

Please cite this article as:

Andrzej Grzybowski, Zbigniew Domański, A sequential algorithm for modelling random movements of chain-like structures, *Scientific Research of the Institute of Mathematics and Computer Science*, 2011, Volume 10, Issue 1, pages 5-10.

The website: <http://www.amcm.pcz.pl/>

Scientific Research of the Institute of Mathematics and Computer Science, 1(10) 2011, 5-10

A SEQUENTIAL ALGORITHM FOR MODELLING RANDOM MOVEMENTS OF CHAIN-LIKE STRUCTURES

Andrzej Grzybowski, Zbigniew Domański

*Institute of Mathematics, Czestochowa University of Technology, Poland
azgrzybowski@gmail.com*

Abstract. We introduce a sequential cellular-automata-like algorithm enabling efficient sampling of vast search spaces related to the kinetics of long polymers. As an example, with the help of our algorithm we study the movements of linear polymers in the vicinity of a flat membrane with holes in it.

Introduction

Chain-like structures are common in Nature, and thus they are widely encountered in biological, physical and technological processes. An important example of such processes involves a biopolymer transfer through a cell membrane which is vital to virtually all living organisms [1-3]. Polymers are molecules formed with a large number of covalently bonded, small repeating units called monomers. Their linear lengths are large compared to the molecular diameter, so that polymers are seen as long flexible strings. Within one of the possible and useful physical pictures, these strings can be thought of as long chains undergoing a random walk composed of steps of fixed step length followed by a random change in direction after each step.

From the physical point of view, polymers being classical systems can be studied by computer simulation. However, the complexity of possible polymer conformations along with the volume of a solvent make the search space for a hypothetical algorithm huge so that real polymers have to be mapped onto significantly simplified models [4-6]. On the other hand, the simplifications of a polymer model can be less harmful with respect to the real polymer if the search space viewed by an algorithm, appropriate to this given model, may be quickly sampled in a reliable manner.

The aim of this work is to present a numerical tool for the quick analyzing of the kinetics of chain-like structures from the statistical point of view. We do that by introducing a sequential algorithm built up around probabilistic-cellular-automata methodology, i.e. temporal evolution of a polymer within its configuration space is drawn with the help of the probabilistic transition rules among accessible polymer conformations. The cellular automata philosophy reduces the total number of

degrees of freedom of the polymer by restricting the allowed moves to the steps alignment with the edges of an underlying lattice.

Below, we define our algorithm in a rather formal way. As a bridge between this abstract approach and the world of statistical physics, we consider the kinetics of a polymer lying on the plane perpendicular to the flat membrane. The polymer is modelled as a string of identical bars connected by point-like joints. For the sake of clarity, we restrict ourselves to the simplest case of a so-called freely-jointed chain, i.e. a polymer whose bonds are uncorrelated.

1. Idea of sequential algorithm introduction - basic definitions and assumptions

In the description of the algorithm, we use terms and denotations, which we define below.

An abstract *2D chain position* is a finite sequence $\mathbf{c} = \{c_1, c_2, \dots, c_n\}$ of *2D* points $c_i = (x_i, y_i)$ such that distance d between any pair of its successive elements is less than given limit $g : d(c_i, c_{i+1}) < g, \quad i = 1, 2, \dots, n-1$.

The elements of sequence \mathbf{c} are called *segments* of the chain. Segment c_n is called the *head* of the chain, while element c_1 is called the *tail*. Number n is called the *length* of the chain.

Assumption 1 (discretization of motion space): The chain moves along the integer lattice nodes, i.e. coordinates (x_i, y_i) of the chain segments are integer.

The *movement trajectory* is a sequence of consecutive chain positions stored in matrix \mathbf{C} whose i -th row is interpreted as a chain position at moment i . Thus element c_{ij} denotes the position of segment j at moment i .

The movement of the chain consists of a sequence of *moves* which transforms the chain from one position to another.

Assumption 2 (sequentialization of the move): Every single move of a chain can be sequentialized into a sequence of *steps*, i.e. moves made sequentially by chain segments. A single step *may* transform given segment c_j only to one of its neighbouring nodes, i.e. $d(c_{ij}, c_{i\pm 1, j}) < g$.

The *First to move* (FTM) segment is the segment which in a given move is chosen by the algorithm to make the step as the first from all the chain segments. The choice is realized according a given probability distribution defined on the chain segments. The distribution will be denoted as FTMD.

The steps may be influenced by an *outer law* given by a probability distribution defined on the neighbouring nodes. The *outer law probability distribution* (OLPD) may depend on the position of the segment in the motion space. The OLPD reflects

the existence of restrictions and constraints imposed on the system from the environment as well as by the laws of physics.

The step made by the FTM segment as well as all the following steps may be subject to some **addition restrictions connected with the assumed features** of the chain. These restrictions will be denoted as AFR. For example, one of such restrictions is the upper limit for the distance between the successive segments. This restriction assures continuity of chain, and - by assumption - it does not concern the FTM segment. Another example of the AFR restrictions may be a requirement that in a given node, at most a given number of segments can be placed, etc.

Assumption 3 (*chain nature of the move*): Every single move of a chain is started by only one segment, chosen according to the FTMD. Then all the segments make their steps sequentially according to the OLPD and the AFR.

The above assumption allows us to simulate the movement of the described chain structure effectively and efficiently. However, in many practical problems, such as biopolymer behaviour inside a living tissue, one should also take into account some additional constraints connected with the biochemical nature of the system. Thus we define additionally the cost connected with chain structure. The **structure** of the chain is defined by the relative mutual positions and related interactions of the segments. The **cost** of the chain position and its structure is function F representing its fitness connected with its structure and/or other external (environmental) properties. The lower cost, the better *fitness* of the chain structure and position.

Assumption 4 (*acceptance of new chain position*): The new position of the chain is accepted (by Nature) with a probability depending on its cost.

The above assumptions and ideas are implemented in the following **sequential algorithm for a chain-like structure movement simulation**.

Step 0. (*Initialization*) Set the initial (current) **chain position** c_{curr} and evaluate its **current cost** function value F_C .

Step 1. (*FTM segment selection*) According the given FTMD, select FTM segment $c_{curr,f}$.

Step 2. (*Step choice for FTM segment*) According the given OLPD and AFR, select the neighbouring node for the next position of the segment $c_{new,f}$.

Step 3. (Move complement - successive steps of remaining segments)

To obtain a new chain position, c_{new} , sequentially choose segments $c_{new,i}$, $i = f - 1, K, 1$ and draw the neighbouring nodes for their next position according the OLPD and AFR. This process is terminated for the first k , $k = f - 1, \dots, 1$ for which the following condition holds: $d(c_{curr,k}, c_{new,k+1}) < g$. If $k > 1$, then for $i = 1, K, k$ we assume $c_{new,i} = c_{curr,i}$. Next sequentially choose segments c_i , $i = f + 1, K, n$ and draw the neighbouring nodes for their next position according to the OLPD and AFR. This process is terminated for the first k , $k = f + 1, K, n$ for

which the following condition holds: $d(c_{curr,k}, c_{new,k-1}) < g$. If $k < n$, then for $i = k, K, n$ we assume $c_{new,i} = c_{curr,i}$

Step 4. (Acceptance of new position)

Compute the cost of new position F_N and the difference $d = F_N - F_C$. If $d < 0$, accept c_{new} . Alternatively, accept c_{new} only if random variable U having a uniform p.d. on interval $[0,1]$ satisfies $U \leq A(d)$. If c_{new} is accepted, then c_{curr} is replaced by c_{new} ; else c_{curr} remains as is.

Step 5. Terminate the algorithm if the stopping criterion is met; else return to Step 1.

Step 6. Return the final position of the chain, its cost and various statistics connected with the simulated movement trajectories.

The nondecreasing real function $A(d)$ that appears in Step 4 of the above algorithm represents the attitude of Nature towards the acceptance of worse states. If Nature accepts all states, one may assume $A(d) = 1$. Otherwise, similarly as in the famous Metropolis algorithm, we propose to use function $A(d) = \exp[-d/T]$, where T is a parameter which may be additionally subject to change during the movement process, see e.g. [7].

2. Example of computations

To illustrate the possible usage of the introduced algorithm, we consider a simple example of a chain-like structure passing through a hole in a boundary. Let us consider the situation presented in Figure 1. The initial chain position is the following: $\{(24,2), (25,3), (24,4), (25,5), (24,6), (25,7), (24,8), (25,9), (24,10), (25,11), (24,1), (25,13), (24,14), (25,15), (24,16)\}$. The length of the considered chain equals $n = 15$. In our example, function $A(d) = 1$, and it reflects the fact that here the moves of the chain are not dependent on its structure. We also assume that the OLPD says that the most probable neighbouring nodes are those in the upper right directions, so the chain is expected to move up and right rather than in other directions. The FTMD assigns to each segment its probability to be chosen as the FTM one proportional to its distance from the tail (the head is the most probable segment to move as the FTM). Additionally, we assume the existence of a boundary in the motion space. It is given by straight line $y = 20$. However, in this boundary, there is a hole at points $(30,20)$ and $(31,20)$, compare Figure 1a.

Now, let us assume, that we want to study how many moves are required for the chain to place the first of its segments into the hole. Obviously, it is a random number, but we can estimate it with the help of Monte Carlo simulations. In the presented run, this event was realized at move 74, and the chain position at that moment is presented in Figure 1b.

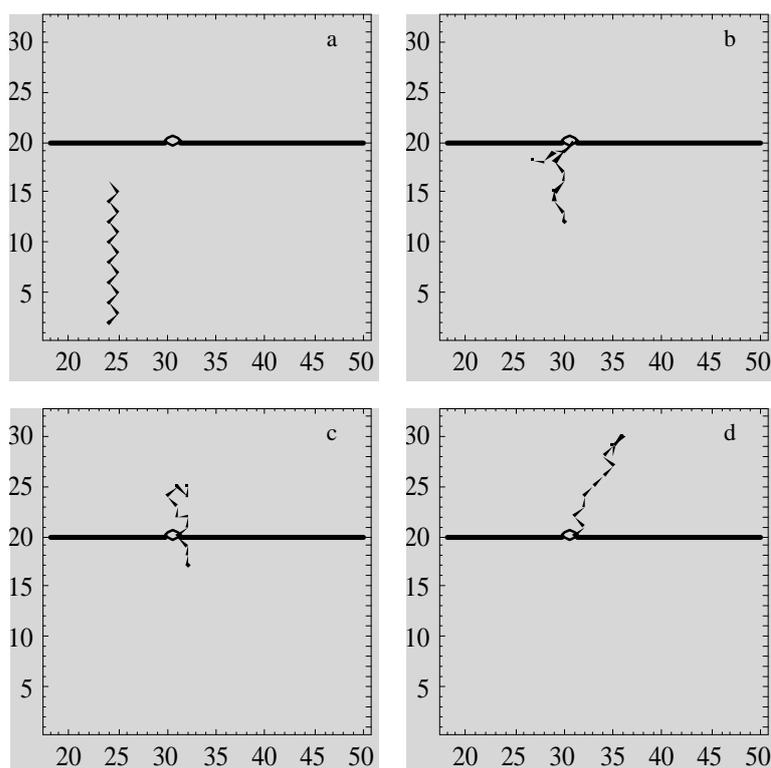


Fig. 1. Kinetics of chain-like structure: a) initial position of 15-segment length chain; b) position after 74 moves - first time a chain segment is placed into hole; c) polymer is passing through hole - position after move 122; d) position after move 158 - entire chain traversed hole

Next, it may be interesting in how many moves it takes to get the entire chain outside the hole. In the analyzed run, it happened at move 158, and the position is presented in Figure 1d. Many other statistical characteristics of the movement process as well as the influence of various parameters describing both the chain structure itself and the distributions involved in the process can be easily examined with the help of the presented algorithm. It is worth emphasising that the algorithm is very efficient. We have obtained our results with relatively little numerical effort, modest memory and CPU resources. The simulations can be even run on netbooks.

Remarks

We applied our algorithm to a simple freely-joined model of a polymer being aware of its serious shortcoming, i.e. the freely-joined model does not take into account the interactions among the monomers. Moreover, the interactions appear-

ing between the monomers and solvent molecules are ignored. We have chosen such a simple model only because of the transparency of the algorithm presentation, e.g. we have not used Assumption 4 and Step 4. Such a drawback, however, can be easily overcome by an appropriate account of the monomer-monomer as well as monomer-solvent interactions within Assumption 4. Applications of the presented algorithm to more realistic polymer models are in progress.

References

- [1] Meller A., Nivon L., Branton D., Voltage-driven DNA translocations through a nanopore, *Phys. Rev. Letters* 2001, 86, 3435-3438.
- [2] Kantor Y., Kardar M., Anomalous dynamics of forced translocation, *Phys. Rev.* 2004, E69, 21806.
- [3] Gershow M., Golovchenko J.A., Recapturing and trapping single molecules with a solid-state nanopore, *Nature Nanotechnology* 2007, 2, 775-779, 2007, doi:10.1038/nnano.381.
- [4] Meller A., Dynamics of polynucleotide transport through nanometre-scale pores, *J. Phys.: Condens. Matter* 2003, 15, R581-R607.
- [5] van Leeuwen J.M., Drzwiński A., Stochastic lattice models for the dynamics of linear polymers, *Physics Reports* 2009, 475, 53-90, doi: 10.1016/j.physrep.2009.04.001.
- [6] Drzwiński A., van Leeuwen J.M., Crossover from reptation to Rouse dynamics in the extended Rubinstein-Duke model, *Phys. Rev.* 2007, E77, 031802, doi: 10.1103/PhysRevE.77.031802.
- [7] Spall J.C., *Introduction to Stochastic Search and Optimization; Estimation, Simulation, and Control*, A John Wiley & Sons. Inc., Publication, 2003.