Exploration and Simulation of Entropic Forces

Wieger Steggerda ¹
Supervised by Erik Verlinde ²

Institute for Theoretical Physics
University of Amsterdam

¹ wieger.steggerda@gmail.com
² e.p.verlinde@uva.nl

Bachelor thesis Physics, extent 12 EC, carried out between May 1 and August 31 2010. Revised in July 2011.
Abstract

January this year (2010), Erik Verlinde published his paper, *On the Origin of Gravity and the Laws of Newton*. In this paper the Dutch professor from the Institute for Theoretical Physics Amsterdam (ITFA), and also supervisor of this thesis, suggests that gravity is an entropic force. Until this day there is no other theory which has a reasonable explanation on why gravity exists, only the effects it causes. In this project multiple entropic forces will be looked into, from which one specific. Elasticity will be described with probability theory and simulated. Concluding that computers are very suit for calculating and visualising entropic forces. Furthermore entropic forces are able to give a very smooth and consistent force which makes them a possible candidate for the origin of fundamental forces.

Samenvatting (Dutch abstract)

Contents

1 Introduction Into Entropic Forces ........................................ 4
   1.1 Examples of Entropic Forces ................................... 5
      1.1.1 Pressure .............................................. 5
      1.1.2 Osmosis .............................................. 5
      1.1.3 Hydrophobic Effect ..................................... 6

2 Ideal Chain ........................................................................ 7
   2.1 Elasticity ..................................................................... 7
   2.2 Ideal Chain Model .................................................. 9
      2.2.1 Probability Density Function of a Random Vector ... 9
      2.2.2 Expectation Value of a Random Vector ............... 10
      2.2.3 Variance of a Random Vector ............................ 10
      2.2.4 Central Limit Theorem in Random Vectors .......... 10
      2.2.5 The Probability Density Function of the End to End Vector .................................................. 11
      2.2.6 The Expectation Value of the End to End Vector Length .................................................. 12
   2.3 Elasticity Explained With the Ideal Chain Model .......... 13

3 Simulating the Ideal Chain ................................................ 14
   3.1 Input and Output of the Simulation .............................. 15
   3.2 The results ................................................................ 16

4 Conclusion and Discussion .................................................. 17

A Mathematical Probabilities ............................................... 22
   A.1 Probability Density Function .................................... 22
   A.2 Expectation Value .................................................. 22
   A.3 Variance ............................................................... 22
   A.4 Normal Distribution ................................................ 23
      A.4.1 The derivative of the normal distribution .......... 24
      A.4.2 Integrating the normal distribution ............... 24
      A.4.3 The expectation value of the normal distribution . 25
      A.4.4 The variance of the normal distribution ............ 25
   A.5 Central Limit Theorem .............................................. 26

B Pseudo Code For Simulation of the Ideal Chain .................. 28
   B.1 Creating a Random Vector ...................................... 28
   B.2 The Basic Algorithm ............................................. 29
   B.3 Speeding Up the Basic Algorithm ................................ 30
1 Introduction Into Entropic Forces

Back in 1915, Einstein revolutionized physics with the general theory of relativity. Because the gravitational force on an object is proportional to the mass \( m \) and \( F = ma \), the gravitational acceleration doesn’t depend on \( m \). Einstein therefore stated that gravity is a geometric phenomenon; that gravity is a mere fictitious force. Nearly a century after (2010), Erik Verlinde published On the Origin of Gravity and the laws of Newton[1]. In this paper, the professor states that gravity is not a fundamental but an entropic force. This new insight not only describes gravity, it also explains why it exists.

In physics, especially theoretical physics, there isn’t a lot of research into entropic forces. Entropic forces aren’t fundamental and therefore not very interesting to a purist. However, the new perspective on gravity makes entropic forces a lot more interesting.

Entropic forces are caused by the second law of thermal dynamics. The law that states that entropy always increases. The force is present in systems were some macro states are more likely to occur, while there is no preferred micro state, or configuration. When, for example, \( N \) coins are flipped, and for every heads a line of 1 cm is drawn, the line’s size will be somewhere about \( N/2 \) cm long, if \( N \) is large enough. There is, of course, no time and force in this system, but it shows that the length of the line, the macro state, is likely to be somewhere around \( N/2 \) cm, whilst the results of the coins, the micro state, is completely random.

Let’s take a look at the thermodynamic identity,

\[
dE = TdS - PdV + \mu dN
\]

where \( E \) is the energy, \( T \) the temperature, \( P \) the pressure, \( V \) the volume, \( \mu \) the chemical potential and \( N \) the number of particles. The last two terms can often be neglected,

\[
dE = TdS
\]

If the entropy depends on some spatial variable \( x \), the derivative to that variable can be made so that the left side of the equation will result in a force,

\[
F(x) = \frac{dE}{dx} = T \frac{dS}{dx}
\] (1)

This formula is the essence of an entropic force, notice the proportionality with the temperature.
1.1 Examples of Entropic Forces

1.1.1 Pressure

Pressure is probably the most easy entropic force one can think about. In a gas it is very unlikely that all molecules are close to each other, even if the molecules wouldn’t interact with each other at all. The consequence is that gasses expand and thus produce pressure. The ideal gas law shows that the force is proportional to the temperature, as predicted in the introduction,

\[ P = \frac{NkT}{V} \]

1.1.2 Osmosis

Osmosis is an effect that, like pressure, is caused by the small probability to have high and low concentrations close to each other. Osmosis is the diffusion of molecules from a place of high concentration to a place of low concentration until the concentration on both sides is equal. When a partially-permeable membrane has a solution of different concentrations on both sides, water will flow through the membrane, from the low to the high concentration, trying to make the concentrations equal. In Figure 1 the water level rises and therefore creates a pressure on the membrane.

Figure 1: Schematic representation of osmosis. The water flow to the higher solution causes the water to rise, which causes pressure on the membrane.
1.1.3 Hydrophobic Effect

The hydrophobic effect occurs when water comes into contact with a non-polar molecules, such as oil. It causes water to separate from oil. Some plants, especially the lotus flower, have hydrophobic leaves. Water cannot attach to the leaves and falls of the leaf, making it hard for fungi to grow on the leaves. A $\text{H}_2\text{O}$ molecule can accept and donate two bonds with other water molecules, making a network of bonds. Near a hydrophobic substance, the water bonds are broken, creating less configurations for the network to appear in. Therefore the multiplicity is the highest if there is the least amount of contact between the water and the hydrophobic substance.
2 Ideal Chain

A polymer, see Figure 3, is a molecule that consists of a chain of similar nodes called monomers. There are natural polymers like starch, DNA and rubber from the rubber tree; and there are synthetic polymers like plastic, nylon and synthetic rubber. A common feature to polymers is elasticity. Since different polymers consist of different monomers, it is likely that elasticity isn’t emerged by the building materials but rather by the common shape of the polymers. The structure can be captured in vectors.

Let

- \( N + 1 \) be the number of monomers;
- \( r_i \) with \( i \in \{1, \ldots, N\} \) be the vector between monomers \( i \) and \( i + 1 \);
- \( l \) be the distance between monomers. So \( |r_i| = l \) for all \( i \);
- \( R = \sum_{i\in\{1,\ldots,N\}} r_i \) be the end to end vector. It is the vector from the first to the last monomer.

In this chapter it is essential to have a good understanding of probability theory. In Appendix \( A \) is a quick overview of probability theory used in this thesis.

2.1 Elasticity

In the previous section the end to end vector is defined as,

\[
R = \sum_{i\in\{1,\ldots,N\}} r_i
\]

Imagine the polymer to be stretched completely so that \( R = Nl \), and then is put into a heat bath. Because of the interactions with the heat bath, the configuration (conformation in biology and chemistry) of the polymer will be modified a bit. If a completely unfolded polymer is modified a bit, it will end up a bit shorter. In other words, there is an entropic force that contracts the polymer if its completely stretched. Also if \( R = 0 \), there will be a force that enlarges the molecule. Assuming this force, \( F(R) \), is continuous, there must be a point where there is no force at all.
Figure 3: Representations of a polymer. Created with a self-written computer program described in Section 3.
2.2 Ideal Chain Model

The ideal chain model describes a polymer in which there is only a force between the adjacent monomers. Therefore, in a polymer in thermal equilibrium, the vectors \( r_i \) are independent for all \( i \in \{1, ..., N\} \) and have no preference for certain angles and thus there is no preferred micro state. Let’s look at the components of the end to end vector \( R \).

\[
R_k = \sum_{i=1}^{N} r_{ki}, \quad k \in x, y, z
\]

These components are the sum of independent random variables. So if the mean, \( \langle r_{ki} \rangle \), and variance, \( \sigma_{ki}^2 \), are finite for all \( i \), the Central Limit Theorem \( \text{A.5} \) can be applied. This means the \( x, y, z \) components of \( R \) will be normally distributed if \( N \) is large enough. The next subsections are devoted to calculate the mean and variance of random vectors and to calculate a probability density function for \( R \). As stated earlier a random vector doesn’t have a preference for certain angles. In other words, this problem has a spherical symmetry. Because \( r \) has a fixed length, it is useful to transition into spherical coordinates.

\[
\begin{align*}
  r_x &= l \sin \theta \cos \phi \\
  r_y &= l \sin \theta \sin \phi \\
  r_z &= l \cos \theta
\end{align*}
\]

With \( \phi \in [0, 2\pi) \) and \( \theta \in [0, \pi] \). With surface element: \( dA = l^2 \sin \theta d\theta d\phi \).

2.2.1 Probability Density Function of a Random Vector

To calculate the probability density function \( f_r \) of a random vector \( r \) it is important to know the domain of this function. Since the vector has fixed length \( l \) the domain of \( f_r \) is the sphere with radius \( l, lS^2 \). If two surfaces are the same size the chance that they are hit by a random vector is the same. For this reason the surface element \( dA = l^2 \sin \theta d\theta d\phi \) is used in \( f_r \),

\[
P_r(A) = \int_{A} \frac{l^2 \sin \theta}{4\pi l^2} d\theta d\phi \implies P_r(lS^2) = 1
\]

Where \( P_r(A) \) is the chance a random vector hits the surface \( A \). This makes the probability density function,

\[
f_r(r) = \frac{\sin \theta}{4\pi}
\]
2.2.2 Expectation Value of a Random Vector

Due to the spherical symmetry $\langle r_x \rangle = \langle r_y \rangle = \langle r_z \rangle$. Because $r_z = l \cos \theta$ doesn’t depend on $\phi$, $\langle r_z \rangle$ is the least difficult to work out. Calculating the mean, Eq. (6), is easy now,

$$
\langle r_z \rangle = \int_{S^2} r_z \cdot f(x) \cdot d\theta d\phi
\Rightarrow \int_{S^2} l \cos \theta \cdot \frac{\sin \theta}{4\pi} \cdot d\theta d\phi
= \frac{l}{8\pi} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \sin 2\theta \cdot d\theta d\phi
= \frac{l}{4} \int_{0}^{\pi} \sin 2\theta \cdot d\theta = 0
$$

Not very surprising for a spherical symmetric problem.

2.2.3 Variance of a Random Vector

Now for a more interesting value, the variance, Eq. (8),

$$
\sigma^2_z = \langle r_z^2 \rangle - \langle r_z \rangle^2 = \langle r_z^2 \rangle =
\int_{S^2} l^2 \cos^2 \theta \cdot \frac{\sin \theta}{4\pi} \cdot d\theta d\phi
= \frac{l^2}{2} \int_{0}^{\pi} d\theta \cdot \frac{1}{4} (\sin 3\theta + \sin \theta)
= \frac{1}{3} l^2
$$

This result is obtained by writing out the trigonometric functions into exponents.

2.2.4 Central Limit Theorem in Random Vectors

Because the spherical symmetry,

$$
\langle r_x \rangle = \langle r_y \rangle = \langle r_z \rangle = 0
\Rightarrow \sigma^2_x = \sigma^2_y = \sigma^2_z = \frac{1}{3} l^2
$$
It turns out that both the expectation value and variance are indeed finite, so we can apply the Central Limit Theorem on,

\[ R_k = \sum_{i=1}^{N} r_k \quad \text{with} \quad k \in x, y, z \]

The theorem states that the probability density function \( f_{R_k} \) of \( R_k \), is normally distributed with mean \( \mu_k = N \cdot \langle r_k \rangle = 0 \) and variance \( \sigma_k^2 = \frac{N l^2}{3} \).

So,

\[
f_{R_k} = \left( \frac{1}{\sqrt{2\pi \frac{N l^2}{3}}} \right) e^{-\frac{3\sigma_k^2}{2N l^2}}
\]

2.2.5 The Probability Density Function of the End to End Vector

Now all parts are ready to be assembled into the probability density function of \( R \). The domain of \( f_R(R) \) is of course \( \mathbb{R}^3 \). Let’s define a volume \( V = \Delta X \times \Delta Y \times \Delta Z \) where \( \Delta X, \Delta Y \) and \( \Delta Z \) are intervals. The probability of finding \( R \) in \( V \) is equal to the probability of finding \( R_x \) in \( \Delta X \) and \( R_y \) in \( \Delta Y \) and \( R_z \) in \( \Delta Z \),

\[
P_R(V) = P_{R_x}(\Delta X) \cdot P_{R_y}(\Delta Y) \cdot P_{R_z}(\Delta Z) \Rightarrow
\]

\[
\int_V f_R(R)\,dv = \int_{\Delta X} f_{R_x}(x)\,dx \cdot \int_{\Delta Y} f_{R_y}(y)\,dy \cdot \int_{\Delta Z} f_{R_z}(z)\,dz
\]

\[
= \int_{\Delta X} \int_{\Delta Y} \int_{\Delta Z} f_{R_x}(x) \cdot f_{R_y}(y) \cdot f_{R_z}(z) \cdot dx\,dy\,dz
\]

\[
= \int_V f_{R_x}(x) \cdot f_{R_y}(y) \cdot f_{R_z}(z) \cdot dv \Rightarrow
\]

\[
f_R(R) = f_{R_x}(R_x) \cdot f_{R_y}(R_y) \cdot f_{R_z}(R_z)
\]

\[
f_R(R) = \left( \frac{3}{2\pi N l^2} \right)^{\frac{3}{2}} e^{-\frac{3r^2}{2N l^2}}
\]

Notice that \( f_R \) only depends on the magnitude \( R \) of \( R \),

\[
f_R(R) = \left( \frac{3}{2\pi N l^2} \right)^{\frac{3}{2}} e^{-\frac{3r^2}{2N l^2}} = \left( \frac{1}{2\pi \sigma^2} \right)^{\frac{3}{2}} e^{-\frac{r^2}{2\sigma^2}} \tag{2}
\]
2.2.6 The Expectation Value of the End to End Vector Length

Finally it is possible to calculate the expectation value of $|R|$ by using $\sigma^2 = \frac{1}{3} N l^2$ in $f_R(R)$,

$$\langle |R| \rangle = \langle R \rangle = \int_{R^3} r \cdot f_R(r) \cdot dv$$

$$= \int_0^{2\pi} \int_0^\pi \int_0^\infty r f_R(r) \cdot r^2 \sin \theta d\theta d\phi dr$$

Using the derivative $rf_R(r) = -\sigma^2 f_R'(r)$, Eq. (10),

$$\langle R \rangle = -4\pi \sigma^2 \int_0^\infty r^2 f_R'(r) dr$$

Integration by parts,

$$\langle R \rangle = -4\pi \sigma^2 r^2 f_R(r) \bigg|_0^\infty + 4\pi \sigma^2 \int_0^\infty 2r f_R(r) dr$$

$$= 0 + 4\pi \sigma^2 \int_0^\infty 2r f_R(r) dr$$

And again using the derivative $f_R'(r)$,

$$\langle R \rangle = -8\pi \sigma^4 \int_0^\infty f_R'(r) dr$$

$$= -8\pi \sigma^4 f_R(r) \bigg|_0^\infty$$

$$= 2\pi \sigma^4 \left( \frac{3}{2\pi N l^2} \right)^{\frac{3}{2}}$$

$$= 2 \sqrt{\frac{2}{\pi}} \sigma$$

$$= \left( 2 \sqrt{\frac{2}{3\pi}} \right) \sqrt{N l} \approx 0.92 \sqrt{N l}$$

The result predicts that a polymer in a heat bath will shrink to $\left(2 \sqrt{\frac{2}{3\pi}} \right) \sqrt{N l}$ for large $N$, which is only a fraction of its stretched size $N l$.

Some sites on the Internet and other information sources use the root mean square, $\sqrt{\langle R^2 \rangle}$, to calculate the expectation value $\langle R \rangle$. The root mean square is however an approximation and doesn’t give the exact answer.

$$\sqrt{\langle R^2 \rangle} = \sqrt{\langle R_x^2 \rangle + \langle R_y^2 \rangle + \langle R_z^2 \rangle} = \sqrt{N l}$$
2.3 Elasticity Explained With the Ideal Chain Model

With the probability density function of the end to end vector in our grasp, we are but one step away to determine the dynamics of the ideal chain. Let’s take a look at the formula for entropic forces, Eq. [1]

\[ F_r(r) = \frac{dE(r)}{dr} = T \frac{dS(r)}{dr} \]

Using the definition for entropy \( S = k_b \log \Omega \), where \( k_b \) is the Boltzmann constant and \( \Omega \) the number of micro states,

\[ dS(r) = k_b \log \Omega(r + dr) - k_b \log \Omega(r) = k_b \log \frac{\Omega(r + dr)}{\Omega(r)} \]

Since this isn’t a discrete model the multiplicity, \( \Omega(x) \), is infinite for almost all of the cases. There is a way to deal with this, the probability density function is proportional to the multiplicity. So we may conclude,

\[ dS(r) = k_b \log \frac{\Omega(r + dr)}{\Omega(r)} = k_b \log \frac{f_R(r + dr)}{f_R(r)} \]

Now by substituting \( f_R(x) \),

\[ dS(r) = k_b \log \left( e^{-\frac{1}{2\sigma^2}(r^2+2rdr+dr-r^2)} \right) = k_b \log \exp \left( -\frac{2rdr}{2\sigma^2} \right) = -\frac{k_b}{\sigma^2} r dr \]

Finally the entropic force can be calculated,

\[ F_r(r) = T \frac{dS(r)}{dr} = -\frac{k_b T}{\sigma^2} r = -\frac{3k_b T}{Nl^2} r \]  
\[ (3) \]

And because of the spherical symmetry there are no angular terms,

\[ \mathbf{F}(r) = -\frac{3k_b T}{Nl^2} r \]

This is of course Hooke’s law, the formula for a harmonic oscillator where \( \frac{3k_b T}{Nl^2} \) is the force constant. This means that when a mass \( M \) is put at the end of a polymer, and the first monomer is in a fixed position, the polymer will go into a spring like motion with a frequency of,

\[ \omega = \sqrt{\frac{3k_b T}{Nl^2 M}} \]

Also note that the force is indeed proportional to the temperature, as seen in Eq. [1] on page 4.
3 Simulating the Ideal Chain

It would be great to see this force in action to see how the polymer reacts to a heat bath and how this leads to an oscillation, Eq. (3). Luckily this is possible with computer simulations. In this section a computational method is explained which is used to simulate an ideal chain. The simulation program is written in C++. OpenGL is the graphics library used to render the graphics of the ideal chain. The graphics don’t add anything to the measurement, but gives insight into the dynamics of the ideal chain, and makes it possible to check if the program runs naturally, so that there are no unrealistic events occurring in the chain. The rigid rods of fixed length between the monomers are replaced by springs with rest length \( l = 1 \), for it would be very difficult to program and computational intensive to use rigid rods.

To simulate the ideal chain, it has to be translated to a physical representation. Let

- \( \rho_i, v_i, a_i \) respectively be the absolute position, the velocity and the acceleration of the \( i \)th monomer with \( i \in 0, \ldots, N \);
- \( m \) be the mass of the monomers;
- \( M \) be the mass of the last monomer;
- \( k \) be the spring constant between the monomers.

Remember that \( N + 1 \) is the number of monomers. That’s why the range of \( i \) goes from 0 to \( N \). To calculate \( \rho_i \), the velocity needs to be calculated; to calculate \( v_i \) the acceleration needs to be calculated. The first monomer is stationary, and the last one has mass \( M \). For the others the acceleration only depends on the two springs that connect the monomer. Remember that the springs have stationary for length \( l = 1 \).

\[
\begin{align*}
a_0 &= 0 \\
a_N &= -\frac{k}{M}(r_{N-1} - \hat{r}_{N-1}) \\
a_i &= -\frac{k}{m}(r_{i-1} - \hat{r}_{i-1} + r_i - \hat{r}_i)
\end{align*}
\]

Where \( r_i = \rho_{i+1} - \rho_i \). And \( \hat{v} = \frac{v}{|v|} \) for every vector.

The basic algorithm for the simulation is to make small time steps, frames of length \( dt \), and update the positions of the monomers,

\[
r_i(t + dt) = r_i(t) + v_i(t)dt
\]
The velocities are also updated every frame by the acceleration. There is also a term added to simulate a heat bath. In a heat bath, energy is transported by colliding molecules. To simulate this, the velocities of the monomers are changed a bit per time step by a impulse $dp$.

$$v_i(t + dt) = v_i(t) + a_i(t)dt + \frac{dp}{m}$$

Where $dp$ is a random vector of length $dp$.

Take a look at Appendix A for a more detailed discussion about this algorithm.

### 3.1 Input and Output of the Simulation

With the use of *OpenGL* it is possible to render the Ideal Chain every frame, and thus create a short video of the simulation. A video is put up on YouTube at: [http://www.youtube.com/user/WiegerSteggerda](http://www.youtube.com/user/WiegerSteggerda).

The output of the program is a data plot, with the length of the end to end vector set out against time. If indeed the simulated ideal chain acts as predicted in Section 3, it will result in an sinus, but length can only be positive so it will be an absolute sinus.

The inputs parameters for the simulation are:

- $N$, the number of monomers;
- $m$, the mass of one monomer;
- $M$, the mass of the last monomer;
- $dt$, the duration of one time step;
- $k$, the spring constant between the monomers;
- $dp$, the impulses of the colliding atoms in the heat bath;
- $Q$, the ratio of impulses $dp$ per time step per monomer.

To get the most accurate results a large $N$, $k$; and small $dp$, $dt$ are required. The only problem is that these requirements add significantly to the computational time, so a good balance needs to be found.
3.2 The results

Let’s try to link the theory to the newly defined parameters for the simulation, so that a prediction can be made on how the ideal chain behaves under different variables/inputs. The theory states that the angular frequency is,

\[ \omega = \sqrt{\frac{3k_bT}{NL^2M}} \]

This means that when the temperature \( T \) is quadrupled, the frequency is doubled. When the mass \( M \) or the number of monomers \( N \) is quadrupled, the frequency is halved. The temperature isn’t really directly visible in the simulation, but there are variables that represent the heat bath, namely \( dp \) and \( Q \). Both \( k_bT \) and \( (\frac{dp}{2m})^2 \) have the dimension of energy, so they are probably proportional. This means that when \( dp \) is doubled, \( T \) is quadrupled, and thus the frequency also doubles.

The next three graphs have the number of frames passed, or time, on the x-axis, and the end to end distance ratio \( \frac{R}{NL} \) on the y-axis. The first graph, see Figure 4, tests the dependency of \( dp \). For the three plots \( dp_1 = 0.001 \), \( dp_2 = 0.002 \) and \( dp_3 = 0.004 \). The expected result was that the frequency doubled if \( dp \) doubled. Because two periods of the \( dp_2 \) plot fit into one period of \( dp_1 \); and two periods of the \( dp_3 \) plot fit into one period of \( dp_2 \), the frequency is indeed doubled.

The second graph, Figure 5, tests the effect of quadrupling \( M \). \( M_1 = 2500 \), \( M_2 = 10000 \) and \( M_3 = 40000 \), clearly the period of \( M_1 \) fits twice in the period of the \( M_2 \) case, and the period of \( M_2 \) fits twice in the period of \( M_3 \). So again the theory and the simulations agree.

In the third and last graph, Figure 6 the effect of quadrupling \( N \) is tested. \( N_1 = 75 \), \( N_2 = 300 \) and \( N_3 = 1200 \). And again the predictions hold. The frequency is doubled, if \( N \) is multiplied by four.

So in all of the cases, the simulation verifies the theory with a good precision.
4 Conclusion and Discussion

There is of course a very big difference between the simulated model and the theoretical model. The rigid rods of the theoretical model are replaced by springs. One might argue that it is not very surprising to get an oscillation from a lot of connected springs, and that we must not speak of an entropic force, because the force is put into the model directly. If this were the case, there would be no dependence on the temperature \( dp \) which there is. There are also measurements made to verify that the spring constant \( k \), of the springs between the monomers, has no influence of the results. There is some difference because the springs store energy from the heat bath, and this energy is different for different spring constants. If \( k \) is a hundred times smaller, the frequency gets only about 3.4 times bigger. So the effect of \( k \) is very small compared to the other results. For even smaller \( k \) the ideal chain becomes very elastic and unrealistic. For bigger \( k \) the program bugs out, because the forces on the monomers get too big. Combined with a discontinuity in time, this will result in enormous, unrealistic translations. Therefore \( k \) is chosen to be as high as possible for all the measurements, for the most realistic ideal chain simulation.

There are still a few interesting things that could be researched but were simply not possible in the short time span of a bachelor thesis. For example: When will the oscillation die out, and how long does it take? And when is \( N \) too small to give accurate results? It would also be nice to retrieve a formula for the frequency in which all the input variables of the program are used. Also it is not very accurate to measure the frequencies by hand and makes it impossible to get a good margin of error. Though it looks like the result are very accurate.

It comes as a surprise that the plots are so incredibly smooth. One might forget that there are actually \( N \) monomers busy with pushing and pulling the chain at every frame. And this is the point this project has been all about. In nature, the fundamental forces are seen as indivisible, because they are so accurate and consistent. This project showed that from entropic principles, very smooth forces can be created, especially for large \( N \). This might be a clue that there is more under the engine cowl of nature.
Figure 4: Three plots of $\frac{|R|}{N_1}$ as function of time for different values of $dp$.
($N = 1000, M = 10000, Q = 0.1, k = 0.01, dp = 0.001$)
Figure 5: Three plots of $|R_N|$ as function of time for different values of $M$.

($N = 1000, Q = 0.1, k = 0.01, dp = 0.001$)
Figure 6: Three plots of $|\mathcal{N}|$ as a function of time for different values of $N$.

($N = 1200$, $N = 300$, $N = 75$)
Acknowledgements

A huge thanks to Erik Verlinde for his insight and for welcoming the idea to use a computer in this thesis. I have enjoyed this project a lot. And because the subject wasn’t pinned down from the start, it gave me the freedom to research a lot of different aspects of entropic forces and also throwing my own ideas in once in a while. And despite being a busy man, Erik always could sit down to take the time and help me with my progress. I wish him bests of luck in the further development of his theory.

References

A Mathematical Probabilities

A.1 Probability Density Function

The Probability Density Function $f(x)$ is useful when dealing with continuous random variables. It is used to calculate the probability to measure the random variable in a certain interval, $[a, b]$.

$$P([a, b]) = \int_a^b f(x) \, dx$$

(4)

where we integrate over the possible results of a random experiment and $P([a, b])$ denotes the probability this experiment results in a value between $a$ and $b$.

This function doesn’t have to be bound by only one variable. For example, measuring the position of a particle in three dimensions results in three numbers. The next function gives the probability of measuring a random vector in the volume $V$,

$$P(V) = \int_V f(r) \, dv$$

(5)

The most used probability density function in physics is probably the squared absolute wave function $|\Psi|^2$ in quantum mechanics.

Integrating over the whole space results, of course, in a probability of 1,

$$P([-\infty, \infty]) = 1$$

$$P(\text{All space}) = 1$$

A.2 Expectation Value

In a probability density function the expectation value or mean $\langle g(x) \rangle$ of $g(x)$ is,

$$\langle g(x) \rangle := \int_{-\infty}^{\infty} g(x) \cdot f(x) \, dx$$

(6)

where $f(x)$ is a probability density function.

A.3 Variance

The variance $\sigma_x^2$ is defined as,

$$\sigma_x^2 = \langle x - \langle x \rangle^2 \rangle$$

(7)
It depends on the spread of a probability density function around the expectation value. This can be derived further,

\[ \sigma_x^2 = \langle (x - \langle x \rangle)^2 \rangle \]
\[ = \langle x^2 + \langle x \rangle^2 - 2x \langle x \rangle \rangle \]
\[ = \langle x^2 \rangle + \langle x \rangle^2 - 2\langle x \rangle \langle x \rangle \]
\[ = \langle x^2 \rangle - \langle x \rangle^2 \] (8)

This formula is a very useful tool to calculate the variance of probability density functions.

**A.4 Normal Distribution**

A Normal Distribution is a special probability density function.

\[ f(x) = \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right) e^{-\frac{x^2}{2\sigma^2}} \] (9)

Figure 7: Normal distribution with \( \sigma_1 = 1.0 \) and \( \sigma_2 = 0.5 \)
A.4.1 The derivative of the normal distribution

It’s not very difficult to determine the derivative of this function,

\[
f'(x) = -\left(\frac{1}{\sigma^3\sqrt{2\pi}}\right) \cdot xe^{-\frac{x^2}{2\sigma^2}}
\]

(10)

A.4.2 Integrating the normal distribution

Determining the primitive of the normal distribution is not possible. However, it is possible to evaluate the surface under the function with the use of double integrals,

\[
\left(\int_{-\infty}^{\infty} e^{-\frac{x^2}{2\sigma^2}} \, dx\right)^2 = \left(\int_{-\infty}^{\infty} e^{-\frac{x^2}{2\sigma^2}} \, dx\right) \cdot \left(\int_{-\infty}^{\infty} e^{-\frac{y^2}{2\sigma^2}} \, dy\right)
\]

\[
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{x^2+y^2}{2\sigma^2}} \, dx \, dy
\]

\[
= \int_{\mathbb{R}^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \, dA
\]

Making the transition to polar coordinates,

\[
\int_{\mathbb{R}^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \, dA = \int_0^{2\pi} \int_0^{\infty} e^{-\frac{r^2}{2\sigma^2}} \cdot r \, dr \, d\theta
\]

Substituting \( s = \frac{r^2}{2\sigma^2} \) \( \iff r \, dr = \sigma^2 \, ds \) results in,

\[
\int_0^{2\pi} \int_0^{\infty} e^{-\frac{r^2}{2\sigma^2}} \cdot r \, dr \, d\theta = 2\pi \int_0^{\infty} \sigma^2 e^{-s} \, ds
\]

\[
= 2\pi \sigma^2 \left[ -e^{-s} \right]_0^{\infty}
\]

\[
= 2\pi \sigma^2
\]

And thus,

\[
\int_{-\infty}^{\infty} e^{-\frac{x^2}{2\sigma^2}} \, dx = \sqrt{2\pi\sigma^2}
\]

(11)

This means that the surface under the normal distribution is:

\[
\int_{-\infty}^{\infty} f(x) = \int_{-\infty}^{\infty} \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) \cdot e^{-\frac{x^2}{2\sigma^2}} = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) \cdot \sqrt{2\pi\sigma^2} = 1
\]

(12)

The area under \( f(x) \) equals 1, so the normal distribution is a well defined probability density function.
A.4.3 The expectation value of the normal distribution

Applying the definition of the expectation value gives:

$$\langle x \rangle := \int_{-\infty}^{\infty} x \cdot f(x) \, dx = 0$$

Because \( x \cdot f(x) \) is anti-symmetric around zero and finite everywhere, integrating from \(-\infty\) to \(\infty\) gives 0.

A.4.4 The variance of the normal distribution

Since \( \langle x \rangle = 0 \) the variance is equal to,

$$\sigma_x^2 = \langle x^2 \rangle - \langle x \rangle^2 = \langle x^2 \rangle$$

Using the definition of the expectation value,

$$\langle x^2 \rangle = \int_{-\infty}^{\infty} x^2 f(x) \, dx$$

This can be rewritten in a form containing the derivative of \( f(x) \), Eq. (5),

$$\langle x^2 \rangle = -\sigma^2 \int_{-\infty}^{\infty} x f'(x) \, dx$$

Integration by parts gives,

$$\langle x^2 \rangle = -\sigma^2 x f(x) \bigg|_{-\infty}^{\infty} + \sigma^2 \int_{-\infty}^{\infty} f(x) \, dx$$

Integrating over a probability density function results in an area of 1, Eq. (7),

$$\langle x^2 \rangle = -\sigma^2 x f(x) \bigg|_{-\infty}^{\infty} + \sigma^2$$

And finally, to calculate the first term,

$$-\sigma^2 x f(x) \bigg|_{-\infty}^{\infty} = \left( \frac{-\sigma^2}{\sqrt{2\pi \sigma^2}} \right) \cdot xe^{-\frac{x^2}{2\sigma^2}} \bigg|_{-\infty}^{\infty}$$

$$= 0$$

This concludes the calculation,

$$\sigma_x^2 = \langle x^2 \rangle = \sigma^2$$

Obviously, the letter \( \sigma \) in the normal distribution was neatly chosen from the beginning to match the variance \( \sigma_x^2 \).
A.5 Central Limit Theorem

The Central Limit Theorem is a very important theorem in statistics. If $X_1, X_2, \ldots, X_n$ are independent random distributed variables with finite mean $\mu_i$ and finite variance $\sigma_i^2$ and large enough $n$, the sum $S_n = \sum_{i=1}^{n} X_i$ is normally distributed with mean $\mu = \sum_{i=1}^{n} \mu_i$ and variance $\sigma^2 = \sum_{i=1}^{n} \sigma_i^2$.

Proofs of this theorem can be found on the web, in statistics books and syllabi\textsuperscript{[3,4]}.

Let’s put this theory to the test with the use of Mathematica. Four data sets of size 100000 are created with $n_1 = 1$, $n_2 = 2$, $n_3 = 8$ and $n_4 = 16$. Each element in these sets is the sum of $N$ random by a quadratic distribution function $f(x)$. Where

$$f(x) = \begin{cases} \frac{3}{2}x^2 & \text{if } x \in [-1, 1] \\ 0 & \text{otherwise} \end{cases}$$

To create a distribution in Mathematica a random number $R \in [0, 1]$ is picked. $R$ represents the following probability,

$$R = P([-1, x]) = \int_{-1}^{x} f(y)dy = \frac{1}{2}x^3 + \frac{1}{2}$$

$$x = \sqrt[3]{2R-1}$$

The random measurement $\sqrt[3]{2R-1}$ is now distributed by $f(x)$.

The mathematical code for this measurement:

Z[] := 2*Random[] - 1
X[] := Sign[Z[]] * (Abs[Z[]]^(1/3))
(* X[] is actually just the real part of (Z[])^(1/3) *)

(* To sum n measurements *)
S[n_] := Sum[X[], {i, n}]

The mean of these random numbers is $\mu_i = \langle x \rangle = 0$, and the variance is $\sigma_i^2 = \langle x^2 \rangle - \langle x \rangle^2 = \langle x^2 \rangle = \int_{-1}^{1} \frac{3}{2}x^4dx = \frac{3}{10} + \frac{3}{10} = \frac{3}{5}$. The central limit theorem states that these sums are normally distributed with a mean $\mu = \sum_{i=1}^{n} \mu_i = 0$ and a variance $\sigma^2 = \sum_{i=1}^{n} \sigma_i^2 = \sigma_i^2 \cdot n = \frac{3n}{5}$.
Let’s find out whether the central limit theorem holds for this example. Mathematica is used to create data sets of $S[n]$ for different values of $n$. After that, the variance is calculated, and histograms are made from the data sets.

data1 = Table[S[1], {100000}];
data2 = Table[S[2], {100000}];
data3 = Table[S[8], {100000}];
data4 = Table[S[16], {100000}];

Variance[data1] / 1
Variance[data2] / 2
Variance[data3] / 8
Variance[data4] / 16

Histogram[data1]
Histogram[data2]
Histogram[data3]
Histogram[data4]

The results for the variance are,

\[
\sigma_{data1}^2/1 = 0.600233 \\
\sigma_{data2}^2/2 = 0.599631 \\
\sigma_{data3}^2/8 = 0.600507 \\
\sigma_{data4}^2/16 = 0.600865
\]

So the variance stays close to the expected value from the central limit theorem: $\sigma^2 = 0.6n$.

The Histograms are listed in Figure 8. Indeed the higher $n$ the more it resembles a normal distribution.

So even if the probability density function is completely different from a normal distribution, the sum of only a few measurements becomes normally distributed.
Figure 8: The histograms of data1, data2, data3 and data4 converge to a normally distributed function

B Pseudo Code For Simulation of the Ideal Chain

Pseudo code is not code for a program to read, it is code so every human with a basic understanding of programming can understand it. It is therefore not in a specific language, but it should be familiar if the reader has programmed in an (imperative) programming language before. If you have never programmed before you could understand it as well. It is important to know that the equals sign, "=", has to be read as becomes. For example, the expression \( a = a + 1 \) is a common thing to state in a programming language, and has to be read as, "a becomes a plus one", so that the new value for a, is one higher than before.

B.1 Creating a Random Vector

Creating a random unit vector (spherical symmetrical), is less straightforward as it might seem. Simply creating two random angles \( \theta \) and \( \phi \) is not symmetric because the probability density function will be denser at the poles. The following method is simplistic, but it works:

\[
\text{DO} \\
\quad \text{-- create random vector in } [-1,1]^3
\]
\[ v.x = (\text{rand()} \times 2.0) - 1.0 \]
\[ v.y = (\text{rand()} \times 2.0) - 1.0 \]
\[ v.z = (\text{rand()} \times 2.0) - 1.0 \]

WHILE |v| > 1.0 -- accept vector if it is inside the unit sphere

-- normalize the vector
v = v / |v|

Where \text{rand()} returns a random value between zero and one and \( v \) is a vector.

**B.2 The Basic Algorithm**

In Section [3] the basic algorithm is already mentioned. Let

- \( \hat{a}[i] \), \( v[i] \) and \( \rho[i] \) be the acceleration, velocity and the position of the \( i \)'th monomer respectively;
- \( N \) be the number of monomers;
- \( k \) be the spring constant between the monomers;
- \( m \) be the mass of each monomer;
- \( M \) be the mass of the last monomer;
- \( \text{dp()} \) return a random vector of length \( \text{dp} \);
- \( \hat{r}(i) := \rho[i+1] - \rho[i] \).

Here it is in pseudo code for the simulation,

FOR EACH frame do

-- calculate acceleration loop
\[
a[0] = 0
\]
\[
\text{FOR } i = 1 \text{ TO } N-1
\]
\[
a[i] = -k/m \times (r[i-1] - \text{norm}(r[i-1])) + k/m \times (r[i] - \text{norm}(r[i]))
\]
\[
\text{END FOR}
\]
\[
a[N] = -k/M \times (r[N-1] - \text{norm}(r[N-1]))
\]

-- calculate velocity loop
FOR i = 0 TO N
\[ v[i] = v[i] + a[i] \times dt \]

-- collide with an atom every once in a while
IF rand() < Q
    \[ v[i] = v[i] + dp()/m \]
END FOR

-- calculate position loop
FOR i = 0 TO N
    \[ \rho[i] = \rho[i] + v[i] \times dt \]
END FOR
END FOR EACH

### B.3 Speeding Up the Basic Algorithm

The algorithm in the previous section is short and understandable, which is good. It preforms badly on a computer however, because there are for loops over \( a, v \) and \( \rho \). It turns out that the acceleration doesn’t really need to be remembered for it is changed completely for every frame. Therefore it is possible to calculate the acceleration in the velocity loop. This takes care of one of the loops.

FOR EACH frame do
    -- calculate velocity loop
    \[ v[0] = 0 \]
    FOR i = 1 TO N-1
        \[ a = -k/m \times (r(i-1) - \text{norm}(r(i-1))) + k/m \times (r(i) - \text{norm}(r(i))) \]
        \[ v[i] = v[i] + a \times dt \]
        -- collide with an atom every once in a while
        IF rand() < Q
            \[ v[i] = v[i] + dp()/m \]
        END FOR
    \[ a = -k/M \times (r(N-1) - \text{norm}(r(N-1))) \]
    \[ v[N] = v[i] + a \times dt \]

-- calculate position loop
FOR i = 0 TO N
\[
\rho[i] = \rho[i] + v[i] \times dt
\]
END FOR
END FOR EACH

Combining the velocity loop with the position loop is a little bit harder. It is not possible to just put them together in one loop, like this:

FOR EACH frame do
  \[v[0] = 0\]
  FOR \(i = 1\) TO \(N-1\)
    \[a = -\frac{k}{m} \times (r[i-1] - \text{norm}(r[i-1])) + \frac{k}{m} \times (r[i] - \text{norm}(r[i]))\]
    \[v[i] = v[i] + a \times dt\]

    -- collide with an atom every once in a while
    IF rand() < Q
      \[v[i] = v[i] + dp() / m\]
    END IF
  END FOR
  \[a = -\frac{k}{M} \times (r[N-1] - \text{norm}(r[N-1]))\]
  \[v[N] = v[N] + a \times dt\]
  \[\rho[N] = \rho[N] + v[N] \times dt\]
END FOR EACH

When calculating the acceleration, the information \(r(i)\) is used. This memory is still waiting to be updated in the last line of the code. This might not seem too bad, it is however, a very serious problem. First \(r(1)\) will be calculated incorrectly, this will have an added effect on \(r(2)\), which will add extra error to \(r(3)\). So it creates a chain reaction of small errors, and the chain will end up in limbo.

The solution to this problem is to calculate the first value of the velocity, and after that start to calculate the positions \(r(i)\) simultaneously with \(v[i+1]\). In this case the velocities will be updated with the old positions, so that the new positions can be calculated from the just acquired velocities:

FOR EACH frame do
  \[v[0] = 0\]
  \[a = -\frac{k}{m} \times (r(0) - \text{norm}(r(0))) + \frac{k}{m} \times (r(1) - \text{norm}(r(1)))\]
  \[v[1] = v[1] + a \times dt\]
FOR $i = 2$ TO $N-1$
\begin{align*}
a &= -k/m \times (r(i-1) - \text{norm}(r(i-1))) + \\
    &\qquad k/m \times (r(i) - \text{norm}(r(i))) \\
v[i] &= v[i] + a \times dt
\end{align*}

-- collide with an atom every once in a while
IF rand() < $Q$
\begin{align*}
v[i] &= v[i] + dp() / m
\end{align*}

\begin{align*}
\rho[i-1] &= \rho[i-1] + v[i-1] \times dt
\end{align*}
END FOR

\begin{align*}
a &= -k/M \times (r(N-1) - \text{norm}(r(N-1))) \\
v[N] &= v[N] + a \times dt
\end{align*}

\begin{align*}
\rho[N-1] &= \rho[N-1] + v[N-1] \times dt \\
\rho[N] &= \rho[N] + v[N] \times dt
\end{align*}
END FOR EACH

It is hideous, hard to read, code, but it runs like a beast. Three times faster then the first naïve algorithm. The program that was used to calculate the results was running for 45 minutes just to retrieve one plot. If it wasn’t for the improved algorithm this would be 2 hours and 15 minutes. That would be unbearable because measurements don’t always go right the first time.