PAPER • OPEN ACCESS

Fast simulation of a large polymer with untruncated interaction near the collapse transition

To cite this article: Stefan Schnabel and Wolfhard Janke 2022 *J. Phys.: Conf. Ser.* **2241** 012005

View the article online for updates and enhancements.

Fast simulation of a large polymer with untruncated interaction near the collapse transition

Stefan Schnabel and Wolfhard Janke

Institut für Theoretische Physik, Universität Leipzig, IPF 231101, 04081 Leipzig, Germany

E-mail: schnabel@itp.uni-leipzig.de

Abstract. We show how polymer models with untruncated intra-molecular interactions and continuous degrees of freedom can be investigated by means of Monte Carlo techniques even for large systems. To this end we adapt Clisby's method for the simulation of self-avoiding walks and introduce a novel way to implement the well-known Metropolis algorithm.

1. Introduction

In the phase diagram of flexible polymer chains the Θ–point lies in between good and bad solvent. For the good solvent, the system maximizes the polymer–solvent interface area causing the polymer to occupy extended configurations that resemble random walks which are self-avoiding due to the excluded volume interaction. The average end–to–end distance of a chain of length L, therefore, scales like $\langle r_{\mathrm{ee}}\rangle \propto L^{\nu}$ with $\nu \approx 0.588$ in three dimensions. In a bad solvent the surface area is minimized and the polymer forms denser globule-like shapes. The system is expected to grow like a droplet of constant density such that the linear extension is proportional to the third root of the droplet's volume which in turn is proportional to L. Hence, $\langle r_{\mathrm{ee}}\rangle \propto L^{\nu}$ with $\nu = 1/3$. In between at the Θ-point the polymer is supposed to behave like an ideal chain or a Gaussian random walk (GRW) which implies $\nu = 1/2$.

There have been many numerical studies of the behavior of polymers near the Θ–point [1, 2, 3, 4]. In the majority of them lattice polymers were considered since generally such models allow for the investigation of longer chains. For example in a recent publication [5] off-lattice polymers of length $L < 500$ were studied while Vogel et al. [6] simulated lattice polymers with $N = L + 1 = 32000$ monomers. Due to the underlying lattice geometry the latter are less realistic than the former and it is, therefore, desirable to have efficient algorithms for off-lattice polymers at hand. The naïve method to simulate these models is hampered by the large number of terms in the Hamiltonian and requires $\propto L^2$ operations for an individual move. This can relative easily be improved to $\propto L$, e.g. by partitioning of space, but only if the interaction potential is of limited range. Since a number of natural choices like the standard 12-6 Lennard-Jones potential are thus excluded from consideration, a method that is able to also investigate potentials that are non-zero at all distances would be useful.

The paper is organized as follows: In section 2 we briefly discuss the Lennard-Jones polymer model that we are using. Then, in section 3, we describe a parsimonious version of the Metropolis algorithm which allows to decide upon suggested configuration changes without exact knowledge of the change in energy. This is followed by a discussion of the employed data structure, i.e., our version of Clisby's binary tree, in section 4. Some data regarding the performance of our method is presented in section 5, before we show the results of a numerical experiment estimating the Θ-temperature in section 6. Some concluding remarks finish the paper in section 7.

2. Model

We consider a bead–stick polymer model with continuous degrees of freedom in three dimensions. Monomers (beads) are placed at coordinates $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and are connected by bonds of fixed length,

$$\mathbf{b}_i = \mathbf{x}_{i+1} - \mathbf{x}_i, \qquad |\mathbf{b}_i| = b. \tag{1}$$

All monomers interact via a 12–6 Lennard-Jones potential

$$U(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \qquad \sigma = \frac{b}{2^{1/6}} \tag{2}$$

with their diameter σ chosen such that the minimum distance of the interaction equals the bond length b. In the following we will set $\epsilon = 1$ to fix the energy and temperature scales while assuming that $b = 1$ such that all distances are given in units of b. The complete Hamiltonian reads:

$$\mathcal{H} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} U(|\mathbf{x}_i - \mathbf{x}_j|) + \sum_{i=1}^{N-1} \delta(|\mathbf{x}_{i+1} - \mathbf{x}_i| - b). \tag{3}$$

In this notation the total number of beads is denoted by N and the length of the polymer is $L = N - 1$. Interactions of adjacent beads are, of course, constant $U(|\mathbf{x}_i - \mathbf{x}_{i+1}|) = -\epsilon$ due to their constant distance and could be omitted. We include them in the Hamiltonian because it leads to a less complicated program code.

3. A parsimonious Metropolis algorithm

One of the first and most widely applied Monte Carlo methods is the Metropolis algorithm [7]. It generates a Markovian chain of microstates μ_i that follow a Boltzmann distribution

$$P(\mu) \propto e^{-\beta E(\mu)}, \tag{4}$$

where $E(\mu) = \mathcal{H}(\mu)/\epsilon$ is the energy associated with the state μ in units of ϵ and $\beta = \epsilon/k_\mathrm{B}T$ is the inverse temperature in natural units. The transition to a new state μ_{i+1} is done in the following way:

- A new state ν is proposed by a random modification to μ_i. Typically, the probability for this proposal is symmetric $P_\mathrm{prop}(\mu \to \nu) = P_\mathrm{prop}(\nu \to \mu)$.
- The energy difference of the two states $\Delta E = E(\nu) - E(\mu_i)$ is determined and the acceptance probability $P_\mathrm{acc} = \min\left(1, e^{-\beta \Delta E}\right)$ is calculated.
- A (pseudo) random number $\xi \in [0, 1)$ is drawn from a uniform distribution.
- The modification is accepted ($\mu_{i+1} \coloneqq \nu$) if $\xi < P_\mathrm{acc}$ and rejected ($\mu_{i+1} \coloneqq \mu_i$) otherwise.

This sequence can be shortened if $\Delta E < 0$ (and $\beta \geq 0$) since such an update is always accepted.

If this procedure is followed, ΔE will always be calculated with the maximal accuracy that is possible with the respective data type, e.g., a relative accuracy of about 10^{-15} for 64 bit floating–point numbers. However, when doing the comparison in the last step this accuracy is hardly ever required. This raises the question if it is feasible to allow for less accurate representations of ΔE within the program and whether this can be exploited to accelerate the simulation.

We reformulate the algorithm and start with drawing the random number ξ. The condition

$$\xi < P_{\text{acc}} = \min(1, e^{-\beta \Delta E}) \tag{5}$$

is equivalent to

$$\Delta E < -\frac{\ln \xi}{\beta} \tag{6}$$

again assuming $\beta > 0$. If one is able to establish bounds for the change in energy $\Delta E \in [\Delta E_{\min}, \Delta E_{\max}]$ the decision can be made if either $\Delta E_{\max} < -\frac{\ln \xi}{\beta}$ (accept) or $\Delta E_{\min} \geq -\frac{\ln \xi}{\beta}$ (reject). If neither condition holds the estimate has to improved, i.e., the lower and upper bounds have to be narrowed. This approach will be useful if low-accuracy estimates of the change in energy can be obtained substantially faster than estimates with high accuracy or the exact value itself.

4. Binary Trees

Recently, Clisby [8] has introduced a method employing binary trees to the simulation of self-avoiding walks on periodic lattices and has achieved remarkable success. It is now possible to investigate walks with tens of millions of steps. We have adopted this method for walks in continuous space [9] and use the binary trees here as the fundamental data structure for the representation of the polymer.

The tree's leaves, i.e., nodes at the bottom that are connected to only a single other node, correspond to individual monomers and store this particular monomer's position. Nodes at higher levels represent larger parts of the monomers. If the tree is balanced each node in the second level represents the two monomers that correspond to the children-nodes, nodes in the third level correspond to groups of four monomers again comprised of the monomers from the children-nodes and so on. Our intent is to use these higher-level nodes in order to estimate the interactions between groups of polymers and for that purpose we must select which information are to be stored with them. For our current implementation we opted for a simple selection. The node stores the number of monomers it represents as well as the geometric information of a sphere – center $\mathbf{y}$ and radius r – which contains all those monomers. Additional data like the center of mass or higher moments could be used as well. Note that these data can relatively easily, i.e., with complexity $\mathcal{O}(1)$, be derived from the respective data stored in the children-nodes and that the data of the entire tree can thus be created bottom-up within $\propto N$ operations. An important property of this data structure is that transformations that are to be enacted during a Monte Carlo step can be applied to the higher-level nodes. If the monomers represented by a node are to be rotated collectively, the new position of its sphere can be derived directly from the old one without explicit calculation of the individual new monomer positions.

In order to estimate the interaction between groups of monomers represented by two spheres we exploit that the Lennard-Jones potential (2) is monotonic for $r > b$. If two nodes A and B represent groups of n_A and n_B monomers located in spheres around the points $\mathbf{y}_A$ and $\mathbf{y}_B$ with radii r_A and r_B it is clear that for any monomer $k \in A$ and $l \in B$ it is

$$|\mathbf{y}_A - \mathbf{y}_B| - r_A - r_B < |\mathbf{x}_k - \mathbf{x}_l| < |\mathbf{y}_A - \mathbf{y}_B| + r_A + r_B. \tag{7}$$

For the interaction between the two nodes

$$E_{AB} = \sum_{k \in A} \sum_{l \in B} U(|\mathbf{x}_k - \mathbf{x}_l|) \tag{8}$$

it follows that if $|\mathbf{y}_A - \mathbf{y}_B| - r_A - r_B \geq b$, i.e., if $U(r)$ is monotonic in the relevant interval, we can obtain boundaries $E_{AB} \in [E_{AB}^{\min}, E_{AB}^{\max}]$ given by

$$E_{AB}^{\min} = n_A n_B U(|\mathbf{y}_A - \mathbf{y}_B| - r_A - r_B) \tag{9}$$

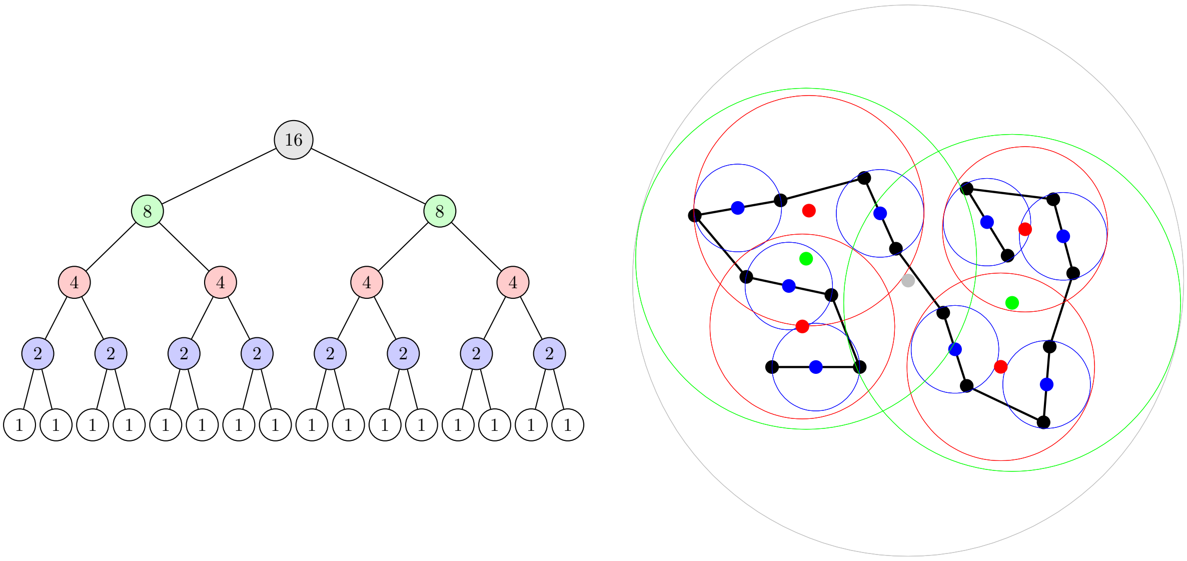

Figure 1. (Left) Binary tree with the numbers of monomers each node represents. (Right) Two-dimensional sketch of the way geometric information from groups of monomers is aggregated as spheres on different levels in the binary tree.

and

$$E_{AB}^{\max} = n_A n_B U(|\mathbf{y}_A - \mathbf{y}_B| + r_A + r_B). \tag{10}$$

If the obtained estimate is not precise enough it can be improved by involving nodes at the next lower level. If the children of node A are labeled A_l and A_r an almost always better[1] estimate is given by

$$E_{AB} = E_{A_l B} + E_{A_r B} \in [E_{A_l B}^{\min} + E_{A_r B}^{\min}, E_{A_l B}^{\max} + E_{A_r B}^{\max}]. \tag{11}$$

In Fig. 1 the binary tree is displayed in its balanced form, where each inner node is parent to two subtrees of equal size. When a Monte Carlo step is performed it can be useful to alter this shape. In our simulation we used two different types of steps. One is the pivot move, where from one fulcrum monomer $\mathbf{x}_p$ on all monomers $\mathbf{x}_k$ with $k \in \{p+1, \ldots, N\}$ are rotated by a random angle around an axis through $\mathbf{x}_p$. The other move is best described using the bond vectors $\mathbf{b}_i$. One of them is randomly changed $\mathbf{b}_{p+1} \to \mathbf{b}'_{p+1}$ while all others remain the same. This amounts to a constant shift $\mathbf{x}_k \to \mathbf{x}_k + \mathbf{b}'_{p+1} - \mathbf{b}_{p+1}$ for all monomers with $k \in \{p+1, \ldots, N\}$. In both cases the polymer decomposes into two parts one of which remains unaltered while the other is modified. It is convenient to rearrange the tree such that each part is represented by an individual node (Fig. 2). This can be done by a sequence of so-called tree–rotations (Fig. 3). Once the tree is in the new shape the change in energy can be estimated by simultaneously estimating the interactions between these two nodes in their pre–update state as well as with the right node's sphere which contains the modified monomers in the post–update position. Both guesses can be improved in parallel until ΔE is known with sufficiently high precision and a decision can be made.

5. Performance

In order to evaluate the performance of the method we performed simulations using our algorithm and a naïve method for different sizes at $T = 4$ which is slightly above T_Θ (see below). The

[1] In rare cases rebalancing in a node's subtree can cause the sphere in one of that node's children to partly lie outside of the node's own sphere even though the monomers are still contained inside it. This can cause bounds of ΔE to become wider instead of narrower.

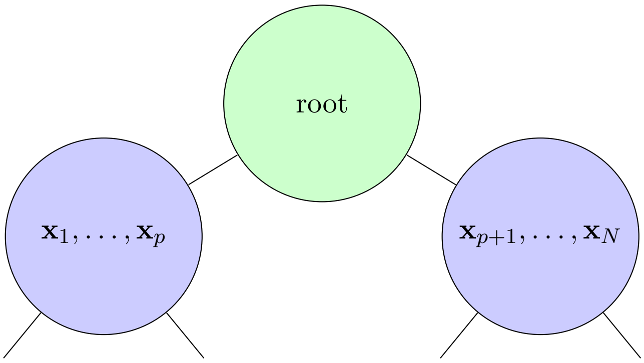
root
$\mathbf{x}_1, \ldots, \mathbf{x}_p$
$\mathbf{x}_{p+1}, \ldots, \mathbf{x}_N$

Figure 2. Top part of a binary tree that has been arranged to facilitate an update of the monomers $\mathbf{x}_1, \ldots, \mathbf{x}_p$. Note that as before the displayed nodes do not store the $\mathbf{x}_i$ but the data for a sphere containing them.

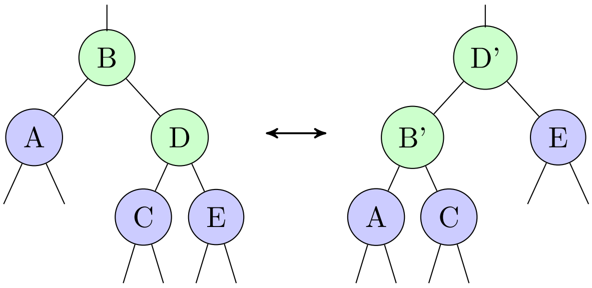

Figure 3. Tree-rotations are used to transform the binary before and after a Monte Carlo step. Only the nodes B and D are being modified and the order from left to right is preserved.

results are shown in Fig. 4. As expected the time required by the standard algorithm grows proportionally with L^2 as soon as the small overhead becomes negligible. In contrast, the more complicated data structures used with our method lead to a slower simulation for very small systems, but it soon ($L > 100$) becomes more efficient when the polymer length is increased. Moreover, the data clearly shows that the time needed for individual updates grows slower than linearly and will for larger sizes than the ones we investigated likely beat $\mathcal{O}(\sqrt{L})$.

6. Critical temperature

At the Θ–point a large enough polymer at large enough scales is supposed to take the shape of a GRW. For smaller chains there might not be a temperature where this behavior is completely realized, but we still use a characteristic of GRW to identify the transition temperature. For a GRW the end–to–end vector $\mathbf{r}_{ee} = \mathbf{x}_{L+1} - \mathbf{x}_1$ is distributed according to a Gaussian distribution

$$P_{3d}(\mathbf{r}_{ee}) = \frac{1}{(2\pi\lambda^2)^{3/2}} e^{-\mathbf{r}_{ee}^2/2\lambda^2}. \tag{12}$$

Hence the distribution of the scalar end–to–end distance $r_{ee} = |\mathbf{r}_{ee}|$ is

$$P_{1d}(r_{ee}) = 4\pi r_{ee}^2 \frac{1}{(2\pi\lambda)^{3/2}} e^{-r_{ee}^2/2\lambda^2} \tag{13}$$

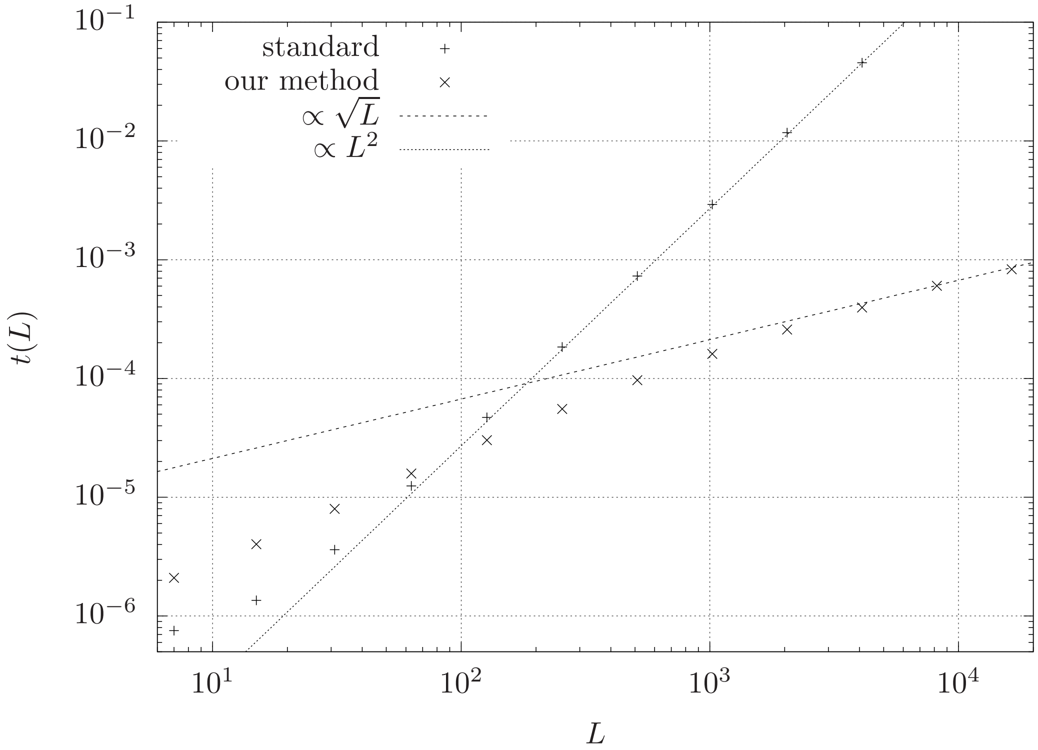

Figure 4. A comparison of the required times (in seconds) for a combination of a pivot and a bond-rotation update with a naive method and our algorithm at $T = 4$.

and it is

$$\frac{P_{1d}(r_{ee})}{r_{ee}^2} \propto e^{-r_{ee}^2/2\lambda^2}. \tag{14}$$

This means that there is a linear relation between $\ln\left(P_{1d}(r_{ee})/r_{ee}^2\right)$ and r_{ee}^2 for a GRW and consequently that a walk is behaving like a GRW if the second–order contribution a_2 of a quadratic fit,

$$\ln\left(P_{1d}(r_{ee})/r_{ee}^2\right) \approx a_0 + a_1 r_{ee}^2 + a_2 (r_{ee}^2)^2, \tag{15}$$

of this relation vanishes. We can, therefore, use the respective temperature as a finite-size estimate $T_\Theta(L)$ for the temperature of the Θ-point. The relevant data for a chain with $L + 1 = 4096$ is shown in Fig. 5.

It is neither entirely clear which analytical form the finite–size Θ-temperatures should take nor how the method of determining $T_\Theta(L)$ influences which corrections to scaling apply. Pragmatically, we have chosen from among multiple plausible approaches listed in Ref. [6], the one that works best,

$$\frac{1}{T_\Theta(L)} = \frac{1}{T_\Theta} + \frac{c_1}{\sqrt{L}} + \frac{c_2}{L}. \tag{16}$$

The fit (shown in Fig. 6) done for sizes $L > 100$ neatly describes the obtained values and provides $T_\Theta = 3.916(3)$, $c_1 = 0.26(1)$, and $c_2 = 10.2(2)$. However, as can be seen in the inset for large L the data shows a slightly more shallow slope than the fit. Using only the four largest sizes $L > 2000$ we obtain $T_\Theta = 3.907(2)$.

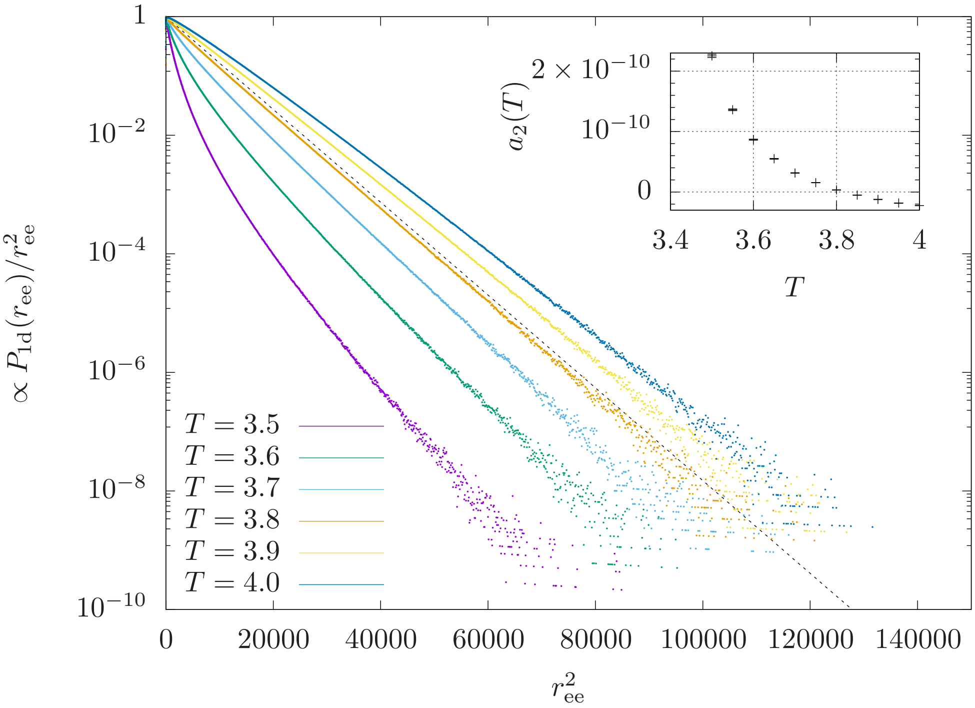

Figure 5. Distributions of the scalar end-to-end distance divided by r_{ee}^2 for different temperatures and $L+1 = 2^{12}$. The curves are normalized to unity at $r_{\mathrm{ee}}^2 = 0$. The dashed line is a guide to the eye. Inset: Fitting the quadratic ansatz from Eq. (15) leads to second-order coefficients $a_2(T)$ that change sign close to $T = 3.82$.

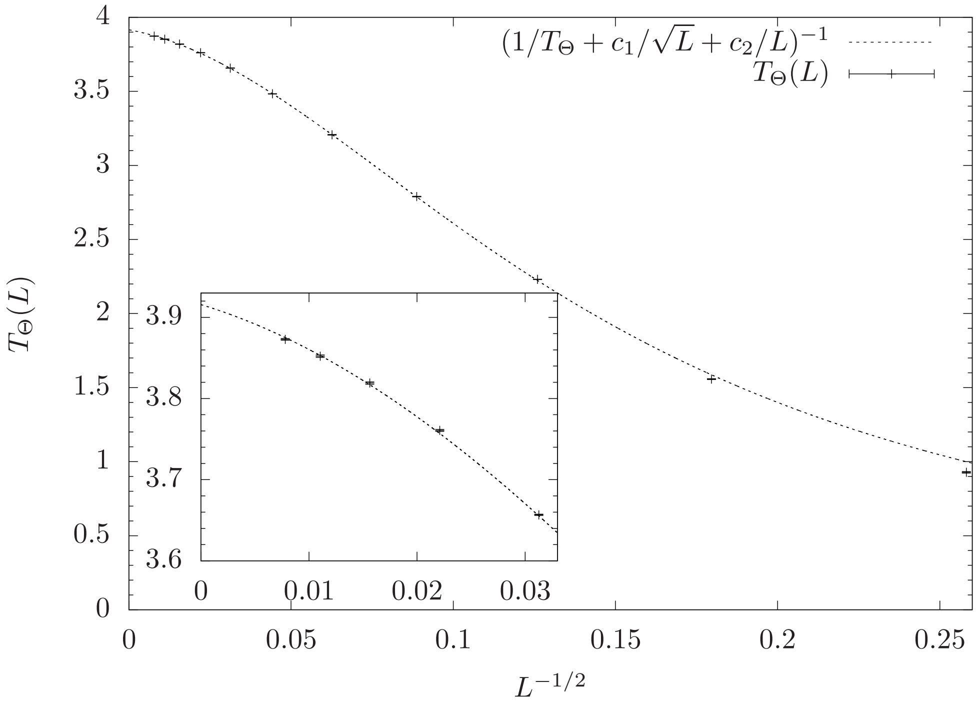

Figure 6. Finite-size estimates for the Θ-temperature. The three rightmost data points ($L < 100$) were excluded from the fit. The inset shows a close-up of the main plot for large L.

7. Conclusion

In this study we have shown how the Metropolis algorithm can be executed in a parsimonious manner. It improves the performance of the method if the change in energy associated with a conformational update can be estimated sufficiently faster than exactly calculated. When applied to a polymer with long-range 12–6 Lennard-Jones at a temperature close to the Θ-point we find that asymptotically the time required for single updates scales with $\mathcal{O}(\sqrt{L})$ or better which is a substantial improvement over the quadratic scaling required by standard methods. We were able to simulate chains with $L + 1 = 16384$ repeat units, i.e., a Hamiltonian with about 10^8 interactions.

Using the distributions of the end–to–end distance we estimated the Θ–temperature as a function of chain length. A simple finite-size scaling analysis suggests that in the infinite chain limit $T_\Theta \approx 3.91$.

Acknowledgement

The project was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant No. 189 853 844–SFB/TRR 102 (project B04). We thank Nathan Clisby for helpful remarks.

References

[1] Grassberger P and Hegger R 1995 *J. Chem. Phys.* **102** 6881
[2] Frauenkorn H and Grassberger P 1997 *J. Chem. Phys.* **107** 9599
[3] Graessley W W and Hayward R C 1999 *Macromolecules* **32** 3510
[4] Rampf F, Paul W and Binder K 2006 *J. Polym. Sci. Part* B: *Polym. Phys.* **44** 2542
[5] Li B, Sun Z, An L and Wang Z 2016 *Sci. China: Chem.* **59** 619
[6] Vogel T, Bachmann M and Janke W 2007 *Phys. Rev.* E **76** 061803
[7] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E 1953 *J. Chem. Phys.* **21** 1087
[8] Clisby N 2010 *J. Stat. Phys.* **140** 349
[9] Schnabel S and Janke W 2020 *Comput. Phys. Commun.* **256** 107414