



The importance of introducing a waiting time for Lattice Monte Carlo simulations of a polymer translocation process

Hendrick W. de Haan^{a,*}, Michel G. Gauthier^{a,b}, Mykyta V. Chubynsky^a, Gary W. Slater^a

^a Department of Physics, University of Ottawa, 150 Louis-Pasteur, Ottawa, Ontario, Canada, K1N 6N5

^b Department of Physics, Simon Fraser University, 8888 University Dr., Burnaby, British Columbia, Canada, V5A 1S6

ARTICLE INFO

Article history:

Received 1 March 2010

Received in revised form 19 July 2010

Accepted 31 July 2010

Available online 7 August 2010

Keywords:

Monte Carlo
Simulations
Translocation
Polymer
Nanopore

ABSTRACT

For a standard Lattice Monte Carlo (LMC) simulation of a random walker subject to a bias, it is impossible to obtain both a correct mean velocity and diffusion coefficient. To correct this, a modified LMC algorithm has been developed where the introduction of a probability of remaining in the current state allows for a distribution of intervals between jumps. In this paper, we demonstrate the impact of this modification for a first-passage problem: the translocation of a polymer through a nanopore. We find that while either approach yields the correct mean first-passage time, the incorporation of a waiting time is necessary to obtain the correct spread of times.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

In many physical processes, the dynamics of a particle in a fluid can be modeled as a (possibly biased) random walk. If we consider the simplest case of a Brownian particle moving in one dimension in the absence of an external bias, it is obvious that the particle must exhibit a zero mean velocity $v = 0$. However, due to thermal fluctuations, there is a non-zero diffusion coefficient

$$D = k_B T / \xi, \quad (1)$$

where $k_B T$ is the usual thermal factor and ξ is the particle's friction coefficient. An external force F leads to the mean velocity

$$v = F / \xi = (D / k_B T) F, \quad (2)$$

but the diffusion coefficient D must remain unchanged (i.e., the dynamics of the particle must obey Eqs. (1) and (2) for any F [1]).

Such systems are often simulated using *Lattice Monte Carlo* (LMC) algorithms. In a standard approach, the evolution of the system is governed by the probabilities of a random walker jumping forwards or backwards (p_{\pm}) on a lattice of spacing a with a time step Δt . In the absence of a bias, choosing $p_{\pm} = \frac{1}{2}$ and

$\Delta t = t_B \equiv a^2 / 2D$ yields the correct $v = 0$ and D . However, there are still some problems with this algorithm, as we mention below.

When $F \neq 0$, in order to satisfy the detailed balance condition we need $p_{\pm}(\epsilon) = 1 / (1 + e^{\mp 2\epsilon})$, where $\epsilon = Fa / 2k_B T$ is the dimensionless force [2]. In order to satisfy Eq. (2), the time step needs to be modified such that jumps under a higher bias take a shorter time: $\Delta t(\epsilon) = (\tanh \epsilon / \epsilon) t_B$ [2]. However, it is easy to prove that the correct $D = a^2 / 2t_B$ is only recovered in the limit $\epsilon \rightarrow 0$. For $\epsilon \neq 0$, D decreases monotonically and, in fact, this decrease is exponential when $\epsilon \gg 1$: $D \sim e^{-2\epsilon}$. Thus, within the framework of the standard LMC approach, it is impossible to obtain correct values for D and $v(\epsilon)$ and satisfy detailed balance if $\epsilon \neq 0$.

To correct for this, the key is to allow for a distribution of the intervals between (successful) jumps [2]. We have developed a modified LMC algorithm where this is achieved by introducing a probability $s(\epsilon)$ of the walker not jumping at a given step of the algorithm. With the addition of this *waiting time*, it is then possible to achieve both a correct v and D . As an example of the necessity of this modification, we apply both the standard and modified LMC algorithms to a first-passage time problem: the translocation of a polymer through a nanopore. Using a simplified translocation model, we demonstrate that while both algorithms yield the same mean translocation times τ , only the modified algorithm yields the correct variance of τ . In particular, the difference between the spread of the passage times is found to be substantial in two regimes: i) very short polymers and ii) moderate to high external fields.

* Corresponding author.

E-mail addresses: hendrick.dehaan@gmail.com (H.W. de Haan), gary.slater@uottawa.ca (G.W. Slater).

2. Simulation setup

2.1. Modified MC approach

As shown in Ref. [2], the correct velocity v and diffusion coefficient D are obtained if the probability of the system remaining in its current state is given by

$$s(\epsilon) = \epsilon^{-1} \coth \epsilon - \text{csch}^2 \epsilon \quad (3)$$

and the jumping probabilities are

$$p_{\pm}(\epsilon) = (1 - s(\epsilon)) / (1 + e^{\mp 2\epsilon}), \quad (4)$$

with the corresponding time step given by

$$\Delta t(\epsilon) = (1 - s(\epsilon)) (\tanh \epsilon / \epsilon) t_B. \quad (5)$$

In fact, an algorithm with any $s(\epsilon)$ between 0 and 1 and with the rest of the parameters given by Eqs. (4) and (5) gives the correct v and satisfies detailed balance; in particular, $s(\epsilon) = 0$ corresponds to the standard algorithm without the waiting time. However, using $s(\epsilon)$ from Eq. (3) is the only way to get the correct D as well.

Another advantage of the new algorithm is that it gives a more accurate solution of the *first-passage problem*. In its simplest formulation, one considers a particle that starts at point x at time $t = 0$ and moves between two walls, at $x - b_1$ and $x + b_2$ ($b_1, b_2 > 0$), and finds the first time τ it reaches one of the walls [the *first-passage time* (FPT)]. The FPT distribution provides the full solution of the problem, but often the first two moments, the mean FPT (MFPT) and the mean-square FPT (MSFPT), are sufficient. One can likewise formulate the first-passage problem for LMC, by considering a particle that starts at site i and finding the time at which it first reaches one of the sites $i - j_1$ and $i + j_2$ ($j_1, j_2 > 0$). If $j_1 = j_2 = 1$ (i.e., the FPT to the neighbouring sites is considered), then this FPT coincides with the time until the next jump (as any jump brings the particle to one of the neighbouring sites), which always has the same distribution as the interval between jumps. It was shown in Ref. [2] that for any $s(\epsilon)$, whenever Eqs. (4) and (5) are satisfied, the mean interval between jumps coincides with the MFPT for the continuum problem when the distances between the initial position and the walls in the latter problem, b_1 and b_2 , are both equal to a (the distance between neighbouring sites in the LMC problem). Moreover, the mean-square interval between jumps can be made to coincide with the MSFPT for the same continuum problem, but this only happens when $s(\epsilon)$ is given by Eq. (3), i.e., for the new algorithm, but not for the standard one without waiting time. That is, the new algorithm gives the correct variance of the FPT, but the standard one does not. In fact, without the waiting time the move to a neighbouring site always occurs at the first step, so the FPT is deterministic and its variance is zero.

Based on the above consideration, it is easy to see that the first two moments of the FPT will be correct in the new algorithm when the walls are located at any of the lattice sites, i.e., for any values of j_1 and j_2 (with the corresponding distances in the continuum problem being $b_1 = j_1 a$ and $b_2 = j_2 a$). Indeed, the detailed balance guarantees that the ratio of the probabilities of the jumps to the left and to the right is correct (i.e., is the same as the ratio of the probabilities to first move to the left by a and to first move to the right by a in the continuum diffusion problem). Therefore the probability of any particular sequence of jumps is correct and so the distribution of the number of jumps needed to reach the walls is correct as well. Then the correct mean and variance of each interval between jumps guarantees that the mean and variance of the duration of the whole process are correct as well.

Interestingly, in the limit $\epsilon \rightarrow 0$, Eqs. (3)–(5) give $s = 2/3$, $p_{\pm} = 1/6$, $\Delta t = t_B/3$. This differs from the standard unbiased algorithm with $s = 0$, $p_{\pm} = 1/2$ and $\Delta t = t_B$. Although both give the

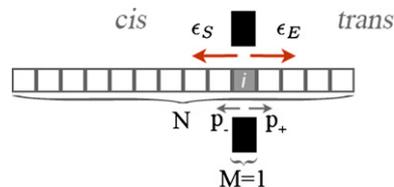


Fig. 1. Schematic of the 1D translocation model for a polymer of length N in a pore of length M (here, $N = 14$ and $M = 1$). The state of the system is given by the translocation coordinate i ($i = 5$ here).

correct D and thus the second moment of the particle distribution, only the new algorithm also gives the correct second moment of the FPT distribution, as well as the fourth moment of the particle distribution [3].

2.2. Translocation model

The process of passage of a polymer (possibly driven by an external force) through a nanopore in a thin membrane (*polymer translocation*) is interesting both theoretically and in view of its potential applications, e.g., to DNA sequencing [4]. A very active area of research, there have been theoretical and simulation studies examining the translocation process (e.g. [5–12]). In this paper we concentrate on one experimentally measurable quantity, the translocation time, i.e. the interval between the moment the first monomer enters the nanopore and the moment the last monomer leaves it.

The underlying assumption of the simplified translocation model used here [14,15] is that the coils the polymer forms on both sides of the wall remain close to equilibrium. In that case, the state of the polymer can be approximately reduced to a single number, the translocation coordinate, which is the length of the polymer that has already translocated [5,6]. In the discretized version, the polymer is modeled as a single line of N sites (Fig. 1) and the translocation coordinate i is the number of the monomer (site) within the pore (the pore length is fixed at $M = 1$ in this work). This reduces the translocation process to a 1D biased random walk. In addition to the external force ϵ_E acting on the walker, there is also an entropic force ϵ_S associated with the change of the entropy of the polymer as it passes through the nanopore. All calculations are performed starting from $i = 1$ (the first monomer is in the pore and all others on the side we label *cis*; the other side is labeled *trans*). The translocation time τ is the time it takes to reach $i = N + 1$, provided that this happens before $i = 0$, which would correspond to the polymer escaping back to the *cis* side. Note that while this quantity is not identical to the FPT as defined above, the two problems are clearly related; the same considerations can be applied to the translocation time, with the conclusion that the new algorithm with the waiting time should provide both the correct mean and the correct variance, whereas when using the algorithm without the waiting time only the mean should be correct. When performing the simulations, we must calculate the net bias ϵ on the polymer in any state i such that we can propagate the system according to Eqs. (3)–(5).

An estimate of the entropic force ϵ_S was developed in [14] and the form for the special case of $M = 1$ is given by

$$\epsilon(s) = \begin{cases} \frac{1-\gamma}{2(N-1)} - \frac{\ln(\bar{z})}{2}, & i = 1, \\ \frac{1-\gamma}{2(N-i)} - \frac{1-\gamma}{2(i-1)}, & 1 < i < N, \\ \frac{\ln(\bar{z})}{2} - \frac{1-\gamma}{2(N-1)}, & i = N. \end{cases} \quad (6)$$

This form is valid for a self-avoiding walk on a 3D lattice with \bar{z} being the effective lattice coordination number. For this work, \bar{z} is set to 3 as for a tetrahedral network. However, the choice of a

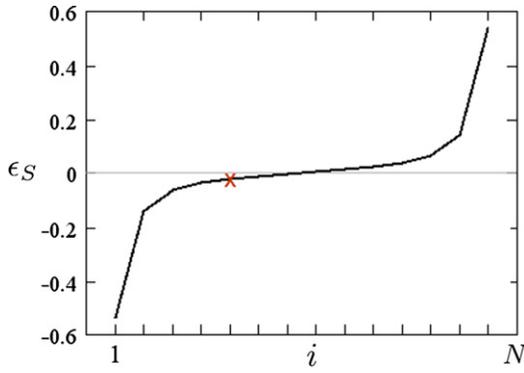


Fig. 2. Entropic force ϵ_S as a function of the translocation coordinate i for $N = 14$ and $M = 1$. The force at $i = 5$ corresponding to Fig. 1 is indicated by an x .

different value corresponding to a different lattice would have no significant impact on the results as \bar{z} appears only at the very beginning or very end of the translocation process. Known as the *surface exponent*, γ corresponds to the entropic penalty arising from the reduced number of states due to the presence of the wall and depends only on the polymer model: $\gamma = 0.50$ for an ideal chain, $\gamma = 0.69$ for a self-avoiding polymer (used here), and $\gamma = 1.00$ for a rod-like chain [13]. A plot of the entropic force as a function of the translocation coordinate i is shown in Fig. 2.

Note that the entropic force favours the side on which there is a greater number of monomers but that the magnitude is small unless almost all of the chain is on the same side. The external force ϵ_E models the application of an external electric field \mathcal{E} to the system. For a nanopore-containing membrane with large *cis* and *trans* reservoirs, the potential drop is mostly across the nanopore. Hence, the resulting force is felt only by monomers within the pore. In the simplified model, this means that for $M = 1$, the dimensionless external force is given by $\epsilon_E = q\mathcal{E}a/2k_B T$ for all i , where q is the charge of a monomer and a is its size. When discussing our results, we use the dimensionless form of the field, $E = q\mathcal{E}a/k_B T$. See Ref. [14] for details.

2.3. Simulation technique

We can use Eqs. (3)–(5) with the total force $\epsilon(i) = \epsilon_S(i) + \epsilon_E$ to perform LMC simulations. After simulating enough successful events, we can construct a distribution of translocation times, determine the mean translocation time $\mu = \langle \tau \rangle$ and the variance $\sigma^2 = \langle \tau^2 \rangle - \langle \tau \rangle^2$. However, given our initial condition of $i = 1$, there are many events in which the polymer ends up on the *cis* side (failed event). In fact, at low fields, it is prohibitively rare for a polymer of any significant length to successfully translocate and direct simulation will not yield satisfactory statistics. Instead, we have developed an exact methodology for solving such systems described by a linked set of master equations [14]. This approach allows us to determine the exact values of all the moments, including $\langle \tau \rangle$ and $\langle \tau^2 \rangle$, at any polymer length and external field. While we do not obtain the distributions themselves, we are able to compute the variance of the translocation times. Unlike in direct LMC simulations, all calculations are deterministic (no random numbers are involved), there is no statistical noise, so the accuracy of the final result is determined entirely by numerical round-off errors, and in this sense, the approach is *numerically exact*. In this work, we employ both approaches. The exact calculation is used to generate a comparison between the modified and standard algorithms over a wide range of N and E values while direct LMC simulations produce the distributions for a selected case.

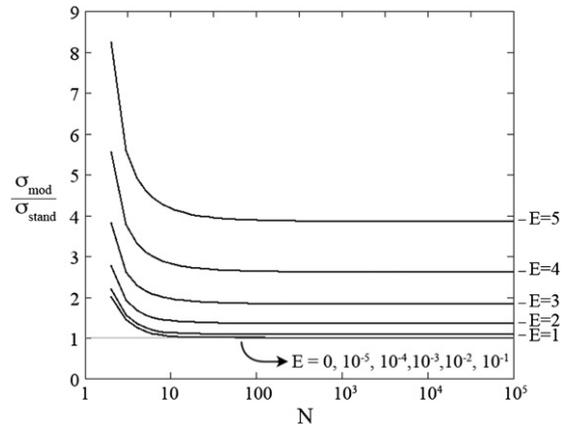


Fig. 3. Standard deviation of translocation times for the modified algorithm normalized by the value obtained using the algorithm without waiting times vs. polymer length N at different external field values E .

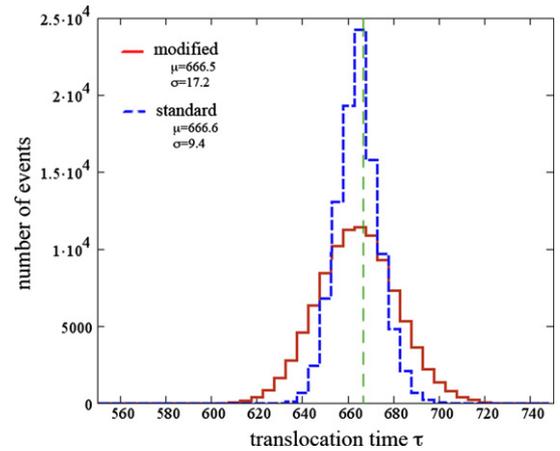


Fig. 4. Distributions of translocation times for 100 000 events generated by Monte Carlo simulations using the modified (solid red line) and standard (dashed blue line) algorithms for a polymer of length $N = 1000$ at an external field of $E = 3$. The exact result, $\langle \tau \rangle = 666.620$, is indicated by the vertical dashed green line. (For interpretation of colours in this figure, the reader is referred to the web version of this article.)

3. Results

Using the exact numerical approach, the first and second moments of the distribution of translocation times were calculated for both the standard (no waiting times) and modified (with waiting times) algorithms. These calculations were performed for a wide range of polymer lengths (ranging from $N = 1$ to $N = 10^5$) and external fields ($E = 0, 10^{-5}, \dots, 10^{-1}, 1, \dots, 5$). As expected, the mean translocation time (τ) is identical within numerical accuracy across the entire range of N and E (not shown). However, the spread of translocation times is found to be significantly different between algorithms (see Fig. 3) in two regions: short polymers and high fields.

We note that the standard algorithm properly captures the N -dependence of σ for long polymers; however, it seriously underestimates σ for fields $E \geq 1$. For short polymers, the standard algorithm gets both the field and size dependences wrong, again grossly underestimating σ . To investigate this difference, direct Monte Carlo simulations were performed to obtain 10^5 successful translocation events using both algorithms for $N = 1000$ at $E = 3$. As shown in Fig. 4, both algorithms return the same mean, but the width of the distribution is almost a factor of two larger using the modified algorithm, in agreement with the exact calculations.

To explain this discrepancy, recall that although the standard LMC method recovers the proper mean velocity, it gives a diffusion coefficient that *decreases* with increasing bias. This artefact leads to an overly deterministic result for the translocation times. Another way to see this is by recalling that the interval between jumps is deterministic without the waiting time. To correct for this, we must introduce a waiting time such that with the proper choice of $s(\epsilon)$, one is able to obtain $D = a^2/2t_B$ at all values of ϵ . Thus, as ϵ increases, the modified algorithm maintains the correct spread.

We also note that there is a discrepancy between the two algorithms even at zero field, but only for short polymers. There are two reasons for this. First, as mentioned, even though the standard algorithm gives the correct D at $\epsilon = 0$, the second moment of the FPT is incorrect; however, the relative error vanishes as $N \rightarrow \infty$ [3]. Second, due to the entropic force, a bias is present even for $E = 0$. Again, this effect vanishes rapidly as N increases. Given the form presented in Fig. 1, the bias due to the entropic force is significant only near the ends of the polymer: $|\epsilon_S| = \ln(\bar{z})/2 \approx 0.55$ for $i = 1$ or $i = N$ (when $N \gg 1$) and decays quickly when moving away from these end points. Taking $E \sim 1$ as the critical field above which the discrepancies between the two algorithms are significant for all N , the effect of a bias should become non-negligible around $\epsilon \approx 0.5$. Hence, for small E , the effect of the entropic force on the calculated distribution of translocation times will be significant only when the time spent near $i = 1$ or $i = N$ is an appreciable portion of the total translocation time, which is true only for very short polymers.

4. Conclusion

We have shown that to obtain correct results using an LMC algorithm to model a random walker, it is necessary to introduce a distribution of intervals between jumps – here achieved by a waiting time (a non-zero probability to stay put during an MC step). We have demonstrated the particular impact this modification has on a first-passage time problem by comparing the standard and modified algorithms in the context of a simple picture of translocation. While either approach yields the same mean first-passage

time, the distribution of these times is strongly affected by including a waiting time – particularly at a very short polymer length or a moderate to high bias. We note that there is no direct relation between the waiting time as considered here and rejection of moves in the Metropolis algorithm [16]: the latter does not produce correct dynamics, only correct equilibrium distributions, and this can be achieved without the waiting time. While this work by itself, strictly speaking, does not prove that the modified algorithm gives the exact dispersion of the translocation times, we expect this on theoretical grounds; this can be tested using a simple system for which the dispersion of the first-passage times is known exactly.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The computational resources of the SHARCNET and HPCVL networks were used.

References

- [1] D.A. McQuarrie, *Statistical Mechanics*, University Science Books, Sausalito, CA, 2000.
- [2] M.G. Gauthier, G.W. Slater, *Phys. Rev. E* 70 (2004) 015103(R).
- [3] M.V. Chubynsky, G.W. Slater, submitted for publication, arXiv:1008.3558.
- [4] C. Dekker, *Nature Nanotechnology* 2 (2007) 209.
- [5] W. Sung, P.J. Park, *Phys. Rev. Lett.* 77 (4) (1996) 783–786.
- [6] M. Muthukumar, *J. Chem. Phys.* 111 (22) (1999) 10371–10374.
- [7] J.K. Wolterink, G.T. Barkema, D. Panja, *Phys. Rev. Lett.* 96 (2006) 208301.
- [8] K. Luo, S.T.T. Ollila, I. Huopaniemi, T. Ala-Nissila, P. Pomorski, M. Karttunen, S.C. Ying, A. Bhattacharya, *Phys. Rev. E* 78 (2008) 050901(R).
- [9] J.P. Hernández-Ortiz, M. Chopra, S. Geier, J.J. de Pablo, *J. Chem. Phys.* 131 (4) (2009) 044904.
- [10] A. Aksimentiev, *Nanoscale* 2 (2010) 448–483.
- [11] A. Bhattacharya, K. Binder, *Phys. Rev. E* 81 (2010) 041804.
- [12] H.W. de Haan, G.W. Slater, *Phys. Rev. E* 81 (2010) 051802.
- [13] E. Eisenriegler, K. Kremer, K. Binder, *J. Chem. Phys.* 77 (1982) 6296–6320.
- [14] M.G. Gauthier, G.W. Slater, *J. Chem. Phys.* 128 (2008) 065103.
- [15] M.G. Gauthier, G.W. Slater, *J. Chem. Phys.* 128 (2008) 205103.
- [16] M.P. Allen, D.J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, Toronto, 1992.