EFFICIENT SAMPLING IN EVENT-DRIVEN ALGORITHMS FOR REACTION-DIFFUSION PROCESSES

HOSSEIN BANI-HASHEMIAN[1], STEFAN HELLANDER[1], PER LÖTSTEDT[1] *

October 27, 2011

[1] *Division of Scientific Computing, Department of Information Technology*
Uppsala University, P. O. Box 337, SE-75105 Uppsala, Sweden
emails: m.h.banihashemian@gmail.com, stefan.hellander@it.uu.se, perl@it.uu.se

Abstract

In event-driven algorithms for simulation of diffusing, colliding, and reacting particles, new positions and events are sampled from the cumulative distribution function (CDF) of a probability distribution. The distribution is sampled frequently and it is important for the efficiency of the algorithm that the sampling is fast. The CDF is known analytically or computed numerically. Analytical formulas are sometimes rather complicated making them difficult to evaluate. The CDF may be stored in a table for interpolation or computed directly when it is needed. Different alternatives are compared for chemically reacting molecules moving by Brownian diffusion in two and three dimensions. The best strategy depends on the dimension of the problem, the length of the time interval, the density of the particles, and the number of different reactions.

Keywords: event-driven algorithm, chemical reactions, diffusion, efficient sampling.

AMS subject classification: 65C05, 65C35, 82C80.
PACS subject classification: 02.50.Ey, 02.70.Uu, 83.10.Rs, 87.10.Rt.

1 Introduction

It is of interest in many fields of physics, chemistry, biology, and medicine to simulate the dynamical evolution of particles diffusing independently in three dimensional space according to Brownian dynamics and interact with each other

*Financial support has been obtained from the Swedish Research Council. Corresponding author: Per Lötstedt, address as above, telephone +46-18-4712972, fax +46-18-523049.

when they are adjacent to each other in kinetic Monte Carlo (KMC) algorithms. Each particle is tracked individually and they may coalesce with a certain probability when they collide or are close together and can split into two products. Examples are found e.g. in [1, 5, 8, 12, 13, 16, 17].

The simulation technique is usually categorized as either *time-driven* or *event-driven* [7]. In a time-driven algorithm, the particles in the system are advanced in time by small time steps Δt, processing events such as collisions and reactions at the end of the time step. The time step is usually longer in an event-driven simulation where Δt is the time between two events in the system but there is also more computational work in each step. The situation is somewhat similar in the numerical integration of stiff ordinary differential equations. An explicit method is easy to program but needs small Δt for stability and an implicit method is more complicated but allows longer Δt. A new event or the new position of a particle is found by sampling a probability distribution. This distribution is Gaussian for a particle in free space but for two particles in the neighborhood of each other with a risk of collision or reaction between them, the distribution is more complicated. The cumulative distribution function (CDF) is required for the sampling using the inverse transform sampling algorithm and it is sometimes known analytically. Since the number of time steps is large and there may be many particles, the sampling has to be computationally efficient.

In this paper, we compare different ways of sampling the distribution in an event-driven process for simulation of biochemical reactions and diffusion. The molecules are assumed to be hard and spherical and are partitioned into single molecules and pairs of molecules as in the Green's Function Reaction Dynamics (GFRD) algorithm by van Zon and ten Wolde [19]. The mathematical model for the diffusing and reacting molecules was proposed by Smoluchowski [14]. The probability density function (PDF) for the positions of two molecules in a pair satisfies a parabolic partial differential equation (PDE). The boundary conditions for the PDE are given by Collins and Kimball [6]. If the molecules react, then an associative event has occurred and a new molecule is created. Another event is when a molecule dissociates into two molecules. Single molecules move inside protective spheres where the risk of collision with other molecules is very small.

While random sampling for a single molecule is simple, it is more complicated for a pair of molecules. The center coordinates of a pair are found by sampling a normal distribution and the PDF of the distance between the molecules is determined by the Smoluchowski equation. The differential operator in this PDE is split into two or three parts corresponding to the coordinate directions in [9]. The sampling is considerably simplified in this way at the cost of a numerical splitting error. The CDF in each step may be known analytically or can be computed numerically from an analytically known or numerically determined PDF. If the CDF is expensive to compute every time it is needed, then it can be tabulated and the relative position of the molecules in the pair is computed by interpolation in the table. In principle, these are the alternatives for the

computational procedure in every event-driven algorithm based on sampling of a CDF.

Different strategies are evaluated for sampling of the radial distance and the angular direction between two molecules in two and three dimensions (2D and 3D). The accuracy in the alternatives is at the same level and the computing time is measured and compared. The algorithms are implemented in MATLAB and executed on a laptop. The difference is significant in some cases. If the CDF is tabulated, then there is an initial cost to compute the table but the sampling using interpolation in a table is much cheaper than computing the CDF several times for each sampling point. Depending on the length of the time interval and the number of molecules, the cost of generating the table may or may not be justified.

The contents of the paper are organized as follows. In the next section, the Smoluchowski equation and the boundary conditions for the PDF are presented. The operator splitting and the different solution methods for the PDE in spherical and cylindrical coordiantes are discussed in Section 3. The efficiency of the sampling techniques is investigated in numerical experiments in Section 4. The best approach for 3D simulation is chosen in Section 5 to verify the correctness of the algorithm. Conclusions are drawn in Section 6. More details and data for a sphercial coordinate system are found in [2].

2 The Smoluchowski equation

Consider a reversible association and dissociation reaction

$$A + B \underset{k_d}{\overset{k_a}{\rightleftharpoons}} C, \tag{2.1}$$

where k_a and k_d are the *association* and *dissociation rates*, respectively. Depending on the value of k_a, A and B may immediately react with each other to form molecule C when their distance is equal to the *reaction radius* σ. Dissociation of C into A and B occurs with the rate k_d. We assume that molecule A is at a fixed position in 3D at the origin of a spherical coordinate system (r, θ, ϕ) and along the z-axis in a cylindrical coordinate system (r, ϕ, z). Molecule B is free to move randomly in space by diffusion with a *diffusion coefficient* D, which is defined to be the sum of the diffusion coefficients in free space of A and B. In the spherical system, both molecules are sphere-shaped and in the cylindrical system, A is cylindrical and B is spherical.

2.1 Spherical coordinates

Let $\mathbf{r} = (r, \theta, \phi)^{\mathbf{T}}$ be the position vector of a point in 3D. The time dependent conditional PDF of the molecule B to be at position $\mathbf{r}$ at time t starting at

position $\mathbf{r}_0$ at time $t_0 < t$, $p_\mathbf{r}(\mathbf{r}, t|\mathbf{r}_0, t_0)$, satisfies the Smoluchowski equation

$$\partial_t p_\mathbf{r} = D\left(\frac{\partial^2 p_\mathbf{r}}{\partial r^2} + \frac{2}{r}\frac{\partial p_\mathbf{r}}{\partial r} + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial p_\mathbf{r}}{\partial\theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 p_\mathbf{r}}{\partial\phi^2}\right), \quad (2.2)$$

with initial condition

$$p_r(\mathbf{r}, t_0|\mathbf{r}_0, t_0) = \delta(\mathbf{r} - \mathbf{r}_0), \quad (2.3)$$

and boundary conditions

$$\lim_{r\to\infty} p_\mathbf{r}(r, \theta, \phi, t|\mathbf{r}_0, t_0) = 0, \quad (2.4)$$

$$4\pi\sigma^2 D\frac{\partial p_\mathbf{r}}{\partial r}\bigg|_{r=\sigma} = k_a p_\mathbf{r}(\mathbf{r}, t|\mathbf{r}_0, t_0)|_{r=\sigma} - k_d(1 - S(t|\mathbf{r}_0, t_0)). \quad (2.5)$$

The probability of the molecules being bound together as C at t is $p_r(*, t|\mathbf{r}_0, t_0)$ and is computed via

$$p_\mathbf{r}(*, t|\mathbf{r}_0, t_0) = \int_{t_0}^{t}\int_0^{2\pi}\int_0^{\pi}\sigma^2\sin\theta D\frac{\partial p_\mathbf{r}(\mathbf{r}, \tau|\mathbf{r}_0, t_0)}{\partial r}\bigg|_{r=\sigma} d\theta d\phi d\tau. \quad (2.6)$$

The survival probability S or the probability that no reaction occurs between the molecules in $[t_0, t]$ is defined as follows:

$$\begin{aligned} S(t|\mathbf{r}_0, t_0) &= 1 - p_r(*, t|\mathbf{r}_0, t_0) \\ &= \int_\sigma^\infty\int_0^{2\pi}\int_0^{\pi}\rho^2\sin\theta p_\mathbf{r}(\rho, \theta, \phi, t|\mathbf{r}_0, t_0)\, d\theta d\phi d\rho. \end{aligned} \quad (2.7)$$

In the case of an irreversible reaction, $k_d = 0$, the boundary condition (2.5) is simplified to

$$4\pi\sigma^2 D\frac{\partial p_\mathbf{r}}{\partial r}\bigg|_{r=\sigma} = k_a p_\mathbf{r}(\mathbf{r}, t|\mathbf{r}_0, t_0)|_{r=\sigma}. \quad (2.8)$$

When $k_a = 0$ in (2.8) we have a homogeneous Neumann boundary condition at $r = \sigma$ and there is no associative reaction and B bounces away from A and as $k_a \to \infty$, we have absorption and $p_\mathbf{r}(\mathbf{r}, t|\mathbf{r}_0, t_0)|_{r=\sigma} = 0$.

2.2 Cylindrical coordinates

Consider an elongated A molecule such as a polymer or a molecular chain along the polar z axis in 3D in a cylindrical coordinate system. The position of B is at (r, ϕ, z) and the Smoluchowski equation for the PDF is

$$\partial_t p_\mathbf{r} = D\left(\frac{\partial^2 p_\mathbf{r}}{\partial r^2} + \frac{1}{r}\frac{\partial p_\mathbf{r}}{\partial r} + \frac{1}{r^2}\frac{\partial^2 p_\mathbf{r}}{\partial\phi^2} + \frac{\partial^2 p_\mathbf{r}}{\partial z^2}\right). \quad (2.9)$$

The initial condition is (2.3) and the solution vanishes when $r \to \infty$ as in (2.4). At $r = \sigma$ we consider only association of B with A

$$2\pi\sigma D\frac{\partial p_\mathbf{r}}{\partial r}\bigg|_{r=\sigma} = k_a p_\mathbf{r}(\mathbf{r}, t|\mathbf{r}_0, t_0)|_{r=\sigma}. \quad (2.10)$$

3 Position sampling techniques

The Smoluchowski PDEs (2.2) and (2.9) are solved using operator splitting or fractional steps and the resulting CDFs after each splitting step are sampled consecutively in the coordinate directions in [9]. Very complicated formulas for the analytical solution, see e.g. [4, 19], and high dimensional interpolation in tables are avoided in this way. A time step Δt is chosen and a straightforward application of the splitting technique introduces an error of $\mathcal{O}(\Delta t)$ in the PDE solution. A refined splitting due to Strang reduces the error to $\mathcal{O}(\Delta t^2)$ [15].

The differential operator in the right hand sides of (2.2) and (2.9) is decomposed into two or three parts. The solution is determined analytically or numerically and integrated analytically or numerically to obtain the CDF in the first coordinate direction. Then a position X is sampled in that direction using the CDF $F(X)$ and inverse transform sampling. Let ξ be uniformly distributed in $[0, 1]$. Then X satisfies $\xi = F(X)$. The procedure is then repeated for the remaining directions.

3.1 Spherical coordinates

The splitting steps in spherical coordinates are:

(i) a *radial* part with derivatives only in the radial direction:

$$\partial_t p_r = D\left(\frac{\partial^2 p_r}{\partial r^2} + \frac{2}{r}\frac{\partial p_r}{\partial r}\right), \tag{3.1}$$

with initial condition

$$p_r(r, t_0|r_0, t_0) = \delta(r - r_0), \tag{3.2}$$

a vanishing solution when $r \to \infty$ and the boundary condition

$$4\pi\sigma^2 D\frac{\partial p_r}{\partial r} = k_a p_r(\sigma, t|r_0, t_0) - k_d(1 - S(t|r_0, t_0)), \tag{3.3}$$

at $r = \sigma$ with the survival probability S defined in (2.7). Determine the CDF and sample for r at $t_0 + \Delta t$.

(ii) an *angular* part with derivatives in the polar and azimuthal directions:

$$\partial_t p_\theta = \frac{D}{r^2}\left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial p_\theta}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 p_\theta}{\partial\phi^2}\right), \tag{3.4}$$

with $\theta \in [0, \pi]$, $\phi \in [0, 2\pi)$ and the initial condition

$$p_\theta(\theta, \phi, t_0|r, 0, \phi_0, t_0) = \delta(\theta). \tag{3.5}$$

The solution p_θ is independent of ϕ. Thus, compute the CDF and sample it for the new θ. The new azimuthal angle at $t_0 + \Delta t$ is obtained by sampling the uniform distribution.

3.1.1 Solution and sampling in the radial direction

Analytical solution

In [10], an analytical solution to (3.1) - (3.3) is derived. By looking for a solution of the form $p_r = f(r, t|r_0) + g(r, t)$ where f is the solution for reflective boundary conditions they obtain

$$\begin{aligned} p_r(r, t|r_0, t_0)4\pi r r_0\sqrt{D} &= \frac{1}{\sqrt{4\pi t}}\left\{ \exp\left(-R_1^2\right) + \exp\left(-R_2^2\right) \right\} \\ &+ B(\alpha, \beta, \gamma, t) + B(\beta, \gamma, \alpha, t) + B(\gamma, \alpha, \beta, t), \end{aligned} \tag{3.6}$$

where

$$R_1 = \frac{(r - r_0)^2}{4Dt}, R_2 = \frac{r + r_0 - 2\sigma}{\sqrt{4Dt}}, \tag{3.7}$$

$$B(\alpha, \beta, \gamma, t) = \frac{\alpha(\gamma + \alpha)(\alpha + \beta)}{(\gamma - \alpha)(\alpha - \beta)} \exp(2ab + b^2) \operatorname{erfc}(a + b). \tag{3.8}$$

The complex complementary error function is denoted by erfc and $-\alpha$, $-\beta$ and $-\gamma$ are the roots of the equation

$$\sigma x^3 + \sqrt{D}\left(1 + \frac{k_a}{k_D}\right) x^2 + \sigma k_d x + \sqrt{D} k_d = 0 \tag{3.9}$$

with $k_D = 4\pi\sigma D$. In [11] they show that the CDF, F_{p_r}, is given by

$$\begin{aligned} F_{p_r}(r, t|r_0, t_0) + (1 - S(t|r_0, t_0)) &= -\frac{\sqrt{Dt}}{r_0\sqrt{\pi}} \left\{\exp\left(R_1^2\right) - \exp\left(-R_2^2\right)\right\} \\ &+ \frac{1}{2}\left\{\operatorname{erf}(R_1) + \operatorname{erf}(R_2)\right\} + E(\alpha)B(\alpha, \beta, \gamma, t) + E(\beta)B(\beta, \gamma, \alpha, t) \\ &+ E(\gamma)B(\gamma, \alpha, \beta, t), \end{aligned} \tag{3.10}$$

where erf is the error function, $E(x) = \frac{1}{r_0 x}\left(r - \frac{\sqrt{D}}{x}\right)$, and $S(t|r_0, t_0)$ is the survival probability in (2.7).

Given two molecules at a distance r_0 at time t_0 we can now sample the new distance r at time t with inverse transform sampling by inverting the CDF numerically. In Figure 3.1 we see how the PDF and CDF change with different values of r_0. If the uniformly distributed $\xi > \max_r F_{p_r}$, then a reaction between A and B has occurred and they are replaced by C.

Numerical solution

To solve the problem (3.1) - (3.3) numerically, we use the Crank-Nicolson finite difference scheme to obtain p_i^n at the i^{th} grid point and time t^n because of its

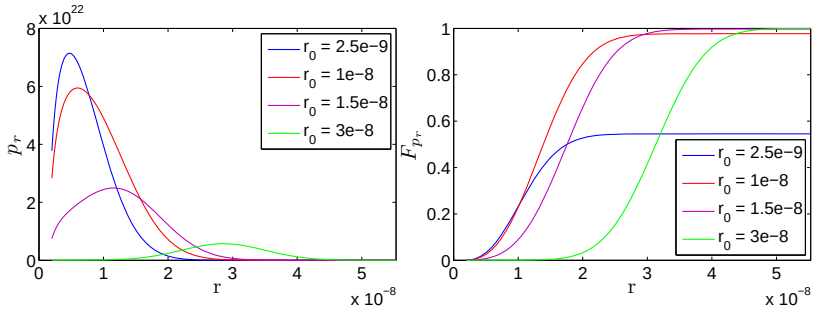

Figure 3.1: The PDF and CDF in the radial direction for different values of r_0. The model parameters are $k_a = 10^{-19}$, $k_d = 2.9894$, $D = 2 \cdot 10^{-12}$, $\sigma = 2 \cdot 10^{-9}$, $\Delta t = 10^{-5}$.

stability and second order accuracy in time and space. The grid is non-uniform in space with step size h_j, always containing the initial position r_0, and uniform in time with time step $\Delta\tau < \Delta t$. The initial condition (3.2) is approximated by $p_j^0 = 0, j \neq i$, and $p_i^0 = 1/h_i$ where i is the index of r_0 in the spatial grid.

The boundary condition at infinity is replaced by the same condition at a suitably large finite distance $r_{\max}$. The distance a particle can travel can be approximated by that of a freely diffusing particle. Thus, a reasonable choice is $r_{\max} = c\sqrt{2D\Delta t}$ for some constant c. We have found that $c = 4$ is sufficiently large.

The boundary condition (3.3) is approximated at $t = t^n$ by

$$4\pi\sigma^2 D \, \frac{p_2^n - p_1^n}{h_2} = k_a p_1^n - k_d(1 - S(t^n)) \,, \qquad 1 \leqslant n \leqslant N \,, \tag{3.11}$$

where the survival probability $S(t^n)$ is integrated with the trapezoidal rule using p_i^{n-1}. Finally the CDF is approximated, again using the trapezoidal rule.

The most critical requirement for implementing the numerical scheme is that r_0 must be one of the discretization points in the r direction, due to its presence in the initial condition. In order to have such a grid in space, the subinterval $[\sigma, r_0]$ is first discretized into $M_{\text{left}} = 10$ points. Then using the same step size we discretize the subinterval $[r_0, r_{\max}]$. To obtain a sufficiently accurate solution we may then have to refine the grid to the left of r_0.

When the initial position r_0 is close to the reaction radius σ, the numerical solution exhibits a decaying oscillatory behaviour around r_0 unless $\Delta\tau$ is sufficiently small. Because of this the numerical method is not recommended when the molecules are located initially in or too close to the bound state. One way of resolving the issue with the oscillatory behaviour is to use an adaptive grid algorithm, but then a major effort is necessary for the implementation.

Interpolation in a precomputed lookup table

Using either of the methods above we precompute a 3D lookup table in $r_0, \Delta t$, and r for the CDF with fixed model parameters k_a, k_d, σ and D to be used for the approximate sampling of the distribution. This method involves two stages: creating the lookup table and performing interpolation between the data points in the table.

When creating the table we first choose n_{r_0} points in the direction of r_0 and n_t points in the direction of Δt. Then the CDF is computed at discrete r-values for every $n_{r_0} \cdot n_t$ possible combination of the r_0 and Δt values using either the numerical scheme, by computing the analytical solution given by (3.10) or by computing the analytical PDF given by (3.6) and then the CDF by numerical integration.

If the table is generated by one of the analytical formulas, we can use a fixed number of points in r-space, but when creating the table with the numerical method the spatial grid will depend on r_0. Then we need to keep an additional table with information about the points in the r-direction and their CDF values.

Now, for given r_0 and Δt, a new r is determined by first retrieving the four points in the table surrounding r_0 and Δt with their corresponding CDFs. Then a ξ uniformly distributed in $[0,\ 1]$ is sampled, and finally we compute r_ξ by performing trilinear interpolation in the table. The interpolation region, i.e. the table cell containing ξ from the CDF of interest, can have a very irregular shape. This is due to the difference in the curvature of the graphs of the four neighbouring CDFs, a part of which forms four sides of the cell. In Figure 3.2, the panel to the left represents such a cell in one of the worst cases.

A feasible interpolation technique for computing r_ξ is via two bilinear interpolations between the vertices of the faces ${}_1p_1p'p'_1p_1$ and ${}_2p_2p'p'_2p_2$ of the cell, see Figure 3.2, followed by one linear interpolation resulting in a trilinear interpolation, but in the general case the vertices do not have the same radial coordinate r. One way to handle this problem is to interpolate inside a regularly cube-shaped region circumscribing the original cell (as the right panel in Figure 3.2 illustrates). Then the new vertices ${}_1P$,${}_1P'$, P'_1, and P_1 have the same r coordinate equal to $r^{(1)} := \min\left\{r_{{}_1p}, r_{p_1}, r_{{}_1p'}, r_{p'_1}\right\}$, and the same situation holds for ${}_2P$,${}_2P'$, P'_2, and P_2 where $r = r^{(2)} := \max\left\{r_{{}_2p}, r_{p_2}, r_{{}_2p'}, r_{p'_2}\right\}$. The CDF values at these new points are also needed, which can be calculated by linear interpolation.

Now that all the conditions for performing the bilinear interpolations are satisfied, we find two points Q and R normally located on opposite sides of the target point, in the sense that they all belong to the same CDF and ξ is between the values of the cumulative distribution function at Q and R, or $F_Q \leq \xi \leq F_R$. Considering Q and R as interpolation points, r_ξ is obtained by linear interpolation. If ξ is not between F_Q and F_R, r_ξ is computed via linear extrapolation.

Here we create two lookup tables using 200×200 pairs of sampled r_0 and Δt values. The computed analytical CDF values are stored in one table and

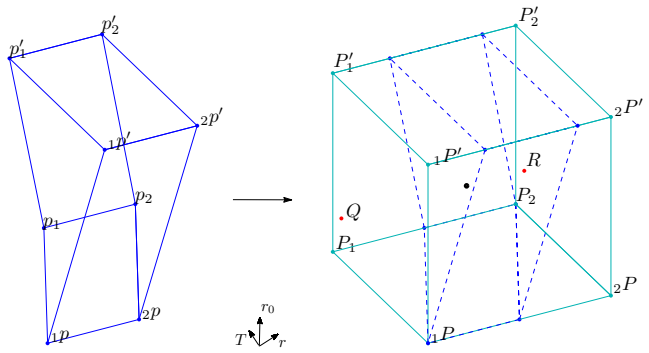

Figure 3.2: A cell from the 3D lookup table whose entries are the CDF values obtained from numerical method. This cell contains a data point with value $\xi = 0.9595$. The result from interpolation is shown for $r_0 = 8.8962 \cdot 10^{-8}$ and $\Delta t = 1.2558 \cdot 10^{-4}$, and the model parameters are fixed as $k_a = 10^{-19}$, $k_d = 2.9894$, $D = 2 \cdot 10^{-12}$, $\sigma = 2 \cdot 10^{-9}$.

the other one contains the numerical values. For fixed r_0 and Δt, we used the same spatial discretization in r for both the analytical and the numerical method. The number of time steps $\Delta\tau$ to reach Δt in the numerical scheme is changed according to a piecewise constant function of r_0 to accelerate the computations.

Interpolation without lookup table

When computing the CDF analytically or numerically for given r_0 and Δt without storing it in a table we still need to invert it numerically to perform inverse transform sampling. To this end the CDF is computed on a grid $r_1 < r_2 < \ldots < r_N$ and then for a given ξ we find i such that $F^{-1}(r_i) \leq \xi \leq F^{-1}(r_{i+1})$ by performing a linear search. Finally, to find r_ξ such that $F^{-1}(r_\xi) \approx \xi$ is found by inverse linear interpolation.

3.1.2 Solution and sampling in the angular direction

Applying finite difference schemes to solve the problem (3.4), (3.5), involves difficulties due to the singularity in the equation at $\theta = 0$. This is the reason why we will only compare computing the analytical solution directly with linear interpolation in a precomputed look-up table.

Analytical solution

The analytical solution of the angular part of the Smoluchowski equation (3.4), (3.5), is given in the form of an infinite series, see e.g. [18, 21],

$$p_\theta(\theta, t_0 + \Delta t | r, 0, \phi_0, t_0) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi r^2} \exp\left(-l(l+1)\frac{D\Delta t}{r^2}\right) P_l(\cos\theta)\,, \tag{3.12}$$

where P_l denotes the l^{th}-degree Legendre polynomial. Note that for a fixed radial distance r, the conditional PDF (3.12) is a function of the polar angle θ and time t and independent of the azimuthal angle ϕ. The convergence of the series in (3.12) can be verified analytically using standard techniques and the rate of convergence is mainly influenced by the value of the parameter $\hat{t} = D\Delta t/r^2$. The PDF for different values of $\hat{t}$ is plotted in Figure 3.3. As we can see, p_θ tends to a uniform distribution as $\hat{t}$ grows.

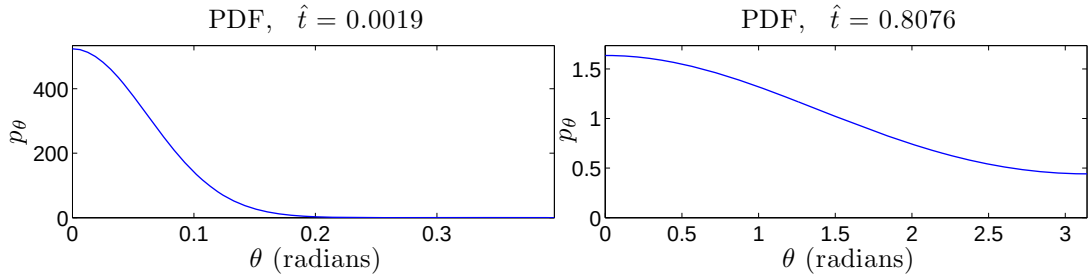

Figure 3.3: The PDF in the polar direction for two different values of $\hat{t}$.

For practical computations, the infinite series (3.12) has to be truncated to a finite summation. A possible choice of the level of truncation is the first positive integer $l_{\max}$ for which the absolute value of the remainder term

$$\mathcal{R} = \sum_{l=l_{\max}+1}^{\infty} (2l+1) \exp\left(-l(l+1)\hat{t}\right) P_l(\cos\theta) \tag{3.13}$$

is small. Since $|P_l(x)| \leq 1$ for any $x \in [-1, 1]$ and every positive integer l, we have

$$|\mathcal{R}| \leq \sum_{l=l_{\max}+1}^{\infty} (2l+1) \exp\left(-l(l+1)\hat{t}\right)\,. \tag{3.14}$$

Approximating the sum in (3.14) by an infinite integral and requiring that the integral is less than a small number ϵ, gives the following criterion

$$-(l_{\max}+1)(l_{\max}+2)\hat{t} \leq \ln \hat{t}\epsilon\,, \tag{3.15}$$

for truncating (3.12) at a proper level.

The criterion (3.14) is expressed in terms of $\hat{t}$ and for a fixed value of $\hat{t}$, we can estimate the PDF p_θ at a number of not necessarily equidistant grid points for $\theta \in [0, \pi]$. However, for $\hat{t}$ small p_θ decays quickly with growing θ and unnecessary computations are avoided by introducing another stopping criterion given by

$$|a_{l_{\max}}| \leq \epsilon' |s_{l_{\max}}| \,. \tag{3.16}$$

Here, ϵ' is small and positive, $a_{l_{\max}}$ is the $l_{\max}$-th term of the series (3.12), and $s_{l_{\max}}$ denotes the sum of the series after truncation at the index $l_{\max}$. When this criterion is met the PDF has stopped changing within some given relative tolerance ϵ'.

In Figure 3.4, the total number of terms needed to reach the stopping criterion (3.15) with $\epsilon = 10^{-5}$ is plotted against θ and the natural logarithm of $\hat{t}$. The series converges slowly for small values of $\hat{t}$ (in the worst case we need 3049 terms) but when $\hat{t}$ is large the stopping criterion is fulfilled after only a few terms. However, when $\hat{t}$ is small the criterion in (3.16) is reached quickly and the number of points where p_θ is computed in the θ-direction is small. We have chosen $\epsilon' = 0.2$.

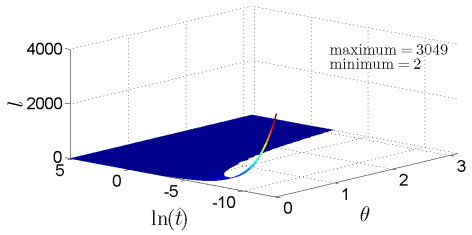

Figure 3.4: The number of terms $l = l_{\max} + 1$ of the infinite series (3.12) sufficient to reach convergence with $\epsilon = 10^{-5}$ are displayed as a function of θ and $\ln \hat{t}$. When $\ln \hat{t}$ is small the stopping criterion (3.16) is reached quickly, seen as the blank areas in the figure.

Having the PDF p_θ obtained as described above, the corresponding CDF is evaluated numerically using the trapezoidal rule and a polar position of a molecule can be sampled via inverse transform sampling.

Interpolation in a precomputed lookup table

The second approach employed to sample polar positions is to perform linear interpolation between precomputed CDF values tabulated in a 2D lookup table. This method consists of two stages: creating the lookup table and linear interpolation between the entries of the table.

The procedure of creating the CDF lookup table is as follows. Assume that we have a set of n_t preselected but not necessarily equally spaced values $\hat{t}_i$ in

a suitable range, sorted in increasing order. For every member of this set and on a fixed grid consisting of n_θ points in the polar direction, we compute the corresponding CDF via the method described in the previous section. Depending on the value of $\hat{t}$ and the stopping criteria, the n_t vectors of the precomputed CDF values may have different lengths of at most n_θ forcing us to introduce a data structure similar to what was proposed in Section 3.1.1 for the radial direction.

After creating the table, for any arbitrary value of r and Δt for which $\hat{t}_1 \leq D\Delta t/r^2 \leq \hat{t}_{n_t}$ we find the four surrounding data points forming a parallelogram-shaped region. A new polar position is then sampled from the CDF by bilinear interpolation.

3.2 Cylindrical coordinates

The dimensional splitting in cylindrical coordinates is:

(i) a *radial* part, with derivatives only in the radial direction:

$$\partial_t p_r = D\left(\frac{\partial^2 p_r}{\partial r^2} + \frac{1}{r}\frac{\partial p_r}{\partial r}\right), \tag{3.17}$$

with the same initial condition as in (3.2), a vanishing solution when $r \to \infty$, and a boundary condition at $r = \sigma$

$$2\pi\sigma D\frac{\partial p_r}{\partial r} = k_a p_r(\sigma, t|r_0, t_0). \tag{3.18}$$

Determine the CDF from p_r and sample for r at $t_0 + \Delta t$.

(ii) in the *axial* and *azimuthal* directions, the new positions are obtained by sampling normal distributions in the coordinates z and $r\phi$ with the angle chosen modulo 2π.

The solution of (3.17) is also the PDF of the relative distance between two molecules moving on a flat 2D membrane and ϕ is the angular direction.

3.2.1 Solution and sampling in the radial direction

Analytical solution

For the radial part of the Smoluchowski equation (3.17) and (3.18), the following exact analytical solution is available in [4, p. 370]:

$$p_r(r, t_0 + \Delta t|r_0, t_0) = \frac{1}{2\pi}\int_0^\infty \exp\left(-Du^2\Delta t\right)\mathcal{C}(u, r, r_0, k, k_a)u\,du\,, \tag{3.19}$$

$$\mathcal{C}(u, r, r_0, k, k_a) = C(u, r, k, k_a)C(u, r_0, k, k_a), \tag{3.20}$$

where $k = 2\pi l\sigma D$ and the function C is defined as a combination of Bessel functions of order n of the first and second kind J_n and Y_n, $n \in \{0, 1\}$,

$$C(u, r, k, k_a) = \frac{J_0(ur)\alpha(u) - Y_0(ur)\beta(u)}{(\alpha(u)^2 + \beta(u)^2)^{1/2}}\,, \tag{3.21}$$

$$\alpha(u) = kuY_1(\sigma u) + k_a Y_0(\sigma u), \quad \beta(u) = kuJ_1(\sigma u) + k_a J_0(\sigma u). \tag{3.22}$$

The CDF is given by

$$F_{p_r}(r, t_0 + \Delta t|r_0, t_0) = \int_\sigma^r 2\pi\rho\, p_r(\rho, t_0 + \Delta t|r_0, t_0)\, d\rho \tag{3.23}$$

and can be determined either by computing the exact PDF (3.19) and then the CDF by numerical integration or by using the analytical formula for the CDF directly for an approximation.

Computing the PDF

The integrand in (3.19) inherits the complex oscillatory behaviour of C, see Figure 3.5, but the term $ue^{-Du^2\Delta t}$ acts as a damping factor decaying rapidly after its maximum at $u = \sqrt{1/2D\Delta t}$. The domains of r and u are truncated in the calculations to the finite intervals $[\sigma, r_{\max}]$ and $[0, u_{\max}]$, respectively,

$$F_{p_r}(r, t_0 + \Delta t|r_0, t_0) \approx \int_\sigma^{r_{\max}} 2\pi\rho \int_0^{u_{\max}} ue^{-Du^2\Delta t}\, \mathcal{C}(u, r, r_0, k, k_a)\, du\, d\rho\,, \tag{3.24}$$

where $r_{\max} = r_0 + 4\sqrt{2D\Delta t}$ and $u_{\max} = \sqrt{\frac{1}{D\Delta t} \ln\left(\frac{1}{2\epsilon KD\Delta t}\right)}$ is chosen to satisfy the condition $K \int_{u_{\max}}^\infty ue^{-Du^2\Delta t}\, du \le \epsilon$ for a small positive ϵ assuming that $|\mathcal{C}(u, r, r_0, k, k_a)| \le K$. Due to the complicated form of $\mathcal{C}$, it is difficult to find an exact upper bound K. Nevertheless for our parameter values and based on observations from a set of graphs, $K = 1$ appears to be a quite satisfactory choice.

The outer integral in (3.24) is computed by the trapezoidal rule and for the inner integral, an adaptive quadrature scheme based on the Gauss-Kronrod rule is employed as implemented in MATLAB's built-in routine `quadgk`. Using algebraic transformations, `quadgk` can also handle the singularity of the integrand at $u = 0$.

Depending on the values of r_0 and Δt, the truncations and approximations explained above may result in oscillations in the computed PDF solution when $r < r_0$. There, the PDF remains at a constant level just before it starts growing to its peak. To lessen the effect of artificial oscillations, we replace the lower limit of the outer integral by $r_{\min} = \min\{\sigma, r_0 - 4\sqrt{2D\Delta t}\}$ and assume that if $r_{\min} > \sigma$ then $p_r(r, t|r_0, t_0) = p_r(r_{\min}, t|r_0, t_0) = 0$ for $r \in [\sigma, r_{\min}]$.

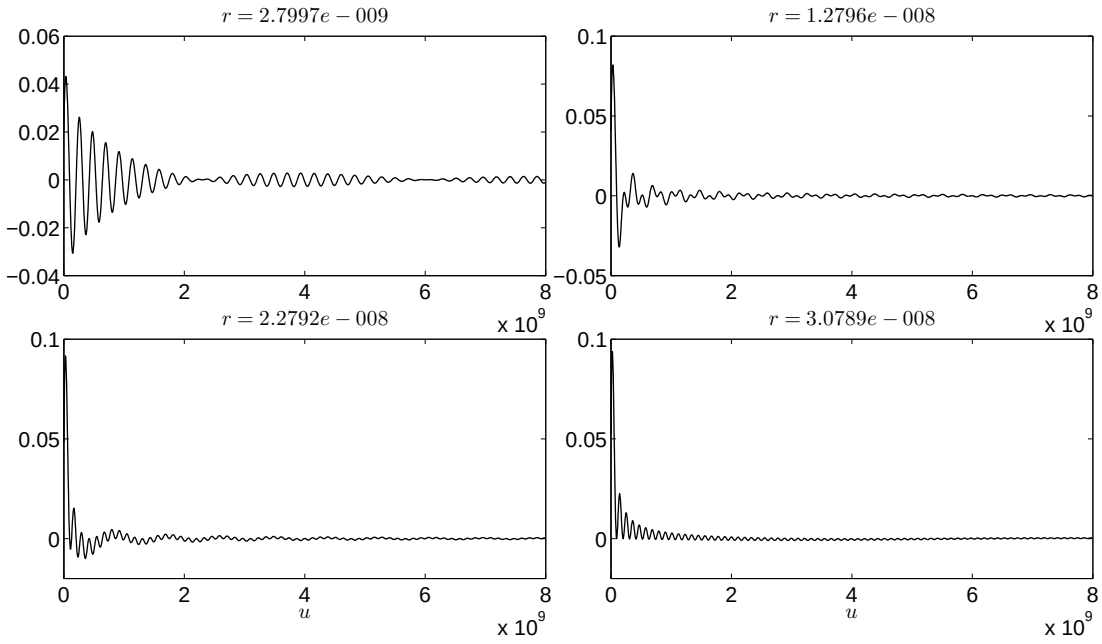

Figure 3.5: The oscillatory behaviour of the function $\mathcal{C}(u, r, r_0, k, k_a)$ for four different values of r with $r_0 = 3 \cdot 10^{-8}$ and $\Delta t = 10^{-7}$.

Computing the semi-analytical CDF

After truncating the r domain at a finite distance $r_{\max} = r_0 + 4\sqrt{2D\Delta t}$ and a change of the order of integration and using that $\int_a^b rA_0(r)dr = bA_1(b) - aA_1(a)$ with $A = J$ or Y we can rewrite (3.23) as

$$F_{p_r}(r, t|r_0, t_0) = \int_0^\infty e^{-Du^2\Delta t} R(u)\, du\,, \tag{3.25}$$

where

$$R(u) = \frac{C(u, r_0, k, k_a)\gamma(u)}{(\alpha(u)^2 + \beta(u)^2)^{1/2}}, \tag{3.26}$$

$$\gamma(u) = \alpha(u)(r_{\max}J_1(r_{\max}u) - \sigma J_1(\sigma u)) - \beta(u)(r_{\max}Y_1(r_{\max}u) - \sigma Y_1(\sigma u))\,. \tag{3.27}$$

Since no closed expression of (3.25) is known to us, we approximate the integral numerically. First, the upper limit of the integral is replaced by a finite value $u_{\max}$. The integrand exhibits oscillations, whose amplitudes tend to zero as u tends to infinity. Since the oscillations are difficult to predict, we assume that $|R(u)| \leq K'$. Choose $u_{\max}$ such that

$$\operatorname{erf}(u_{\max}\sqrt{D\Delta t}) \geq 1 - \tfrac{2\epsilon\sqrt{D\Delta t}}{K'\sqrt{\pi}} \tag{3.28}$$

and approximate the error function by

$$\operatorname{erf}(x) \approx \left(1 - \exp\left(-x^2 \, \frac{\frac{4}{\pi} + 0.14x^2}{1 + 0.14x^2}\right)\right)^{\frac{1}{2}} , \tag{3.29}$$

as in [20]. Then with $u_{\max}$

$$K' \int_{u_{\max}}^{\infty} e^{-Du^2\Delta t} \, du \leq \epsilon \, , \tag{3.30}$$

where $u_{\max}$ is the real positive root of a biquadratic equation of the form $c_1 u^4 + c_2 u^2 + c_3 = 0$, where $c_1 = 0.14 D\Delta t$, $c_3 = \frac{1}{D\Delta t} \ln\left(1 - \left(1 - \frac{2\epsilon\sqrt{D\Delta t}}{K'\sqrt{\pi}}\right)^2\right)$, $c_2 = \frac{4}{\pi} + c_1 c_3$ and $K' = 10^{-3}$.

Some artificial oscillations may appear in the calculated CDF for small r_0 but they are avoided by defining a lower integration limit $r_{\min} = \min\{\sigma, r_0 - 4\sqrt{2D\Delta t}\}$ as in the previous section.

The look-up table is precomputed either by first computing the PDF in (3.19) and then the CDF using the trapezoidal rule or using the analytic expression in (3.25) for the CDF. The new radial position is found by interpolation in the table as in Section 3.1.1.

4 Comparison of sampling methods

In this section, we compare the sampling methods in Section 3 with respect to accuracy and efficiency. To check the accuracy, we assume that the analytical methods yield the most accurate results, although some approximation errors are involved also there. Another assumption is that the quality of the coding of the algorithms is at the same level. It is possible that a comparison is altered by an improved implementation but in many cases the outcome is clear and the opposite conclusion is unlikely. We performed all the experiments in MATLAB v7.10 (R2010a) on a laptop PC with an Intel Core 2 Duo T7200 processor (2.00GHz, 4MB L2 Cache) and 2GB of RAM running Windows 7 (64 bit).

4.1 Spherical coordinates

The methods for sampling the distributions in spherical coordinates are evaluated using 92225 distinct pairs of values for the initial radial position r_0 and time step Δt. The values are collected from a simulation of one trajectory of a system using the method in [9]. In this way, we get a fair distribution of samples when comparing the overall accuracy and efficiency. For convenience we denote the set of the pairs of values by `PoVsph`. The model parameters in (2.2) and (2.5) are $k_a = 10^{-19}$, $k_d = 2.9894$, $D = 2 \cdot 10^{-12}$, and $\sigma = 2 \cdot 10^{-9}$.

4.1.1 The radial direction

The different methods in Section 3.1.1 to solve (3.1), (3.2), and (3.3) for the PDF and integration of it for the CDF are compared in measurements of their accuracy and efficiency.

Accuracy comparison

We begin by comparing the sampling methods with respect to accuracy. The comparison is made by applying all the methods over the whole set of values `PoVsph`. Very few members of the set whose r_0 values are very close to σ are excluded because then we are unable to use the numerical method. A uniformly distributed ξ is chosen. Then the relative error in per cent in computing r_ξ by inverse transform sampling is collected for all pairs of r_0 and Δt of the set. The error in a method is assumed to be the difference between r_ξ obtained with the method and r_ξ from the analytical method. To investigate the error data, the data are split into 21 groups according to the r_0 values and averaged over all Δt values in the group. The conclusions from the comparisons are the same if the data are instead averaged over the r_0 values.

The results are displayed using *box-and-whisker plots* (or *box-plots* for short) as demonstrated in Figure 4.1. In these plots, each box encloses the middle 50% of the data, i.e. the bottom and top edges of the box are the 25^{th} and 75^{th} quartiles, q_{25} and q_{75}, of the data, respectively. The *whiskers* or dashed T-bars display the highest and the lowest data points not considered to be outliers. With the interquartile range $I_{qr} = q_{75} - q_{25}$, the outliers are defined as values x such that $x < q_{25} - 1.5I_{qr}$ or $x > q_{75} + 1.5I_{qr}$. The line across the box represents the median and the cross, diamond, or asterisk in the box indicates the mean value which is computed excluding the outlier values.

From the box-plots in Figure 4.1, we find that the numerical method with the chosen mesh size is more accurate than the lookup table (lut) interpolation methods based on numerical or analytical data. Except for the leftmost r_0-interval, the lut-interpolation method performs quite well and the difference between the two versions of the lut-interpolation method is small. The two tables are stored on the same grid. Thus, the difference is due to the error from solving the equations numerically. For small values of r_0 or Δt, a large variation is noticed in the relative error due to a few data points in the table. The errors are *positively* or *right skewed* with the mean and the median being only a few per cent.

When two molecules stay close to each other for a while, then r_0 and Δt are small and errors may accumulate over several small time steps. This issue can be conclusively resolved by using high-resolution lookup tables containing more CDFs for small initial particle positions and small time intervals. Other sources of error that influence the accuracy of the lut-interpolation method are: the error in computing the entries of a lookup table and the error introduced

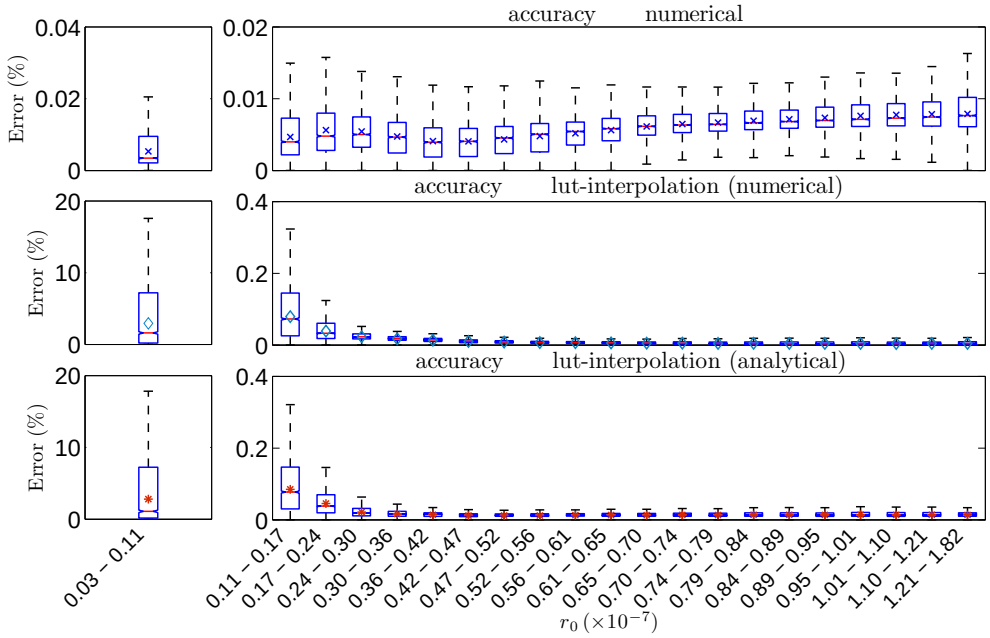

Figure 4.1: Box-plots of percent error in computing r_ξ in spherical coordinates using different sampling methods versus r_0. The experiment is performed for 91735 distinct pairs of values for r_0 and Δt with $\xi = 0.5469$.

by transforming the original region of interpolation in the lookup table into a regularly cube-shaped region (see Figure 3.2).

Efficiency comparison

The required CPU time to compute every single solution that we used in our accuracy experiment in the previous section is measured in Figure 4.2. The data are presented as in Figure 4.1. The accuracy of the methods is about the same in Figure 4.1 but the numerical method, with an average run time of 0.172 s, is noticeably less efficient than the other methods. The conclusion from the plots is that among the sampling methods the analytical method with numerical integration, which yields the sampled position on average within 2.24 ms, is the fastest one. This is approximately 77 times faster than the numerical method and 1.85 times faster than the lut-interpolation method with 4.16 ms average CPU time per run.

The speed of the numerical method is highly dependent on the grid size in the spatial domain and the total number of time steps used to obtain the numerical PDF. When the initial position r_0 is close to σ, then we have to use more grid points in the radial direction to have an accurate solution requiring more time steps to avoid oscillatory solutions. This explains why the numerical method is

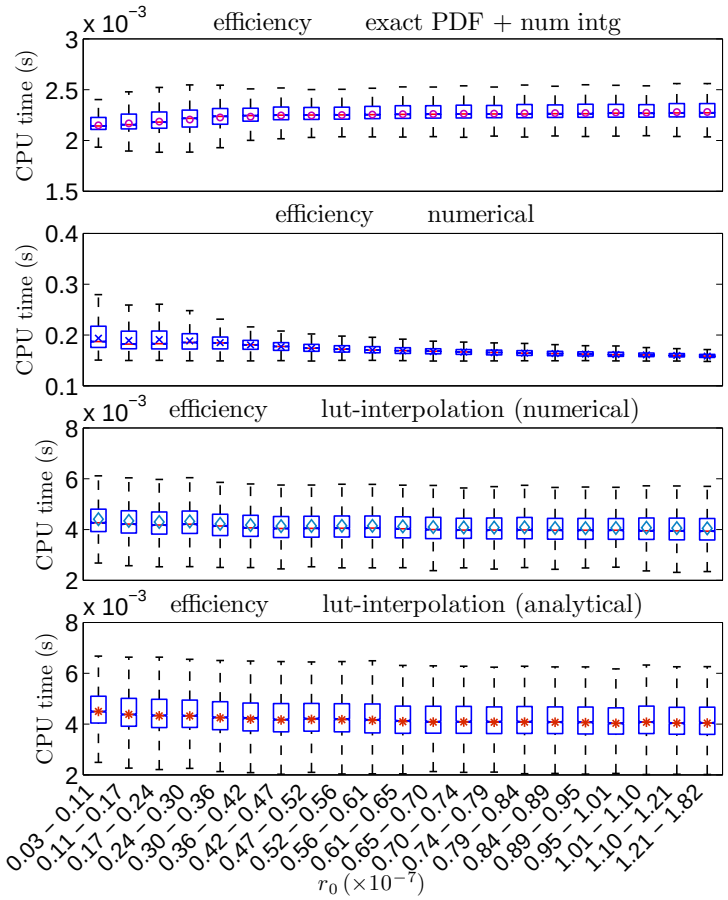

Figure 4.2: Box-plots of the CPU time required to compute r_ξ in spherical coordinates using different sampling methods versus r_0. Here $\xi = 0.5469$ and the experiment is performed for 91735 distinct pairs of values of r_0 and Δt. For every pair, the computations are repeated up to 10 times.

marginally slower in the very first sub-ranges of values for r_0. Adaptively refined grids and time steps would resolve this issue in an efficient manner.

The CDF in the table is computed in two different ways: by numerical integration of the PDF or by the analytical formula (3.10). The PDF is obtained by numerical solution of the Smoluchowski equation (3.1) or by the analytical formula (3.6). There are 200 pairs of values of r_0 in $[3 \cdot 10^{-9}, 2.06 \cdot 10^{-7}]$ and of Δt in $[1 \cdot 10^{-8}, 7.6 \cdot 10^{-4}]$ in the table. For each $(r_0, \Delta t)$ entry, there are several r values and corresponding CDF values. The size of the table is 52.7 MB. The conclusion from Table 4.1 is that the most efficient way to create the table is to use the analytical expression for the PDF and then integrate it numerically.

Method	CPU time (min)
PDF by numerical solution	15.34
Analytical PDF	1.45
Numerical integration of PDF for CDF	0.03
Analytical CDF	1.90

Table 4.1: The CPU time for different parts of the generation of a table.

4.1.2 The angular direction

Two approaches are proposed in Section 3.1.2 to draw random polar positions determined by the solution of the angular part of the Smoluchowski equation in (3.4) and (3.5): a method with an analytically computed PDF and then numerically integrated and interpolation between the CDF values precomputed and stored in a lookup table.

Accuracy comparison

The accuracy of the lut-interpolation method for θ is estimated by comparisons with the solutions obtained by the analytical method. Here we fix ξ and calculate 92225 distinct values for the parameter $\hat{t} = D\Delta t/r^2$ using all the pairs of values of Δt and r in the set `PoVsph`. Then for every value of $\hat{t}$, we compute θ_ξ using both sampling techniques and the relative error in the result obtained from the lut-interpolation method.

The error data are partitioned into 21 sub-ranges based on $\ln \hat{t}$ in Figure 4.3. In the box-plot, it is obvious that the accuracy of the lut-interpolation method is very high with an average relative error of 0.008%. An example of the dependency of the accuracy on the resolution of the precomputed lookup table is observed for small $\hat{t}$-values. For those values, the slope of the p_θ at $\theta = 0$ is large and to obtain a smooth, high-resolution CDF, a finer grid in the polar direction close to $\theta = 0$ is needed.

Efficiency comparison

The efficiency of the analytical method is compared with the lut-interpolation method by measuring the CPU time to compute the solutions in the above accuracy analysis in Figure 4.4. The accuracy of the two methods is comparable in Figure 4.4. The methods are also compared to the sampling method in [3]. There a random walk on a sphere $\mathcal{S}^2$ in $\mathbb{R}^3$ is generated directly by summation of the series for random walk on the sphere $\mathcal{S}^3$ embedded in $\mathbb{R}^4$ and projection to the sphere $\mathcal{S}^2$. The method in [3] determines a θ sample directly with a good accuracy for small $\hat{t}$ while only a few terms are needed in the series in (3.12) for large $\hat{t}$ and there its accuracy is good. The average runtime of the lut-interpolation method is 0.58 ms and for the projection method 0.35 ms. It is clear that the

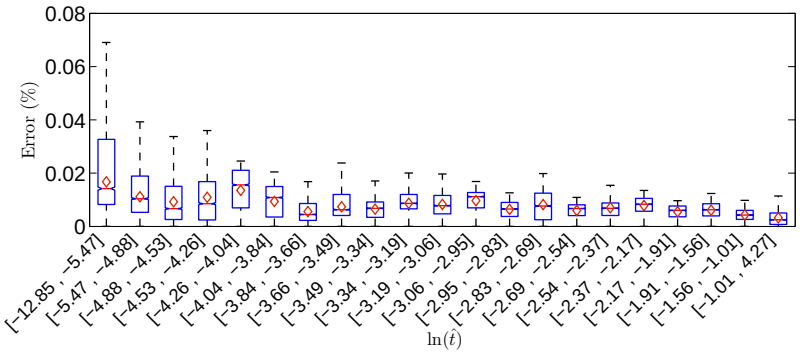

Figure 4.3: Box-plot of percentage error in computing θ_ξ using lut-interpolation methods versus $\ln \hat{t}$. Here $\xi = 0.8147$ and the experiment is performed for 92225 distinct values of $\hat{t}$.

lut-interpolation method is much more efficient than the analytical method but that the algorithm in [3] is the fastest when it works for small $\hat{t}$.

In Section 3.1.2, we observed that for small values of the parameter $\hat{t}$, evaluating p_θ via its analytical formula (3.12) can be computationally very challenging since many terms of the infinite series have to be summed to reach convergence. The problem becomes less severe for larger values of $\hat{t}$ explaining the gradual decrease in the mean CPU time for the analytical method as $\hat{t}$ increases in Figure 4.4.

The CDF in the table for interpolation in Figure 4.4 is computed at 300 $\hat{t}$ values where $\ln \hat{t}$ is between -12.85 and 5.23. The CDF is evaluated for at most 500 θ values for each $\hat{t}$. The total number of CDF data values in the table is 76862 requiring 483 KB of memory. The elapsed CPU time for its calculation is 16.7 s.

4.2 Cylindrical coordinates

The sampling methods in Section 3.2.1 for cylindrical coordinates are evaluated at 10000 randomly generated distinct pairs of values for the initial radial position r_0 and time step Δt in the set `PoVcyl`. The parameters in the model in (2.9) and (2.10) are $k_a = 10^{-11}$, $D = 10^{-12}$, and $\sigma = 2 \cdot 10^{-9}$.

4.2.1 The radial direction

The analytical method to determine the CDF based on (3.25) and the numerical integration of the PDF in (3.19) in Section 3.2.1 are compared to the lut-interpolation method with precomputed tables of the CDF.

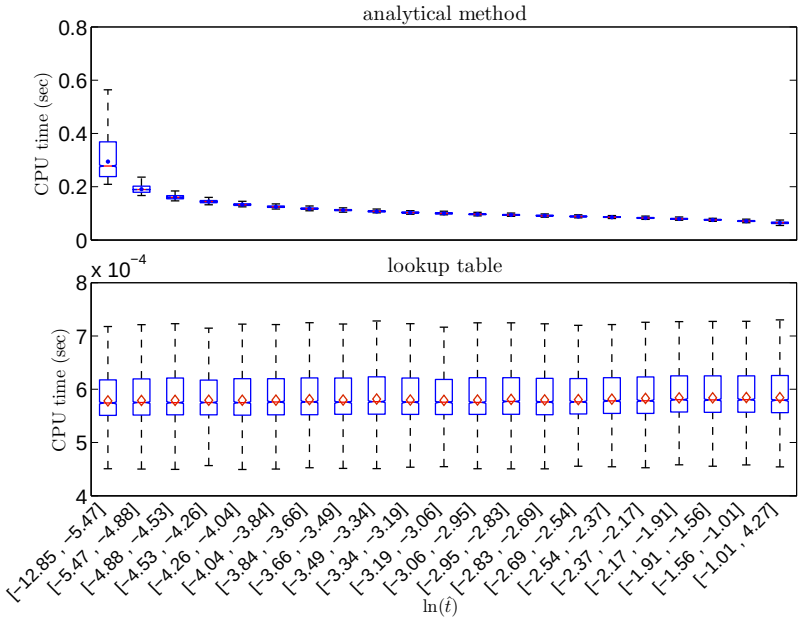

Figure 4.4: Box-plots of the CPU time required to compute θ_ξ using the analytical method and the lut-interpolation method versus $\ln(\hat{t})$. Here $\xi = 0.8147$ and the experiment is performed for 92225 distinct values of $\hat{t}$. For every pair computations are repeated up to 10 times.

Accuracy comparison

The relative errors in Figure 4.5 are determined by comparing the semi-analytical method to the other methods. The data in `PoVcyl` are grouped according to the value of r_0 and averaged over all Δt values. The mean errors in per cent are very low in most cases. For the smallest r_0-values, the accuracy in the lut-interpolation is improved by refining the table there similarly to what was necessary for the spherical coordinates.

Efficiency comparison

Using the r_0 and Δt data in `PoVcyl`, the efficiency of the sampling methods are compared in Figure 4.6. The average CPU time for the interpolation alternatives is 3.8 ms and 4.7 ms. The method based on numerical solution of the PDE uses 80 ms and the semi-analytical method is the worst with 868 ms. The performance of the semi-analytical method deteriorates for large values of r_0. The reason is that for these values the integrand in (3.25) oscillates with high frequencies making an accurate integration more difficult.

The interpolation table is computed with three methods using the semi-analytical CDF, the analytical PDF and numerical integration, and the numerical PDF integrated numerically. The number of CDF data values are between $1.0{\cdot}10^7$

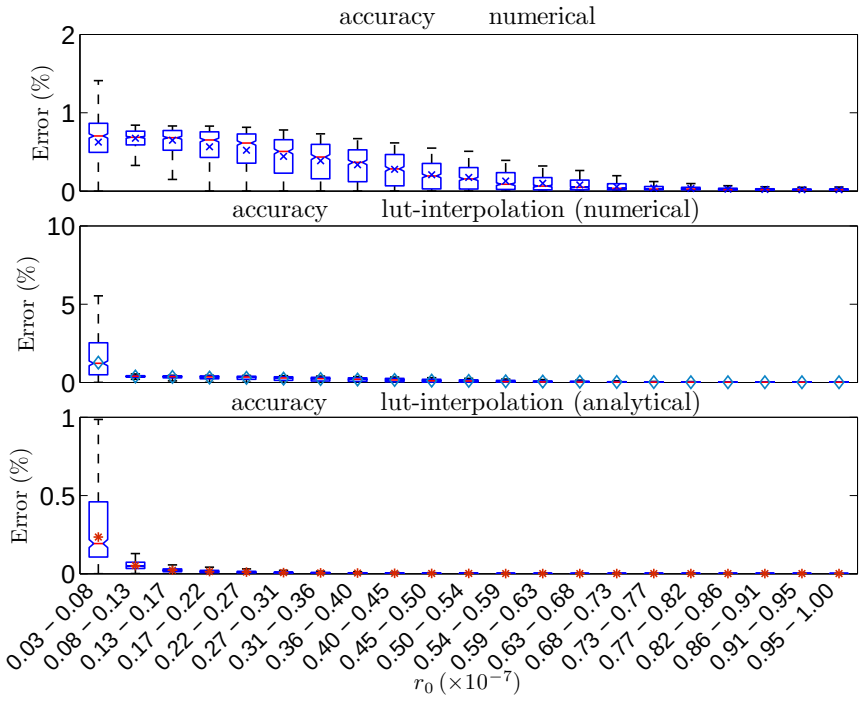

Figure 4.5: Box-plots of per cent error in computing r_ξ in cylindrical coordinates using different sampling methods versus r_0. Here $\xi = 0.4505$ and the experiment is performed for 10000 distinct pairs of values for r_0 and T.

and $1.1 \cdot 10^7$ requiring between 71 and 73 MB of storage. The difference in memory requirements is small but full numerical integration of the CDF is more than 15 time faster than using analytical formulas in Table 4.2.

Method	**CPU time (min)**
Numerical PDF and numerical integration	28.7
Analytical PDF and numerical integration	431
Semi-analytical CDF	455.5

Table 4.2: The CPU time for different ways of generating the table.

4.3 Sampling for the next reaction event

Sometimes we want to first sample the time to the next event and then the new positions. For a pair of molecules, the time t to the next reaction satisfies $S(t|r_0) = \xi$ for a uniformly distributed $\xi \in [0, 1]$ with the survival probability S defined in (2.7). Note that $S(t \to \infty|r_0) < 1$ and there is a possibility that a reaction will never occur. In the spherical case S is known analytically [10] and

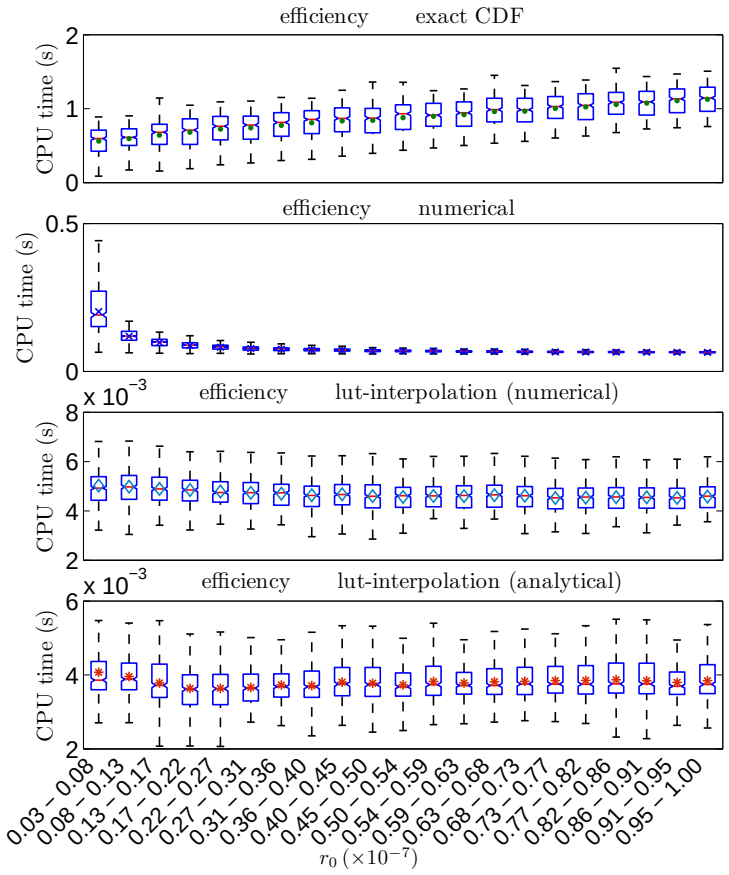

Figure 4.6: Box-plots of the CPU time required to compute r_ξ in cylindrical coordinates using different sampling methods versus r_0. Here $\xi = 0.4505$ and the experiment is performed for 10000 distinct pairs of values of r_0 and Δt. For every pair computations are repeated up to 10 times.

can be inverted numerically in the same manner as the analytical expression for the CDF. The cost of inverting S is of the same order as inverting the CDF. When computing the PDF numerically, we can compute S after each time step $\Delta\tau$ and determine if the molecules have reacted or not. If two molecules have reacted at a certain time, we can stop the computations and will save some time compared to solving the equation until the final time Δt. However, the finite difference approach will still very likely be outperformed by the analytical approach. For the analytical approach in the cylindrical case, we can compute S by numerically integrating the expression for the PDF (3.19). This will be a fairly expensive way of inverting S, and the conclusion that the numerical approach is more efficient will still hold.

4.4 Estimates of total elapsed time

Suppose that the number of time steps is N in a simulation, that there are m_i pairs of particles in step i, that the CPU time for sampling for each pair in every step is t_{sampl}, and that the elapsed time between the events is Δt_i. Ignoring single particles for which sampling is inexpensive, the total CPU time for sampling T_{sampl} and the final time T are then

$$T_{\text{sampl}} = \sum_{i=1}^{N} m_i t_{\text{sampl}} = N\overline{m}t_{\text{sampl}}, \quad T = \sum_{i=1}^{N} \Delta t_i = N\overline{\Delta t},$$

where $\overline{m}$ and $\overline{\Delta t}$ are the average number of pairs in every step and the average time step. Suppose that the CPU time to create one table is T_{tab} and that there are R reactions. Hence, the total CPU time for generating the tables and simulating the system to T is

$$RT_{\text{tab}} + t_{\text{sampl}}\overline{m}T/\overline{\Delta t}. \tag{4.1}$$

Let ρ be the density of particles and v the volume around a single particle or a pair. The radius of v is proportional to $\sqrt{2D\overline{\Delta t}}$. Hence, $1/\rho \sim v \sim \overline{\Delta t}^{3/2}$ and we have $\overline{m} \sim \rho$. The conclusion is that $\overline{m}T/\overline{\Delta t} \sim \rho^{5/3}T$. If T is large then the cost of computing the table is discounted but if the density of the particles is low then the first term in (4.1) may dominate. As an example, take the polar direction in Section 4.1.2. There $T_{\text{tab}} = 16.7$ s, $t_{\text{sampl}} = 6 \cdot 10^{-4}$ s and the break even point when the analytical method without a table is as efficient is when $\overline{m}T/\overline{\Delta t} \approx 90$, i.e. if $\overline{m} = 1$ then 90 time steps suffice. For larger values of $\overline{m}T/\overline{\Delta t}$ the table will win.

5 Simulation of a pair of molecules

Based on the accuracy and efficiency analysis in the previous section, the best technique is chosen to generate particle positions for simulation of reaction-diffusion processes in the spherical and the cylindrical coordinate systems. The algorithm from [9] can be outlined as follows:

1. Given r_0 and Δt, sample for the new radial position r.

2. Given r and Δt, sample for the new polar angle θ or the new azimuthal angle ϕ in the spherical and cylindrical cases, respectively.

3. Given r and Δt, sample for the azimuthal angle ϕ or the axial position z in the spherical and cylindrical cases, respectively.

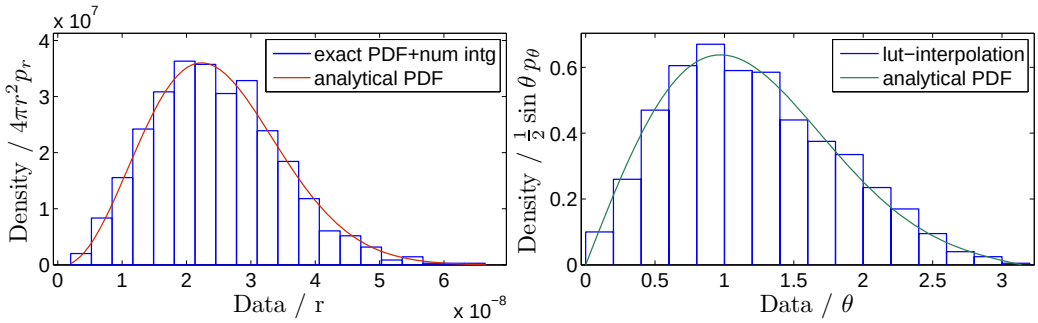

Figure 5.1: The probability densities $4\pi r^2 p_r$ in the radial direction with $\Delta t = 5 \cdot 10^{-5}, r_0 = 10^{-8}$ and $T = 5 \cdot 10^{-3}$ (left) and $\frac{1}{2}\sin\theta p_\theta$ in the polar direction with $\Delta t = 7 \cdot 10^{-4}, r_0 = 5 \cdot 10^{-8}$ and $T = 7 \cdot 10^{-2}$ (right) obtained by sampling are compared to the analytical density at T.

5.1 Spherical coordinates

The optimal method according to Sections 4.1.1 and 4.1.2 in the spherical case is sampling radial positions by numerical integration of the analytical PDF and polar positions using interpolation from precomputed tabulated CDF data. The species A and B react with each other and form C as in (2.1) with A fixed at the origin with a diffusion coefficient $D = 2 \cdot 10^{-12}$ and reaction radius $\sigma = 2 \cdot 10^{-9}$. We consider two cases: a reversible reaction with $k_a = 10^{-19}$, $k_d = 2.9894$ and *free diffusion* with $k_a = k_d = 0$.

The new positions of 1000 particles are sampled in the radial and the polar directions and collected in a histogram in Figure 5.1 for the reversible reaction. The weighted p_r and p_θ are compared to the analytical solutions with good agreement.

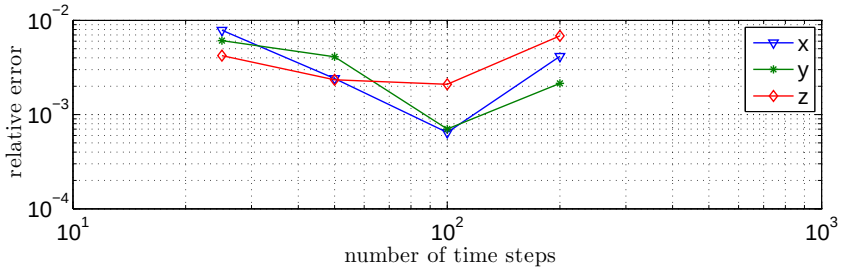

Figure 5.2: The relative difference between the standard deviation of the simulated position and the analytical Gaussian standard deviation for the position of freely diffusing molecules at $T = 5 \cdot 10^{-5}$ with $D = 2 \cdot 10^{-12}$ using different Δt.

When no reaction takes place, the new positions in Cartesian coordinates are determined using the above operator splitting algorithm starting at a point away from the origin. One molecule is simulated in a time interval and $2 \cdot 10^5$

trajectories are generated for each fixed splitting time step Δt. In the comparison of the computed and analytical standard deviations in Figure 5.2, there is no clear tendency when Δt is varied and other errors such as the statistical variation appear to be much larger than the variation due to Δt. The error is already quite low but to reduce the statistical error further by a factor 10 we would need $2 \cdot 10^7$ realizations of the process.

5.2 Cylindrical coordinates

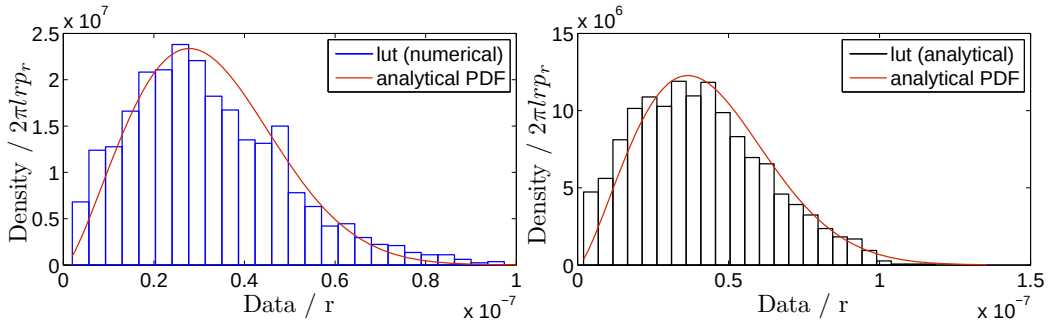

Figure 5.3: The probability densities $2\pi r \ell p_r$ with $\ell = 1$ in the radial direction with $\Delta t = 2 \cdot 10^{-4}, r_0 = 2 \cdot 10^{-8}$ and $T = 1.4 \cdot 10^{-2}$ computed numerically (left) and analytically with $\Delta t = 5 \cdot 10^{-4}, r_0 = 9 \cdot 10^{-9}$ and $T = 10^{-2}$ (right) are compared to the analytical solution at T.

The stationary A molecule is a polymer along the z axis in a cylindrical coordinate system in this example and the B molecule is moving in 3D space and can react with the polymer by association. The parameters are $k_a = 10^{-11}$, $D = 10^{-12}$, and $\sigma = 2 \cdot 10^{-9}$. The distribution of the radial position is found at the final time T using tabulated data in Figure 5.3 using the statistics from 2000 particles. The accuracy is good for large r_0, but for r_0 close to σ the accuracy is worse. To obtain good accuracy in this case we may need a very high resolution in the table.

6 Summary and conclusions

Event-driven algorithms such as the GFRD algorithm [19] for simulation of biochemical reaction-diffusion processes need fast evaluations of CDFs in the sampling for the next events or the next positions of the particles in the system. The relative positions of two particles in a pair are governed by a PDF satisfying the Smoluchowski equation. We have compared different methods for the sampling when it is split into a radial part and an angular part in a spherical and a cylin-

drical coordinate system as in [9]. Although we investigate a particular model, the conclusions are of general interest for event-driven algorithms.

The Smoluchowski equation in the radial direction in a spherical coordinate system is solved by an analytical method and the Crank-Nicolson finite difference scheme. The sampling is performed using the analytical CDF of the distribution, the CDF calculated by numerical integration of the analytical PDF, and linear interpolation between the entries of a 3D lookup table with analytically or numerically precomputed CDFs for a set of model parameter values. The lowest CPU time to determine a random radial position is obtained by numerical integration of the PDF. Presumably, the results obtained from this method have the highest accuracy in spite of the fact that the accuracy is influenced by errors introduced by truncating infinite sums and numerical approximations in the integration. Almost as efficient are the methods based on interpolation in a table. A different implementation of the methods might change the recommendation in this case. The explanation why the tables are not the best alternative is that the formulas to compute the PDF and the CDF are relatively simple.

In the polar direction in a spherical coordinate system, the CDF is evaluated partly analytically and partly numerically and there are two approaches to sample from the CDF: using the computed CDF directly and tabulating pre-evaluated CDFs in a 2D lookup table first and then performing linear interpolation between the data points in the table. Using the lookup table is here orders of magnitude faster than the partially analytical method. It is unlikely that this difference is due to the implementation of the methods. The accuracy of the interpolation in the table is very good, but it depends on the resolution in the table. The CDF is here more complicated to compute than in the previous case and tables are the preferred choice. The generation of the table is inexpensive in this case.

The radial direction in a cylindrical coordinate system is sampled with a semi-analytically defined CDF, one computed numerically, and the CDF stored in 3D tables for inverse interpolation. The tables provide the new r value much faster than the other two methods. It is improbable that this conclusion will be changed by different implementations of the algorithms. The explanation to this result compared to the radial direction in the spherical system is that in the cylinder case the analytical CDF is rather complicated to evaluate with a highly oscillatory integrand while the closed formula in the spherical case has only a few terms. Computing the table entries numerically is here the cheapest method.

In general, using lookup tables is recommended but it depends on the application and the available analytical formulas, the density of the particles, and the amount of time that the simulation will run, as the method has an initial cost in time for creating the tables and a cost in memory space for data storage. One table is needed for each reaction in the biochemical system. Since tables are finite data sets, limitations are imposed on the longest distance that two molecules can move apart and the longest time step that can be taken during the simulation. Furthermore, there is a shortest distance between the initial r_0 to the reaction

radius σ that a table can resolve with sufficient accuracy. Refinements of the tables are necessary for certain combinations of parameter values. The cost of such refinements are higher the higher the dimension of the table is. This is another argument for splitting the spatial sampling since the dimensions of the tables are usually lower if the sampling in space is made in a sequence of steps.

References

[1] S. S. Andrews, N. J. Addy, R. Brent, and A. P. Arkin. Detailed simulations of cell biology with Smoldyn 2.1. *PLoS Comput. Biol.*, 6(3):e1000705, 2010.

[2] M. H. Bani-Hashemian. Accurate and Efficient Solution of the Smoluchowski Equation. Master's thesis, Department of Information Technology, Uppsala University, Uppsala, Sweden, 2011. *Available online at:* `http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-156440` .

[3] T. Carlsson, T. Ekholm, and C. Elvingson. Algorithm for generating a Brownian motion on a sphere. *J. Phys. A: Math. Theor.*, 43:505001, 2010.

[4] H. S. Carslaw and J. C. Jaeger. *Conduction of Heat in Solids*. Oxford University Press, Oxford, second edition, 1959.

[5] J. S. Coggan, T. M. Bartol, E. Esquenazi, J. R. Stiles, S. Lamont, M. E. Martone, D. K. Berg, M. H. Ellisman, and T. J. Sejnowski. Evidence for ectopic neurotransmission at a neuronal synapse. *Science*, 309:446–451, 2005.

[6] F. C. Collins and G. E. Kimball. Diffusion-controlled reaction rates. *J. Colloid. Sci.*, 4:425–437, 1949.

[7] A. Donev. Asynchronous event-driven particle algorithms. *Simulation*, 85(4):229–242, 2009.

[8] A. Donev, V. V. Bulatov, T. Oppelstrup, G. H. Gilmer, B. Sadigh, and M. H. Kalos. A first-passage kinetic Monte Carlo algorithm for complex diffusion-reaction systems. *J. Comput. Phys.*, 229:3214–3236, 2010.

[9] S. Hellander and P. Lötstedt. Flexible single molecule simulation of reaction-diffusion processes. *J. Comput. Phys.*, 230(10):3948–3965, May 2011.

[10] H. Kim and K. J. Shin. Exact solution of the reversible diffusion-influenced reaction for an isolated pair in three dimensions. *Phys. Rev. Lett.*, 82(7):1578–1581, 1999.

[11] H. Kim, K. J. Shin, and M. Yang. Dynamic correlation effect in reversible diffusion-influenced reactions: Brownian dynamics simulation in three dimensions. *J. Chem. Phys.*, 111(3):1068–1075, 1999.

[12] S. J Plimpton and A. Slepoy. Microbial cell modeling via reacting diffusive particles. *J. Physics: Conf. Series*, 16(1):305, 2005.

[13] A. Scala, Th. Voigtmann, and C. De Michele. Event-driven Brownian dynamics for hard spheres. *J. Chem. Phys.*, 126(13):134109, 2007.

[14] M. Smoluchowski. Versuch einer mathematischen Theorie der Koagulationskinetik kolloider Lösungen. *Z. Phys. Chem.*, 92:129–168, 1917.

[15] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968.

[16] K. Takahashi, S. Tănase-Nicola, and P. R. ten Wolde. Spatio-temporal correlations can dratically change the response of a MAPK pathway. *Proc. Natl. Acad. Sci. USA.*, 107(6):2473–2478, 2010.

[17] J. Dalla Torre, J.-L. Bocquet, N. V. Doan, E. Adam, and A. Barbu. JERK, an event-based Kinetic Monte Carlo model to predict microstructure evolution of materials under irradiation. *Phil. Mag.*, 85(4):549–558, 2005.

[18] V. Tulovsky and L. Papiez. Formula for the fundamental solution of the heat equation on the sphere. *Appl. Math. Lett.*, 14:881–884, 2001.

[19] J. S. van Zon and P. R. ten Wolde. Green's-Function Reaction Dynamics: A particle-based approach for simulating biochemical networks in time and space. *J. Chem. Phys.*, 123(23):234910, 2005.

[20] S. Winitzki. Uniform approximations for transcendental functions. In V. Kumar, M. Gavrilova, C. Tan, and P. L'Ecuyer, editors, *Computational Science and Its Applications - ICCSA 2003*, volume 2667 of *Lecture Notes in Computer Science*, pages 962–962. Springer, 2003.

[21] K. Yosida. Brownian motion on the surface of the 3-sphere. *Ann. Math. Statistics*, 20:292–296, 1949.