

# Simulating a Simple Branched Polymer Using Monte Carlo Methods

H.C. Bronk

A thesis submitted for the degree of  
Bachelor of Science in Physics and Astronomy  
Utrecht University

Supervisor: Prof. G.T. Barkema

Februari, 2013

## Abstract

In this report we present a model for a simple branched polymer as a collection of stiff rods and nodes. We study the effect of the inverse temperature  $\beta$ , elasticity  $\kappa$ , chemical potential  $\mu$  and polymer length  $L$  on the radius of gyration, by building a Monte Carlo simulation that allows for a variation in the polymer's length and number of nodes. We use the finite size scaling method to achieve a series of data collapses that lead to an expression for the radius of gyration as a function of  $\beta\kappa$  and  $\beta\mu$ . We further find that  $R_g \sim N^{0.35}$ .

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Model</b>	<b>2</b>
2.1	Monte Carlo simulation . . . . .	3
2.2	Moves . . . . .	4
2.2.1	Energy invariant moves . . . . .	4
2.2.2	Vibration . . . . .	5
2.2.3	Adding or removing a node . . . . .	5
2.3	Radius of gyration . . . . .	6
<b>3</b>	<b>Results</b>	<b>7</b>
3.1	Summary and conclusion . . . . .	10

# 1 Introduction

A biological virus is a small agent that is responsible for a wide range of human diseases. One of the reasons they are intensively studied is to develop medicines that combat diseases, and vaccines to prevent the spread of infections. A key step in the spread of a virus is the insertion of its genetic information in a living cell of the infected organism, which is necessary for its reproduction. In this report we present a study of a model that allows for the calculation of the physical size of the genomic information, which could lead to an explanation of the mechanism that inserts it into the host cell.

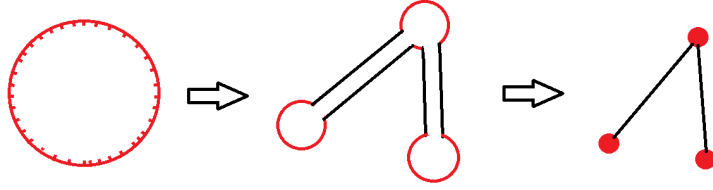
A virus particle, or virion, is made up of nucleic acid and an enveloping capsid, which is a protective protein layer. The nucleic acid is a polymer consisting of nucleotides that encode the genetic information either in DNA or RNA form. In this work we assume the genetic information is stored in RNA. One theory that explains how the RNA leaves the virion and enters the host cell states that the nucleic acid inside the capsid is under high pressure. After the virion connects to the cell a hole is opened that causes the RNA to flush into the cell. The host then replicates the RNA many times, and encapsulates the RNA strains in new capsids. This creates so many virus particles inside the host cell that it ruptures and dies, allowing the new virions to infect other cells.

To determine the validity of the theory suggesting pressure difference as the transportation method it is useful to compare the size of an unconstrained polymer and a capsid. A measure one could use for such a comparison is the radius of gyration  $R_G$  (see section 2.3), which is the average distance of the nucleotides to the RNA strain's center of gravity. As De Gennes<sup>1</sup> has shown,  $R_G \sim N^\alpha$ , where  $N$  is the number of nucleotides and  $\alpha = \frac{1}{4}$ . This was verified by Kurt<sup>2</sup>, who used Monte Carlo methods to simulate a simple branched polymer composed of a fixed number of stiff rods. He found  $\alpha = 0.26 \pm 0.03$ .

The aim of this work is to further explore the use of Monte Carlo methods to simulate simple branched polymers, by allowing a variation in the number of stiff rods. This gives the algorithm more flexibility and should allow it to travel through phase space in larger steps, but presents interesting theoretical problems. We subsequently use the simulation to find a function for the radius of gyration depending on the length of the polymer and the inverse temperature multiplied by the chemical potential,  $\beta\mu$ .

## 2 Model

This work focuses on circular, single-stranded RNA viruses, which means that the polymer resembles a spoked cart wheel without a central axis. This configuration allows the spokes on opposite sides of the wheel to bond together, so the RNA ring forms a stiff rod with loose nucleotides at the end points forming a semicircle. Similar flexible sites also allow for bends and branches in the polymer, which leads us to model the RNA polymer as a combination of stiff rods and flexible nodes, see figure 1. We make a further simplification by ignoring the physical and chemical properties of the nucleotides. This leaves a set of nodes with positions and connections to other nodes, which are said to be neighbouring nodes, or neighbours. The collection of these nodes is called the system, with a single instance in time being a state of the system. At certain times during a simulation the system state will be used to measure certain quantities such as the number of nodes and the radius of gyration. Hence, our goal is to take an initial system and change it each time step of the simulation in such a way that the process eventually reaches an equilibrium that accurately models a polymer.



**Figure 1:** *On the left is an example of a circular polymer, with all nucleotides free. In the middle is the same circular polymer, but with nucleotides on the black straight segments bonding with the nucleotides directly opposite their position. These segments form stiff rods, with the free nucleotides that make up the red circles providing flexibility. This structure can be simplified to a set of rods and nodes, as shown on the right.*

## 2.1 Monte Carlo simulation

The system is initialized as a single stiff rod between two nodes, from which we generate subsequent states by a process known as a Markov chain. The Markov chain is a process that randomly generates a new state from the current state, without taking previous states into account. This new state is selected and accepted with a certain transition probability. It has been proven that a Markov process can tend to any distribution by choosing the transition probabilities such that they satisfy the conditions of detailed balance and ergodicity. Since every state differs from its previous state by the application of a single, reversible move (see section 2.2), every system state is reachable from every other system state, satisfying the condition of ergodicity.

The condition of detailed balance states that<sup>3</sup>

$$p_\mu P(\mu \rightarrow \nu) = p_\nu P(\nu \rightarrow \mu), \quad (1)$$

where  $p_\mu$  is the probability of the system being in state  $\mu$ , and  $P(\mu \rightarrow \nu)$  is the probability of the system moving from state  $\mu$  to state  $\nu$ . Since we want the model to represent a physical polymer, the Markov chain should tend to the Boltzmann distribution once it reaches equilibrium. Therefore we choose  $p_\mu$  and  $p_\nu$  to be Boltzmann probabilities and obtain

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{p_\nu}{p_\mu} = e^{-\beta(E_\nu - E_\mu)}, \quad (2)$$

where  $E_\mu$  is the energy associated with state  $\mu$ . Moving from one state to another requires the simulation to both select and accept a new state, so by setting  $g(\mu \rightarrow \nu)$  as the probability of selecting state  $\nu$  when in state  $\mu$ , and  $A(\mu \rightarrow \nu)$  as the probability of accepting the move to state  $\nu$ , we arrive at

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{g(\mu \rightarrow \nu) A(\mu \rightarrow \nu)}{g(\nu \rightarrow \mu) A(\nu \rightarrow \mu)} = e^{-\beta(E_\nu - E_\mu)}. \quad (3)$$

Most moves have the same forward and backward selection probabilities, which cancel each other out. To satisfy the condition of detailed balance we are therefore only required to choose the acceptance ratio equal to the Boltzmann weight of the system's change in energy. According to the Metropolis algorithm, the optimal way of doing this is by choosing

$$A(\mu \rightarrow \nu) = \begin{cases} e^{-\beta(E_\nu - E_\mu)} & \text{when } E_\nu > E_\mu \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

This leaves us with calculating the energy change that occurs when moving from state  $\mu$  to state  $\nu$ , which we do by evaluating the Hamiltonian of our system. We design the Hamiltonian in such

a way that the polymer is allowed some flexibility in its length and number of nodes. A rigid polymer with a fixed number of nodes has less possible state transitions, which limits the speed with which the simulation comes to thermal equilibrium. To model the energy associated with the flexibility we use a spring like term:

$$E = \kappa \left( \sum_{i=1}^N l_i - L \right)^2. \quad (5)$$

Because polymers with a lot of nodes have many free nucleotides we let the energy increase linearly with the number of nodes:

$$E = \kappa \left( \sum_{i=1}^N l_i - L \right)^2 + \mu N. \quad (6)$$

An extra term is necessary to compensate for the fact that changing the number of nodes by  $\Delta N$  adds a factor  $\Lambda^{3\Delta N}$  to eq. (2), see section 2.2.3 for details. This gives us the Hamiltonian we use to model our system:

$$H(N) = \kappa \left( \sum_{i=1}^N l_i - L \right)^2 + \left( \mu - \frac{1}{\beta} \ln \Lambda^3 \right) N, \quad (7)$$

where  $N$  is the number of nodes,  $\kappa$  a spring constant,  $l_i$  the length of segment  $i$ ,  $L$  the ideal length of the entire polymer,  $\mu$  the chemical potential,  $\beta$  the inverse temperature and  $\Lambda$  the thermal wavelength. This Hamiltonian produces a system close to its ideal length, and with a number of nodes that depends on the inverse temperature and chemical potential. The polymer's length and the number of nodes are allowed to fluctuate to increase the simulation speed, by allowing more phase transitions.

The Boltzmann weight associated with a certain system is given by

$$\exp[-\beta H(N)] = \Lambda^{3N} \exp \left[ \beta \kappa \left( \sum_{i=1}^N l_i - L \right)^2 + \beta \mu N \right]. \quad (8)$$

This equation shows that it is possible to use  $\beta\kappa$  and  $\beta\mu$  as simulation parameters, instead of using  $\beta$ ,  $\kappa$  and  $\mu$  separately. The reduction makes it easier to characterize the model and explain its results.

## 2.2 Moves

The progression of the system along the Markov chain occurs through the use of five *moves*, which are transitions from one state into another. It is important that the collection of these moves satisfy the previously outlined conditions of detailed balance and ergodicity. We guarantee this by setting the Metropolis acceptance ratios such that each move individually satisfies these conditions.

### 2.2.1 Energy invariant moves

We first consider two moves that do not change the energy of the system at all. They merely change the configuration of the system without changing the number of nodes, or the length of the bonds between the nodes. These moves are *reflection* and *moveEnd*.

Reflection selects an anchor node and reflects all nodes on one side of it in a randomly chosen plane through the anchor node. MoveEnd takes one of the nodes that have only one neighbour, excluding the origin node for implementation reasons, and connects it to a randomly selected other node. The position of the node is then set such that the displacement to its neighbour is the same as it was before reconnecting it.

Neither of these nodes change the length of the bonds, nor the number of nodes. They therefore do not alter the energy of the system. Since the selection probabilities going forward and backward are also the same, the acceptance ratio equals one, and the move is accepted regardless of the direction.

### 2.2.2 Vibration

The most simple change possible to the state of the system that changes the energy is a slight displacement of one of its nodes. This corresponds to the vibration of the polymer in nature. The change in energy associated with such a displacement depends only on the first term in the Hamiltonian (eq. 7) and is equal to

$$\begin{aligned} dE_L = E_\nu - E_\mu &= k \left( \sum_{i=1}^n l_i - L \right)_\nu^2 - k \left( \sum_{i=1}^n l_i - L \right)_\mu^2 \\ &= k (dL_\nu^2 - dL_\mu^2), \end{aligned} \quad (9)$$

where we use  $dL_\mu$  to denote the difference between the ideal and actual length of the polymer in state  $\mu$ .

The forward and backward selection probabilities should depend on the number of nodes, and since this is the same before and after the state change they should both equal  $1/N$ . From this it follows that

$$\frac{A(\mu \rightarrow \nu)}{A(\nu \rightarrow \mu)} = e^{-\beta dE_L} \quad (10)$$

$$A(\mu \rightarrow \nu) = A(\nu \rightarrow \mu) = \begin{cases} e^{-\beta dE_L} & \text{when } dE_L > 0 \\ 1 & \text{otherwise} \end{cases} \quad (11)$$

which means that the probability of displacing a node and the probability of it changing back in the next step are equal, as required by the condition of detailed balance.

### 2.2.3 Adding or removing a node

We allow for the possibility of adding and removing nodes, so we include moves to that effect. This creates a model that is less restricted to a certain region of phase space than it would otherwise be.

Adding a node to the system is always done in between two other nodes for implementation reasons. The node is added by randomly selecting a segment of the chain between two nodes and placing the new node randomly within a uniform sphere centered around the middle of the segment, with a radius of half the segment's length. This prevents the program from adding nodes that increase the length of the chain by a large amount, which produces a state that is unlikely to be accepted. Undoing this operation simply requires the removal of a node randomly selected from all the nodes with two neighbours. Note that deletion of a node is forbidden if the node would lie outside the sphere formed around the midpoint between its two neighbours. This prevents the simulation from removing a node that can not be added back on the next time step.

This outline leads to the following selection probabilities:

$$g(\mu \rightarrow \nu) = \frac{1}{N_\mu - 1} \frac{\Lambda^3}{V} \quad (12)$$

$$g(\nu \rightarrow \mu) = \frac{1}{N_{\nu,2}} \quad (13)$$

$N_\mu$  refers to the number of nodes in the system in state  $\mu$ ,  $N_{\nu,2}$  is the number of nodes in state  $\nu$  that have two neighbours,  $\Lambda^3$  is the cubed thermal wavelength and  $V$  the volume of the sphere.

The change in energy associated with adding a node is

$$\begin{aligned} dE &= E_\nu - E_\mu = k(dL_\nu^2 - dL_\mu^2) + \mu - \frac{1}{\beta} \ln \Lambda^3 \\ &= d\tilde{E} - \frac{1}{\beta} \ln \Lambda^3 \end{aligned} \quad (14)$$

Substituting in eq. (3) gives us:

$$\frac{A(\mu \rightarrow \nu)}{A(\nu \rightarrow \mu)} = \frac{N_\mu - 1}{N_{\nu,2}} \frac{V}{\Lambda^3} e^{-\beta dE} = \frac{V(N_\mu - 1)}{N_{\nu,2}} e^{-\beta d\tilde{E}} \quad (15)$$

As before we always accept a move when it decreases the total energy, by setting the corresponding acceptance ratio to 1. This also gives us the acceptance ratio for the move that results in a higher energy.

$$A(\mu \rightarrow \nu) = \begin{cases} \frac{V(N_\mu - 1)}{N_{\nu,2}} e^{-\beta d\tilde{E}} & \text{when } d\tilde{E} > 0 \\ 1 & \text{otherwise} \end{cases} \quad (16)$$

and similar for removing a node:

$$A(\mu \rightarrow \nu) = \begin{cases} \frac{N_{\mu,2}}{V(N_\nu - 1)} e^{-\beta d\tilde{E}} & \text{when } d\tilde{E} > 0 \\ 1 & \text{otherwise} \end{cases} \quad (17)$$

## 2.3 Radius of gyration

The radius of gyration, also known as *gyradius*, can be used as a measure for the size of a polymer chain. It is defined as:

$$R_g^2 \equiv \frac{1}{N} \sum_{k=1}^N (\mathbf{r}_k - \mathbf{r}_m)^2, \quad (18)$$

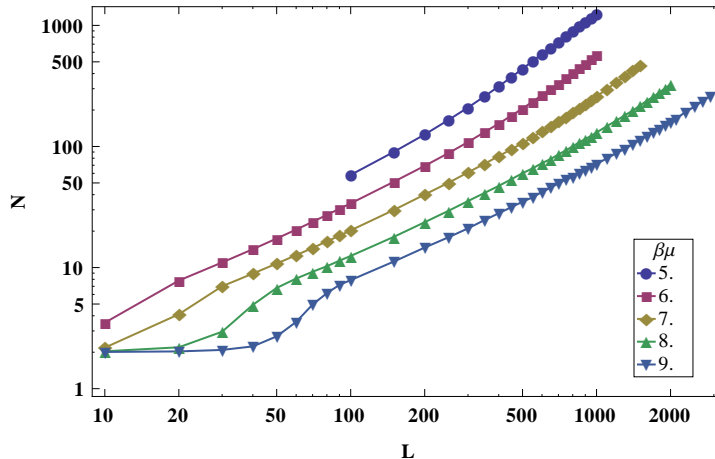
where  $\mathbf{r}_m$  is the polymer's center of gravity, and  $N$  the number of identical constituent parts. Its definition is similar to that of the variance of a discrete random variable, and if the polymer's center of gravity is chosen as mean, then the radius of gyration is an indicator of the spatial spread of the polymer.

Since this model uses a series of straight line segments instead of identical particles the above definition is not suitable and has to be modified. We do this by first calculating the radius of gyration contribution due to one line segment. At position  $x$  along the line,  $r(x)$  is the distance to the center of mass and  $\rho(x)$  the mass distribution along the line. The line has length  $l$  with a constant mass distribution, which gives us:

$$R_{g,1}^2 \equiv \frac{\int_0^l r^2(x) \rho(x) dx}{\int_0^l \rho(x) dx} = \frac{1}{l} \int_0^l r^2(x) dx \quad (19)$$

By applying the cosine rule and setting  $r_1$  and  $r_2$  as the distance between the endpoints of the line and the center of mass, we find that

$$r^2(x) = x^2 + x \frac{r_2^2 - r_1^2 - l^2}{l} + r_1^2 \quad (20)$$



**Figure 2:** A log-log plot of  $N$  as function of  $L$ , for several values of  $\beta\mu$ . At high  $\beta\mu$  and small  $L$ , the system stays in the ground state of two nodes most of the time. After  $L$  becomes sufficiently large, the number of nodes grows exponentially with  $\beta\mu$  and  $L$ .

Substituting eq. (20) into eq. (19) results in a formula for the contribution of the radius of gyration for one line segment.

$$R_{g,1}^2 = \frac{1}{2} \left( r_1^2 + r_2^2 - \frac{l^2}{3} \right) \quad (21)$$

Which means that the squared radius for the entire polymer is given by

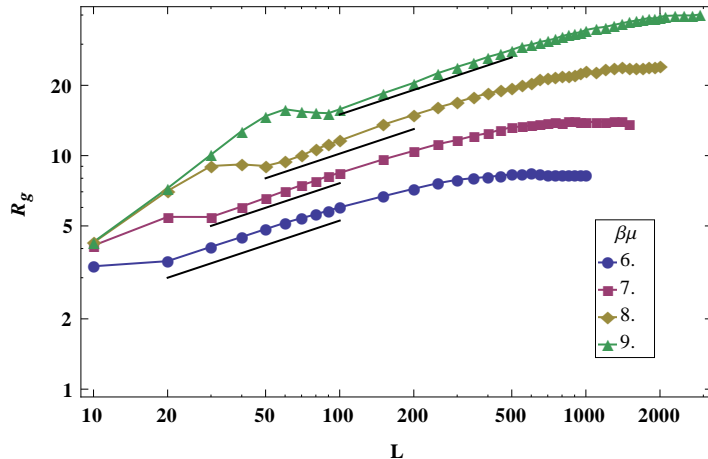
$$R_g^2 = \frac{1}{2(N-1)} \sum_{i=1}^{N-1} \left( r_i^2 + r_{i+1}^2 - \frac{l_{i,i+1}^2}{3} \right) \quad (22)$$

### 3 Results

To find a relationship between the radius of gyration, the polymer length and  $\beta\mu$ , we perform a series of simulations with  $10 \leq L \leq 3000$  and  $5 \leq \beta\mu \leq 9$ . For each combination of these parameters we simulate 50 runs with a simulation time long enough to take 500 measurements from an equilibrated system. This simulation time is longer for systems with a low  $\beta\mu$  and high  $L$ , because under these conditions the system equilibrium contains many nodes, which increases the thermalisation time. With three concurrent processes, the fifty runs per parameter combination can take between a few seconds and approximately 30 minutes to complete. The resulting number of nodes  $N$ , and radius of gyration  $R_g$ , in equilibrium, are averaged over all runs and shown in figures 2 and 3.

The log-log plot in figure 2 shows the exponential increase in nodes from increasing the polymer length and decreasing  $\beta\mu$ , which explains the exponentially increasing simulation time. However, longer polymers increase the probability of adding nodes only linearly, so we did not expect this result. The low number of nodes for small values of  $L$  ( $L < 100$  at  $\beta\mu = 9$ ) is because  $\beta\mu$  is so high that there is only a small probability of a third node being added to the system. For this reason measurements with  $\beta\mu > 9$  are not shown. The simulations with  $\beta\mu = 5$ , as shown by the blue dots, produce systems with so many nodes that the simulation becomes very slow. Likewise, the number of moves required before the system is thermalised becomes very large, making it unfeasible to accurately simulate configurations with  $\beta\mu < 6$ . Figure 3 shows the radius





**Figure 3:** A log-log plot of  $R_g(L)$ , with several values of  $\beta\mu$  shown with plot markers in the legend. After the initial irregular behaviour at small  $L$  values, for which the model was not designed, the increase in  $\log R_g$  has a constant slope of 0.35, as indicated by the black lines. After a certain value of  $L$ , depending on  $\beta\mu$ , the radius of gyration does not increase further.

of gyration as a function of the polymer length  $L$  for various  $\beta\mu$ . Note that we have not included the measurements taken from simulation runs with  $\beta\mu = 5$ . The straight black lines indicate the 0.35 slope, which matches the slope of the measurements for certain ranges of  $L$ , but is likely a coincidence. This shows that there is no exponent of  $\alpha = 0.25$  present in the data, as was expected from previous work. We further see the radius stabilize for large  $L$ .

The similar shape of the lines in both graphs suggest that a data collapse is possible. We achieve this by trying several exponents and judging the quality of the collapse by looking at the graph. The data is scaled in the horizontal direction with a factor  $(\beta\mu)^{-0.55}$  and shown in figure 4. The same exponent was used to scale figure 3 in the horizontal direction, and a further scaling in the vertical direction with a factor  $(\beta\mu)^{-1.38}$  results in figure 5. This data collapse allows us to calculate  $R_g$  for any value of  $\beta\mu$  and  $L$  as large as necessary for  $R_g(\beta\mu, L)$  to flatten out.

To also include  $\beta\kappa$  in the collapse we rerun the experiment with varying  $\beta\mu$ , several values of  $\beta\kappa$ , and  $L$  such that it corresponds to the value of  $\beta\mu$  at the right tail end of the lines in figure 3. By doing this we look at the effect of  $\beta\kappa$  at polymers that are so long that increasing their length further does not affect  $R_g$ . These results are shown in figure 6. We again attempt to collapse the data by eye and notice that a straight line fit is possible. The functions used to scale the data are shown in eq. (23) and eq. (24).

$$\eta(\beta\kappa, \beta\mu) = (\beta\kappa)^a \ln \beta\mu \quad (23)$$

$$\chi(\beta\kappa, R_g) = (\beta\kappa)^b \ln R_g(\eta) \quad (24)$$

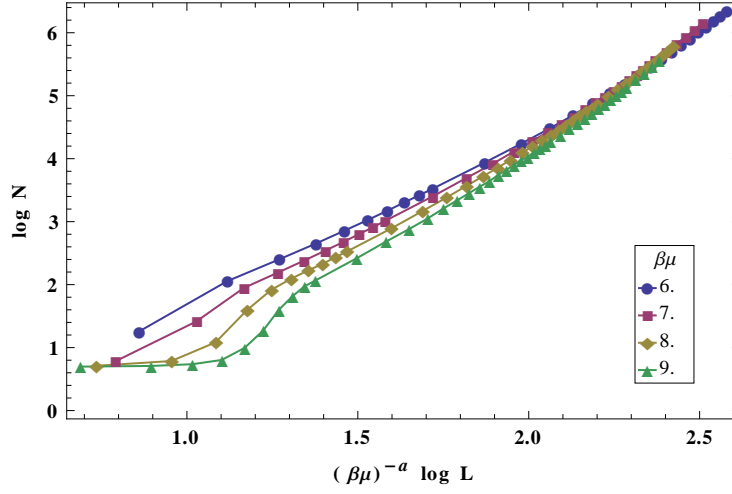
We equate  $\chi(\beta\kappa, R_g)$  to a function obtained from a linear model fit through the scaled data, which gives

$$\chi(\beta\kappa, R_g) = c + d\eta, \quad (25)$$

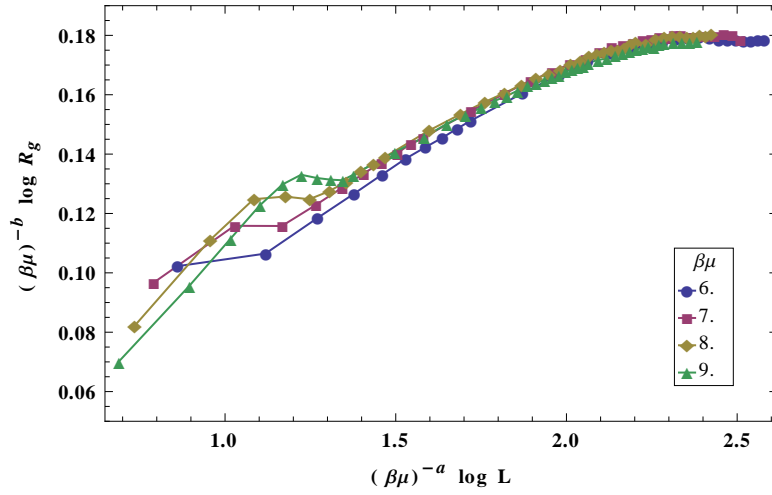
and subsequently:

$$R_g(\beta\kappa, \beta\mu) = \exp [(\beta\kappa)^{-b}(c + d(\beta\kappa)^a \ln(\beta\mu))] \quad (26)$$

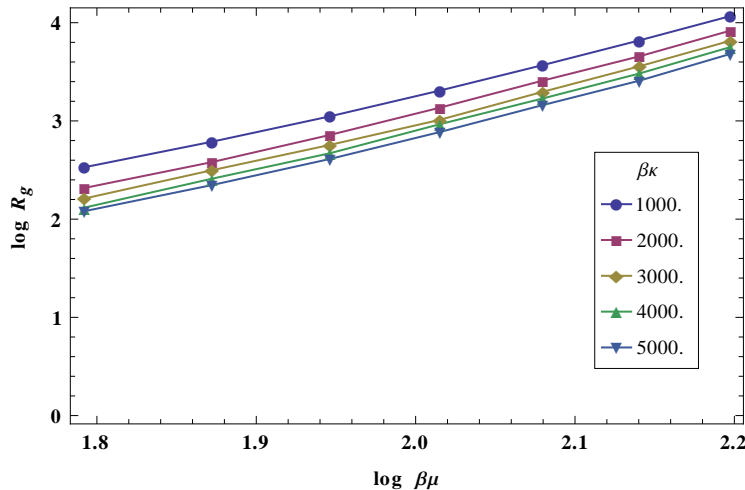
This function is used to estimate the measurements taken, and the resulting sum of the squared differences between the measurements and the estimates is used as error estimate. A parameter sweep over  $a$  and  $b$  allows us to minimize the error and find the corresponding best values for the fit



**Figure 4:** A collapse of the  $N(L)$  plot shown in figure 2, shown on a log-log scale. The horizontal axis is scaled with a factor  $(\beta\mu)^{-a}$ , with  $a = -0.55$ . This scaling results in a collapse of the lines when  $L$  is sufficiently large. Each line shown corresponds to a  $\beta\mu$  value as shown in the legend.



**Figure 5:** The data collapse corresponding to figure 3, showing  $\log R_g$  vs  $\log L$  scaled by factors  $(\beta\mu)^{-b}$  and  $(\beta\mu)^{-a}$  respectively.  $a = -0.55, b = -1.38$ . Lines are shown for several values of  $\beta\mu$ . The overlapping lines on the right hand side of the figure indicate the possibility of calculating  $R_g$  from  $\beta\mu, L$  and  $\beta\kappa = 5000$  directly, without performing a simulation.



**Figure 6:** A log-log plot of  $R_g$  as function of  $\beta\mu$ , for several values of  $\beta\kappa$ .  $L$  was chosen such that for each value of  $\beta\mu$  a further increase in  $L$  would not yield a change in  $R_g$ . The similar shapes of the lines indicate that a data collapse is possible.

$a$	$-0.05610 \pm 0.00001$
$b$	$-0.06581 \pm 0.00001$
$c$	$-2.93 \pm 0.04$
$d$	$3.68 \pm 0.03$

**Table 1:** Fit parameters for eq. (26) and their standard errors corresponding to the 95% confidence interval.

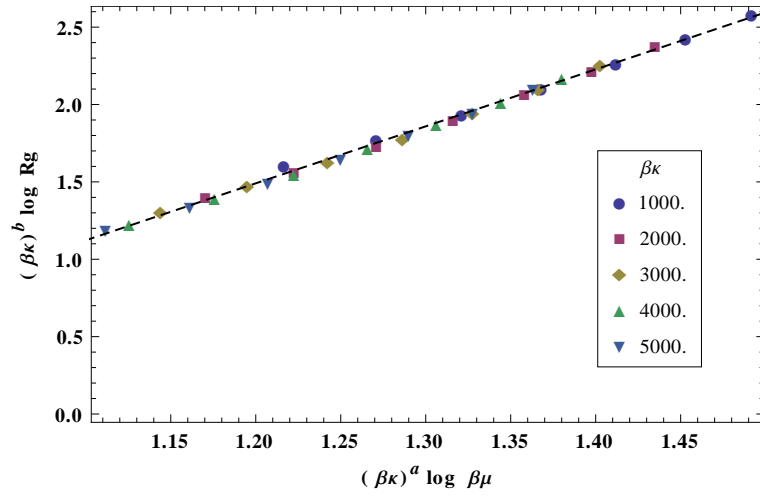
parameters. The parameters are shown in 1, and the collapsed data with the fitted line in figure 7.

### 3.1 Summary and conclusion

We modeled a simple branched polymer as a collection of nodes and edges and used this model to construct a Monte Carlo simulation. Using this simulation we have measured the radius of gyration for various values of  $\beta\kappa$ ,  $\beta\mu$  and  $L$ , and achieved a data collapse that has resulted in  $R_g(\beta\kappa, \beta\mu) = \exp [(\beta\kappa)^{-b}(c + d(\beta\kappa)^a \ln(\beta\mu))]$ , with parameters  $a, b, c$ , and  $d$  as shown in table 1. We further found that the radius of gyration grows with the length of the polymer with an exponent  $\alpha \approx 0.35$ , indicating a discrepancy between our data and previous research.

## References

- [1] P.-G. De Gennes. Statistics of branching and hairpin helices for the dAT copolymer. *Biopolymers*, 6:715–729, 1968.
- [2] Hasan Kurt. Simulating RNA molecules by Monte Carlo methods. Bachelor’s thesis, Utrecht University, Utrecht, The Netherlands, May 2012.
- [3] M.E.J. Newman and G.T. Barkema. *Monte Carlo Methods in Statistical Physics*. Oxford University Press, 1999.



**Figure 7:** A data collapse of  $R_g(\beta\mu)$ , corresponding to figure 6, achieved by transforming the function to  $(\beta\kappa)^b \ln R_g((\beta\kappa)^a \ln \beta\mu)$ , where  $a = -0.05610$  and  $b = -0.06581$ . The straight line fit results in a formula for  $R_g(\beta\kappa, \beta\mu)$ , provided  $L$  is chosen sufficiently large.