentration in Ω_P by</p> $p(x, t + \Delta t) = \beta \tilde{p}(x, t + \Delta t) + \sum_{x_i(t + \Delta t) \in \Omega_P \setminus \Omega_B} \delta(x - x_i(t + \Delta t)) \quad \text{for } x \in \Omega_P.$ <p>(B5) Update the number of BD particles by</p> $N_B(t + \Delta t) = N_B(t) + N_{B,1} - \{x_i(t + \Delta t) \in \Omega_P \setminus \Omega_B, i = 1, \dots, N_B(t)\} .$ <p>Terminate computation of trajectories of BD molecules which landed in $\Omega_P \setminus \Omega_B$ (i.e., the BD particles which satisfy $x_i(t + \Delta t) \in \Omega_P \setminus \Omega_B$). Then continue with step (B1) for time $t + \Delta t$.</p>
--

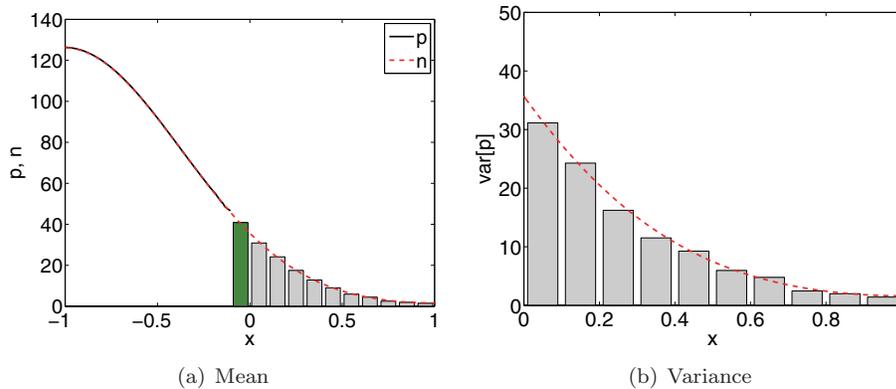


FIG. 4.2. Simulation results of a diffusion process in $\Omega = (-1, 1)$ with no flux boundary conditions and initial conditions $n(x, 0) = 100\delta(x + 0.95)$ with $\Omega_P = (-1, 0)$, $\Omega_B = (-0.1, 1)$, and $O = (-0.1, 0)$ averaged over 1000 realizations. Dashed (red in the electronic version) line: expected outcome. (a) Solid line: mean value in Ω_P ; bars: particle concentrations in $\Omega_B \setminus O$ (gray) and O (green in the electronic version). (b) Gray bars: measured variances in particle concentrations. Parameters as for Figure 3.3.

the probability that two or more molecules are initiated in Ω_B during one time step is negligible.

In order to highlight the advantages of the overlap region, we simulate the same diffusion process as in Figure 3.3 with $\Omega_P = (-1, 0)$ and $\Omega_B = (-0.1, 1)$. Then the overlap region is $O = \Omega_P \cap \Omega_B = (-0.1, 0)$. The results are shown in Figure 4.2. As before, the mean outcome matches well with the exact solution, with stochastic fluctuations inside the overlap region O due to the mixed description. In Figure 4.2(b) we see that the introduction of the overlap region indeed reduced the problem of high variances inside $\Omega_B \setminus \Omega_P$. A proof similar to Theorem 3.1 can be used to show that the PBD algorithm (B1)–(B5) satisfies Conditions (C.1) and (C.3) for $\Delta t > 0$. We will now show further that the algorithm describes a diffusion process exactly in the

limit $\Delta t \rightarrow 0$.

THEOREM 4.1. *Suppose that a PDE-description of the system is used in $\Omega_P = (-\infty, 0)$, and a BD simulation in $\Omega_B = (-d, \infty)$ with the overlap region $O = (-d, 0)$, where $d > 0$. In the limit that $\Delta t \rightarrow 0$, the expected concentration distribution $P_P(x, t) = \mathbb{E}[p(x, t)]$ in the continuum regime and the expected concentration distribution $P_B(x, t)$ in the BD regime obey the equations*

$$(4.3) \quad \frac{\partial P_P}{\partial t} = D \frac{\partial^2 P_P}{\partial x^2} + D \frac{\partial P_B}{\partial x} \Big|_{x \rightarrow -d_+} \delta(x + d), \quad x \in \Omega_P,$$

$$(4.4) \quad \frac{\partial P_B}{\partial t} = D \frac{\partial^2 P_B}{\partial x^2} - D \frac{\partial P_P}{\partial x} \Big|_{x \rightarrow 0_-} \delta(x), \quad x \in \Omega_B,$$

where

$$P_P(0, t) = 0, \quad P_B(-d, t) = 0 \quad \text{for } t > 0.$$

Extend each distribution to the whole line by defining $P_P(x, t) = 0$ for $x \in (0, \infty)$ and $P_B(x, t) = 0$ for $x \in (-\infty, -d)$. Then the sum of these two processes $n(x, t) = P_P(x, t) + P_B(x, t)$, $x \in \Omega$, $t > 0$ satisfies the diffusion equation (3.1).

Proof. Consider the function

$$(4.5) \quad n(x, t) = P_P(x, t) + P_B(x, t).$$

Clearly

$$\frac{\partial n}{\partial t} = D \frac{\partial^2 n}{\partial x^2} \quad \text{for } x \in (-\infty, -d) \cup (-d, 0) \cup (0, \infty).$$

Since each of P_P and P_B is continuous at $x = -d$ and $x = 0$, function n will be continuous there. Moreover, since

$$\left[\frac{\partial P_P}{\partial x} \right]_{x=0_-}^{x=0_+} = - \frac{\partial P_P}{\partial x} (0_-, t), \quad \left[\frac{\partial P_B}{\partial x} \right]_{x=0_-}^{x=0_+} = \frac{\partial P_P}{\partial x} (0_-, t),$$

$\partial n / \partial x$ is continuous at $x = 0$. Similarly, since

$$\left[\frac{\partial P_P}{\partial x} \right]_{x=-d_-}^{x=-d_+} = - \frac{\partial P_B}{\partial x} (-d_+, t), \quad \left[\frac{\partial P_B}{\partial x} \right]_{x=-d_-}^{x=-d_+} = \frac{\partial P_B}{\partial x} (-d_+, t),$$

$\partial n / \partial x$ is also continuous at $x = -d$. By standard regularity results this is enough to guarantee that n satisfies the diffusion equation on the whole real line. \square

Remark. For the overlap region to give a different result from the simple PBD algorithm (A1)–(A5), we should choose d much bigger than the $\sqrt{2D\Delta t}$ mean displacement of a particle given the time step Δt as defined in (2.5).

4.1. Dependence of the variance on the system parameters. In order to show that the variances are indeed accurately produced by the PBD algorithm (B1)–(B5), we repeat the numerical experiments conducted in section 3.4.1. We choose $\Omega_P = (-1, 0)$, $O = (-0.1, 0)$, and $\Omega_B = (-0.1, 1)$ and measured the number of molecules situated in $\Omega_B \setminus \Omega_P = [0, 1)$ at the end of the simulation. We again calculate the mean values and standard deviations and present the results in Figure 4.3. We can clearly see that the adjusted algorithm produces more accurate standard deviations than the original algorithm, whilst keeping the mean values correct. We conclude that the algorithm (B1)–(B5) produces an accurate BD simulation inside $\Omega_B \setminus \Omega_P$ and therefore satisfies Conditions (C.1)–(C.3). We will use the PBD algorithm (B1)–(B5) in the remainder of this paper.

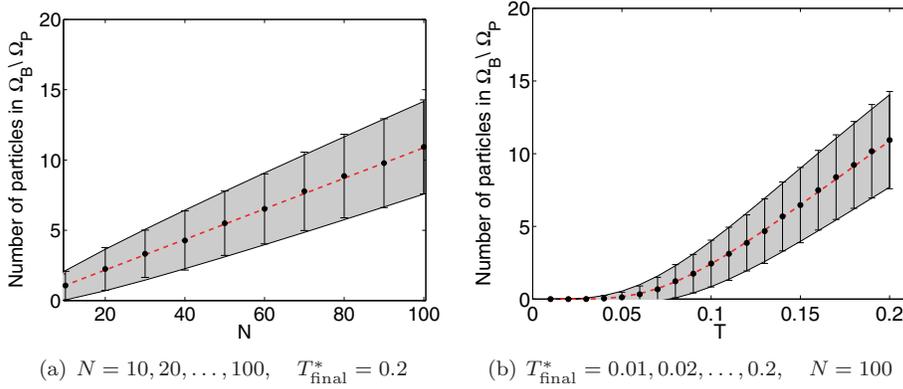
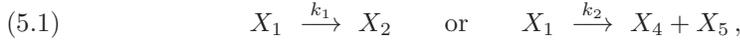


FIG. 4.3. Mean values and standard deviations of the number of particles in $\Omega_B \setminus \Omega_P$ at time T_{final}^* depending on T_{final}^* and N . Dashed line: expected number of particles; shaded area: expected standard deviation; dots: measured mean values; error bars: measured standard deviations. Parameters as for Figure 4.2.

5. Reaction-diffusion systems. In the next step we introduce chemical reactions into the system presented in section 4. We will concentrate on zero-order and first-order reactions [12]. First-order reactions are reactions which only have one reactant, for example,



where X_i denote chemical species and k_1 (resp., k_2) the corresponding rate constant (which has physical units $[\text{sec}^{-1}]$). In what follows, we will denote by \emptyset chemical species which are of no interest to a modeler. Then, considering that X_1 is the only chemical species of interest, we can rewrite the reactions (5.1) as



where $k_d = k_1 + k_2$. We will also consider zero-order reactions. An example is



where the rate constant k_p has physical units $[\text{M sec}^{-1}]$, i.e., it is the production rate per unit of volume and unit of time. It is relatively straightforward to implement zero-order and first-order chemical reactions in the PBD algorithms (A1)–(A5) and (B1)–(B5), because these reactions can be treated in the individual parts of the system (continuum and BD simulation) independently. Note that for higher-order reactions this is not necessarily the case, as particles could react with the continuum inside the overlap region O . This case is not discussed in this paper.

In the continuum regime reactions are represented by the term $R_j(p_1, p_2, \dots, p_M)$ on the right-hand side of the reaction-diffusion PDE (2.2). For example, if the chemical species X_1 is subject to chemical reactions (5.2)–(5.3), then the reaction-diffusion PDE (2.2) takes the form

$$\frac{\partial p_1}{\partial t} = D_1 \frac{\partial^2 p_1}{\partial x^2} - k_d p_1 + k_p.$$

In the BD simulations, the molecules act independently and the reactions can, therefore, be treated individually. A summary of how to implement various reactions in BD simulations can be found in [12].

Although implementation of the zero-order and first-order reactions is relatively straightforward, one still has to consider some special effects that are related to the coupling of the two parts of the domain. First, what happens when a particle that is supposed to react at time t_1 crosses the interface I_1 at an earlier time $t_2 < t_1$? Since we assumed that all information about particles is lost as soon as they cross the interface I_1 , we incorporate it into the continuum, and the reaction at time t_1 does not happen. Second, what happens when a particle is created inside the overlap region O ? A number of solutions to this problem are possible: One could split the creation in equal parts, or declare creation to contribute only to either the continuum or the molecular-based description. We will assume here that all creation inside the overlap region occurs in the form of molecules with exact positions.

Finally, let us note that (reactive) boundary conditions on the external boundary $\partial\Omega$ can be treated according to the corresponding modelling regime, i.e., whether the corresponding segment of $\partial\Omega$ is part of $\partial\Omega_P$ or $\partial\Omega_B$. Derivation of reactive boundary conditions of BD simulations which are consistent with the PDE description can be found in [8]. External boundaries slightly modify the computation of $\tilde{p}(x, t + \Delta t)$ in Ω_P . It is still given by (3.6) but the kernel (3.7) has to be updated to take into account the boundary condition imposed on the external boundary $\partial\Omega$. We have already done this when we showed simulations of the diffusion process in Figures 3.3 and 4.2 in the finite interval $(-1, 1)$ with no flux boundary conditions. However, for small timesteps Δt the change is negligible, since all the action takes place near the interface I_2 .

We conclude this section with three examples which illustrate the behavior of the PBD algorithm (B1)–(B5) for reaction-diffusion systems. They include the modeling of morphogen gradients and chemisorption.

5.1. Example 1: Morphogen gradient. In the first example we compute a steady state for a morphogen gradient model [28, 32, 20]. We consider one chemical species (morphogen) inside the domain $\Omega = (-1, 1)$. All parameters are dimensionless for simplicity. The only reaction inside $\Omega = (-1, 1)$ is the degradation (5.2). Additionally to this reaction, we assume a constant influx J/D through the left-hand boundary $x = -1$ to the continuum subdomain $\Omega_P = (-1, 0)$. We use $O = (-0.1, 0)$ and $\Omega_B = (-0.1, 1)$ with a no flux boundary at $x = 1$. Since we have only first-order reaction (5.2), the exact solution $n(x, t)$ is given by

$$(5.4) \quad \frac{\partial n}{\partial t} = D \frac{\partial^2 n}{\partial x^2} - k_1 n, \quad \frac{\partial n}{\partial x}(-1, t) = -J, \quad \frac{\partial n}{\partial x}(1, t) = 0.$$

This system is initialized with $n(x, 0) = 0$, $x \in \Omega$, and we let it run until it (approximately) reaches the steady state. We use $J = 1000$ and $k_1 = 1$. The second PBD algorithm (B1)–(B5) is run with the same parameters presented in section 4 for time $T_{\text{final}} = 20$, which is (approximately) a time at which the model settles into the steady state. The reaction (5.2) is simulated in a time-driven manner in Ω_B , which means that for each morphogen molecule it is decided randomly at the end of each time step whether it was degraded or not (the probability of degradation of each molecule is equal to $k_d \Delta t$ provided that $k_d \Delta t \ll 1$).

The result of a single simulation of the PBD algorithm (B1)–(B5) is plotted in Figure 5.1(a). We plot the PDE solution in $\Omega_P \setminus O$ as a black line and the gray

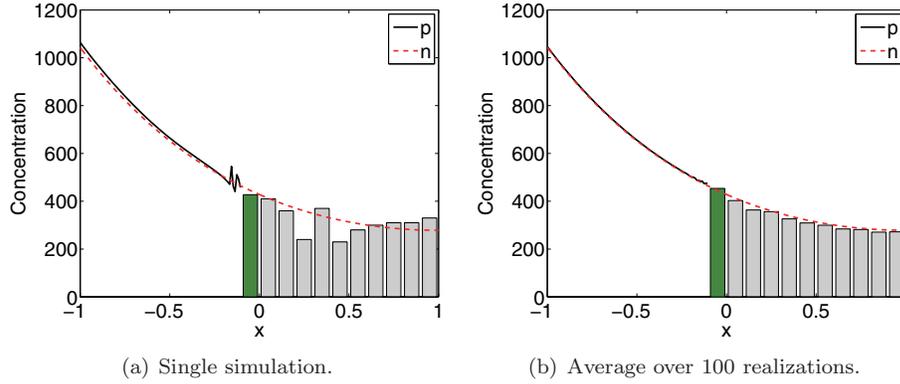


FIG. 5.1. Simulation results for Example 1. Dashed (red in the electronic version) line: exact solution given by (5.4); solid line: $p(x, T_{\text{final}})$; gray bars: spatial concentration of particles at $t = T_{\text{final}}$ in $\Omega_B \setminus O$; dark gray (green in the electronic version) bar: $N_O(T_{\text{final}})$ given by (5.5). Parameters as described in the text.

histogram of molecules in $\Omega_B \setminus O$. In the overlap region O , we compute the total “number” of particles by

$$(5.5) \quad N_O(T_{\text{final}}) \equiv \int_O p(x, T_{\text{final}}) dx + |\{x_i(T_{\text{final}}) \in O, i = 1, 2, \dots, N_B(T_{\text{final}})\}|.$$

The value of $N_O(T_{\text{final}})$ is plotted as the darker (green in the electronic version) bar in Figure 5.1(a). The results of a single simulation of the PBD algorithm (B1)–(B5) are compared with the exact solution, which is obtained by solving the PDE (5.4) numerically until time T_{final} . In this simple example, it is also possible to find an analytical expression for the steady state profile which is approximately equal to the presented dashed line.

Note that the jagged appearance of the continuum solution close to the interface is not numerical error, but represents the fact that as molecules cross from the discrete to the continuum side information about their exact location is lost gradually over time (remember that Figure 5.1(a) shows just one realization of the stochastic process). The corresponding distribution on the discrete side $\Omega_B \setminus \Omega_P$ would be δ function spikes at the location of the particles, which we have in effect locally averaged by our binning process. Thus the jaggedness can be seen as a gradual transition in the solution from isolated discrete particles to a continuum density distribution. An ensemble average over 100 simulations of the PBD algorithm (B1)–(B5) is shown in Figure 5.1(b). The stochastic fluctuations are reduced compared to the single simulation and the results compare well with the exact solution for the expected probability density (5.4).

5.2. Example 2: Reversed morphogen gradient. In this example we introduce a second reaction in addition to (5.2)—a local production of molecules:



where x_s defines the size of the creation zone. As before we consider all parameters to be dimensionless: k_p is defined as the rate of production per unit length. For this system we use no flux boundary conditions on both ends. The combination of localized production (5.6) and degradation (5.2) ensures that the system settles into a nontrivial

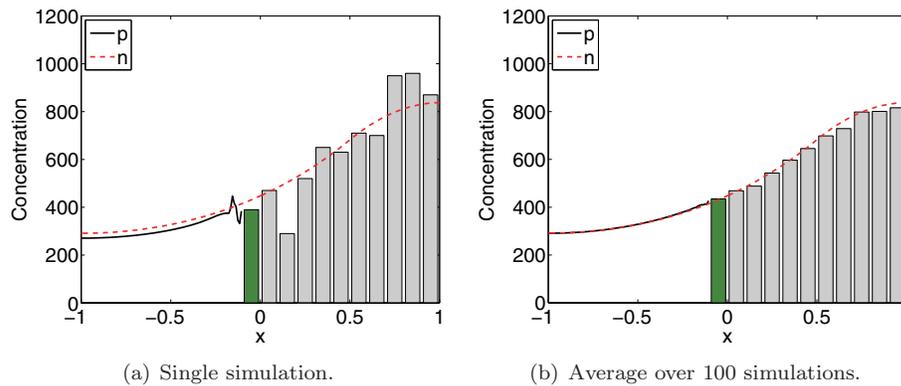


FIG. 5.2. Simulation results for Example 2. Dashed (red in the electronic version) line: exact solution given by (5.7); solid line: $p(x, T_{\text{final}})$ in $\Omega_P \setminus O$; gray bars: spatial concentration of particles at $t = T_{\text{final}}$ in $\Omega_B \setminus O$; dark gray (green in the electronic version) bar: $N_O(T_{\text{final}})$ given by (5.5). Parameters as described in the text.

steady-state which we will compute with the PBD algorithm (B1)–(B5). The exact solution (which the PBD algorithm (B1)–(B5) approximates) can be described by

$$(5.7) \quad \frac{\partial n}{\partial t} = D \frac{\partial^2 n}{\partial x^2} - k_1 n + k_2 \chi_{[x_s, 1]}, \quad \frac{\partial n}{\partial x}(-1, t) = 0, \quad \frac{\partial n}{\partial x}(1, t) = 0,$$

where $\chi_{[x_s, 1]}$ is the characteristic function for the interval $[x_s, 1]$ that takes the value 1 inside and 0 outside of the interval. The production reaction (5.6) was implemented in BD simulations in an event-driven way such that particles can be created at any time in-between time steps and the number of particles created in one time step is not limited; we used $k_p = 1$, $k_d = 2000$, and $x_s = 0.5$. For the PBD simulations we use the same parameters as in section 4. In particular, we have $\Omega_P = (-1, 0)$, $O = (-0.1, 0)$, and $\Omega_B = (-0.1, 1)$.

A single realization of this process is plotted in Figure 5.2(a). We plot the PDE solution in $\Omega_P \setminus O$ as a black line. The concentration of molecules in $\Omega_B \setminus O$ is visualized as a gray histogram. In the overlap region O , we plot $N_O(T_{\text{final}})$ given by (5.5). The concentration gradient is now reversed, as the creation of particles happens near the right-hand boundary. Again, one can clearly see the stochastic fluctuations in this plot which also have effect on the value of $p(x, t)$ far from the overlap region O . However, as we draw an ensemble average over 100 simulations in Figure 5.2(b), the results converge towards the exact solution which is obtained by solving the PDE (5.7). It is plotted as a dashed (red in the electronic version) line in Figure 5.2.

5.3. Example 3: Chemisorption. Our last example is the polymer coating of a virus surface [14, 11]. We will describe it as irreversible adsorption (chemisorption) of polymers to a two-dimensional surface as was introduced in [9]. This example presents a typical application area of PBD algorithms. A detailed model is used close to the reactive boundary where positions of individual molecules influence the dynamics of diffusion-driven adsorption. On the other hand, a less detailed model can be used far away from the adsorbing surface. In the bulk, the behavior of reactive polymers can be described by the macroscopic reaction-diffusion PDE (2.2) in the

form

$$(5.8) \quad \frac{\partial p}{\partial t} = D \frac{\partial^2 p}{\partial z^2} - k_d p$$

in the semi-infinite domain $\Omega = (0, \infty)$ (here, z is the distance from the reactive surface which is at $z = 0$). This equation takes into account two processes which mainly influence chemisorption dynamics [9]: diffusion of polymer molecules and the hydrolysis of reactive groups in the solution. Both processes can be implemented in the BD context as we saw in the previous examples. However, this level of detail is only needed close to the virus surface.

Whenever a polymer molecule interacts with the surface, it is either reflected or (irreversibly) adsorbed. The chemisorption is modeled by a random sequential adsorption (RSA) algorithm [7]: We check whether the corresponding binding site on the surface is free and then the reaction occurs with a certain probability. This probability is related to the reaction rate constant of the binding reaction as given in [8]. The reader can find more details about the model in [9]. In this paper, we show that the PBD algorithms can be used to compute the results from [9]. We will use the same parameters, namely $D = 5 \times 10^{-5} \text{ mm}^2 \text{ s}^{-1}$, $k_d = 1.3 \times 10^{-4} \text{ s}^{-1}$, and $\Delta t = 0.01 \text{ s}$. Then the mean displacement per time step according to (2.5) is $\sqrt{2D\Delta t} = 10^{-3} \text{ mm}$, and we therefore choose the size of the overlap region of the PBD algorithm (B1)–(B5) as $|O| = 10^{-2} \text{ mm}$. From the results in [9], we estimate that a maximum length of $L = 2 \text{ mm}$ is enough to simulate the binding process and use the Dirichlet boundary condition

$$n(L, t) = c_0 \exp(-k_d t),$$

where $c_0 = 1.2 \times 10^4 \text{ molecules/mm}$ is the initial concentration of molecules (i.e., $n(z, 0) \equiv c_0$ for $z \in \Omega$). To apply the PBD algorithm (B1)–(B5), we choose $\Omega_B = [0, 1.01) \text{ mm}$, $O = (1, 1.01) \text{ mm}$, and $\Omega_P = (1, 2) \text{ mm}$. The RSA algorithm was performed using a nearest neighbor exclusion on a 100×100 grid of receptor binding positions on the surface [7].

In Figure 5.3 we plot the concentration profile inside Ω_B of a single simulation at two different times (gray histograms). We compare the results of the PBD algorithm

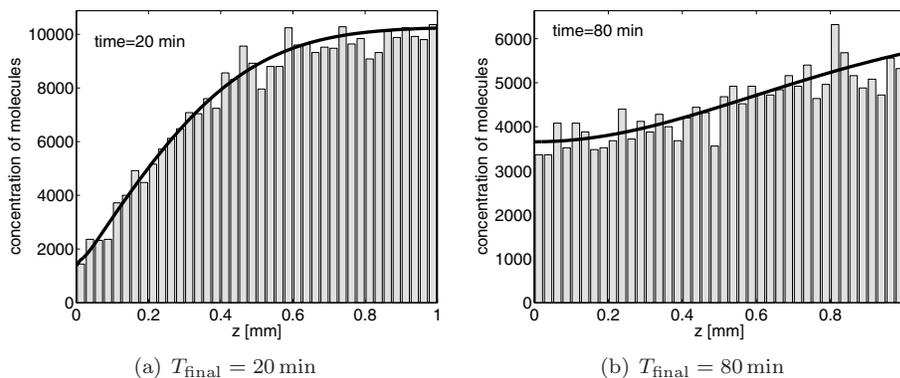


FIG. 5.3. Example 3 (chemisorption to virus surface). Gray histograms: concentration profile in molecules/mm at two given times computed by the PBD algorithm (B1)–(B5); solid line: results of the RSA-PDE model presented in [9]. Parameters as shown in the text.

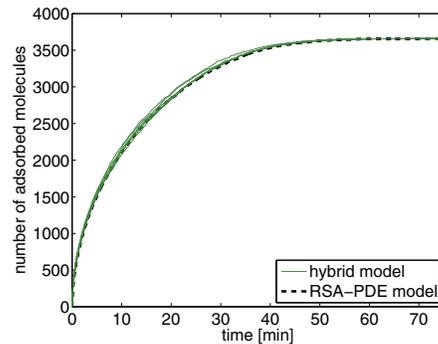


FIG. 5.4. *Example 3 (chemisorption to virus surface). Number of polymer molecules which are bound to the virus surface as a function of time. Solid lines (green in the electronic version): six realizations computed by the PBD algorithm (B1)–(B5); dashed line: results of the RSA-PDE model presented in [9]. Parameters as shown in the text.*

(B1)–(B5) with the results of the RSA-PDE model presented in [9] (black lines). As shown in [9], the RSA-PDE model also compares well with the full BD simulation. The number of molecules which are attached to the surface as a function of time is plotted in Figure 5.4 (six realizations of the PBD algorithm (B1)–(B5) are plotted as green solid lines). Again, we see an excellent agreement with the result from [9] which is plotted as the black dashed line.

6. Discussion. In this paper we have presented two PBD algorithms that combine Brownian dynamics with mean-field reaction-diffusion PDEs. This method produces exact Brownian dynamics simulations in one part of the domain and couples them with mean field approximations in another part of the domain. An algorithm of this type is useful for various application areas in computational biology and beyond, for example, when a detailed description of individual molecules is required near a receptor or ion channel but becomes impractical in the bulk of a cell [5, 16]; or when a detailed stochastic simulation of actin dynamics is required inside filopodia but becomes impractical in the bulk of a cell [33]. Another application area, chemisorption, was discussed in section 5.3. By using our approach it would also be possible to use finite-sized particles in the BD simulation and couple these with the corresponding mean field results presented in [3].

In the literature several hybrid models have been developed in the context of fluid dynamics, but they do not discuss issues that arise from the incorporation of chemical reactions. Alexander, Garcia, and Tartakovsky [1] presents a hybrid model that uses virtual particles at the grid point nearest the interface to calculate fluxes across the boundary and to generate accurate density fluctuations inside the particle region. Reference [31] extends this approach by the introduction of an overlap region similar to O introduced in section 4. An identical flux exchange with particles confined to a grid is presented in [17]. The principal disadvantage of these flux exchange schemes is that the resolution used for the numerical solution of the PDE regime needs to be bigger than the microscopic length-scale of the particles [31], whereas the PBD algorithms presented in this paper work independently of the numerical scheme used in Ω_P . Additionally, the PBD algorithms guarantee exact conservation of mass and nonnegativity of concentrations even for very small numbers of particles.

Chemical reactions in the solution were considered in [9]. This model was discussed in section 5.3. It couples Brownian dynamics of molecules in the solution with a more detailed description of the adsorbing boundary. In [9] a hybrid (RSA-PDE) model has been developed which replaces BD in the solution by solving the PDE (5.8) with a suitable stochastic boundary condition. The PBD algorithms are able to replace the stochastic boundary condition by a (small) BD region close to the surface. Although the hybrid RSA-PDE model introduced in [9] was sufficient in the case of (irreversible) adsorption, the situation is becoming more challenging whenever the binding reaction is reversible [21]. In this case, a molecule which is released by the surface will initially stay close to the surface and can rebind to the same receptor (binding site). This geminate recombination can be captured by the PBD approach. Reversible reactions are common in biological applications [21, 16].

Hybrid approaches for reaction-diffusion processes which couple different modeling approaches have also been introduced in the literature [23, 15, 6, 13]. A mesoscopic lattice-based description coupled with macroscopic Fisher–Kolmogorov–Petrovsky–Piscounov PDE was used in [23] to study front propagation in a lattice-based reaction-diffusion model. A hybrid model for reaction-diffusion systems in porous media that combines pore-scale models with Darcy-scale models is presented in [27]. Flegg, Chapman, and Erban [15] introduced the so-called Two Regime Method which couples a lattice-based (compartment-based) reaction-diffusion model with BD simulations. One advantage of the PBD algorithms over the Two Regime Method is that there are more efficient tools for PDE simulations than for compartment-based reaction-diffusion models. On the other hand, compartment-based models provide more details (including fluctuations) and hybrid models which couple BD simulations with compartment-based models do not require the overlap region [15, 13]. Since it is possible to couple the macroscopic PDE description with (mesoscopic) compartment-based models [6] and compartment-based models with (microscopic) BD simulations [15], then an alternative approach to PBD algorithms would be to use compartment-based models in the overlap region. That is, the computational domain would be divided into three regions where the PDE, compartment-based, and BD descriptions would be used. These three regimes would be coupled using the results from the literature [6, 15]. Compartment-based models and macroscopic PDEs can also be coupled through another intermediate regime using a tau-leap method [13]. In this paper, we showed that PDE models and BD simulations can be coupled without using intermediate compartment-based models.

The PBD algorithms presented in this paper should be seen as a first step towards a more general setting. Some parts of the algorithm (B1)–(B5) extend easily (at least theoretically) into higher dimensions, but in practice additional difficulties arise. One example is the need to sample from a multidimensional probability distribution to find the position of newly created molecules. Additionally, in higher dimensions one can also expect to deal with higher order reactions, including bimolecular reactions. For a discussion of how to implement bimolecular reactions for BD simulations, we refer to [10], but the real problem occurs inside the overlap region $O = \Omega_B \cap \Omega_P$, where a molecule could react with another molecule, or with the continuum. Since the reaction-diffusion PDEs are solved numerically using a suitable mesh, it is important to study methods for coupling individual molecules with the numerical discretization of macroscopic PDEs [18]. For the ion channel application mentioned before, one also needs to think about how to incorporate electrical charges and resulting forces into the system. One can imagine that these forces act as a boundary condition on the continuum model and as an effective force on the particles.

REFERENCES

- [1] F. J. ALEXANDER, A. L. GARCIA, AND D. M. TARTAKOVSKY, *Algorithm refinement for stochastic partial differential equations*, J. Comput. Phys., 182 (2002), pp. 47–66.
- [2] S. ANDREWS AND D. BRAY, *Stochastic simulation of chemical reactions with spatial resolution and single molecular detail*, Phys. Biol., 1 (2004), pp. 137–151.
- [3] M. BRUNA AND S. J. CHAPMAN, *Excluded-volume effects in the diffusion of hard spheres*, Phys. Rev. E, 85 (2012), 011103.
- [4] W. CHEN, R. ERBAN, AND S. J. CHAPMAN, *From Brownian dynamics to Markov chain: An ion channel example*, SIAM J. Appl. Math., submitted.
- [5] B. CORRY, S. KUYUCAK, AND S. CHUNG, *Tests of continuum theories as models of ion channels. I. Poisson-Nernst-Planck theory versus Brownian dynamics*, Biophys. J., 78 (2000), pp. 2364–2381.
- [6] S. ENGBLOM, L. FERM, A. HELLANDER, AND P. LÖTSTEDT, *Simulation of stochastic reaction-diffusion processes on unstructured meshes*, SIAM J. Sci. Comput., 31 (2009), pp. 1774–1797.
- [7] R. ERBAN AND S. J. CHAPMAN, *On chemisorption of polymers to solid surfaces*, J. Stat. Phys., 127 (2007), pp. 1255–1277.
- [8] R. ERBAN AND S. J. CHAPMAN, *Reactive boundary conditions for stochastic simulations of reaction-diffusion processes*, Phys. Biol., 4 (2007), pp. 16–28.
- [9] R. ERBAN AND S. J. CHAPMAN, *Time scale of random sequential adsorption*, Phys. Rev. E, 75 (2007), 041116.
- [10] R. ERBAN AND S. J. CHAPMAN, *Stochastic modelling of reaction-diffusion processes: Algorithms for bimolecular reactions*, Phys. Biol., 6 (2009), 046001.
- [11] R. ERBAN, S. J. CHAPMAN, K. D. FISHER, I. G. KEVREKIDIS, AND L. W. SEYMOUR, *Dynamics of polydisperse irreversible adsorption: A pharmacological example*, Math. Models Methods Appl. Sci., 17 (2007), pp. 759–781.
- [12] R. ERBAN, S. J. CHAPMAN, AND P. MAINI, *A practical guide to stochastic simulations of reaction-diffusion processes*, available online at <http://arxiv.org/abs/0704.1908>, 2007.
- [13] L. FERM, A. HELLANDER, AND P. LÖTSTEDT, *An adaptive algorithm for simulation of stochastic reaction-diffusion processes*, J. Comput. Phys., 229 (2010), pp. 343–360.
- [14] K. FISHER, Y. STALLWOOD, N. GREEN, K. ULBRICH, V. MAUTNER, AND S. L., *Polymer-coated adenovirus permits efficient retargeting and evades neutralising antibodies*, Gene Therapy, 8 (2001), pp. 341–348.
- [15] M. B. FLEGG, S. J. CHAPMAN, AND R. ERBAN, *The two regime method for optimizing stochastic reaction-diffusion simulations*, Royal Society Interface, 9 (2012), pp. 859–868.
- [16] M. B. FLEGG, S. RUEDIGER, AND R. ERBAN, *Diffusive spatio-temporal noise in a first-passage time model for intracellular calcium release*, J. Chem. Phys., 138 (2013), 154103.
- [17] E. FLEKKØY, J. FEDER, AND G. WAGNER, *Coupling particles and fields in a diffusive hybrid model*, Phys. Rev. E, 64 (2001), 066302.
- [18] B. FRANZ AND R. ERBAN, *Hybrid modelling of individual movement and collective behaviour*, in Dispersal, Individual Movement and Spatial Ecology: A Mathematical Perspective, M. Lewis, P. Maini, and S. Petrovskii, eds., Lecture Notes in Math. 2071, Springer, Heidelberg, 2013.
- [19] J. HATTNE, D. FANGE, AND J. ELF, *Stochastic reaction-diffusion simulation with MesoRD*, Bioinformatics, 21 (2005), pp. 2923–2924.
- [20] M. HOWARD, *How to build a robust intracellular concentration gradient*, Trends in Cell Biology, 22 (2012), pp. 311–317.
- [21] J. LIPKOVÁ, K. ZYGALAKIS, S. J. CHAPMAN, AND R. ERBAN, *Analysis of brownian dynamics simulations of reversible bimolecular reactions*, SIAM J. Appl. Math., 71 (2011), pp. 714–730.
- [22] K. LIPKOW, S. ANDREWS, AND D. BRAY, *Simulated diffusion of phosphorylated chey through the cytoplasm of escherichia coli*, J. Bacteriol., 187 (2005), pp. 45–53.
- [23] E. MORO, *Hybrid method for simulating front propagation in reaction-diffusion systems*, Phys. Rev. E, 69 (2004), 060101.
- [24] G. MOY, B. CORRY, S. KUYUCAK, AND S. CHUNG, *Tests of continuum theories as models of ion channels. i. Poisson-Boltzmann theory versus Brownian dynamics*, Biophys. J., 78 (2000), pp. 2349–2363.
- [25] J. MURRAY, *Mathematical Biology II: Spatial Models and Biomedical Applications*, 3rd ed., Springer-Verlag, New York, 2003.
- [26] K. TAKAHASHI, S. TANASE-NICOLA, AND P. TEN WOLDE, *Spatio-temporal correlations can drastically change the response of a MAPK pathway*, Proc. Natl. Acad. Sci. USA, 107 (2010), pp. 2473–2478.

- [27] A. M. TARTAKOVSKY, D. M. TARTAKOVSKY, T. D. SCHEIBE, AND P. MEAKIN, *Hybrid simulations of reaction-diffusion systems in porous media*, SIAM J. Sci. Comput., 30 (2008), pp. 2799–2816.
- [28] F. TOSTEVIN, P. TEN WOLDE, AND M. HOWARD, *Fundamental limits to position determination by concentration gradients*, PLOS Comput. Biol., 3 (2007), pp. 763–771.
- [29] J. VAN ZON AND P. TEN WOLDE, *Simulating biochemical networks at the particle level and in time and space: Green's function reaction dynamics*, Phys. Rev. Lett., 94 (2005), p. 128103.
- [30] J. VON NEUMANN, *Mathematical Foundations of Quantum Mechanics*, Princeton University Press, Princeton, NJ, 1955.
- [31] G. WAGNER AND E. FLEKKØY, *Hybrid computations with flux exchange*, Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 362 (2004), pp. 1655–1665.
- [32] L. WOLPERT, R. BEDDINGTON, T. JESSEL, P. LAWRENCE, E. MEYEROWITZ, AND J. SMITH, *Principles of Development*, Oxford University Press, Oxford, UK, 2002.
- [33] P. ZHURAVLEV AND G. PAPOIAN, *Molecular noise of capping protein binding induces macroscopic instability in filopodial dynamics*, Proc. Natl. Acad. Sci., 106 (2009), pp. 11570–11575.