Analysis of Homopolymer Melts Using SOMA (SOft coarse grained Monte-Carlo Acceleration)

UBC

Physics &
Astronomy

Matthew Sahota
University of British Columbia
MATH 499

Table of Contents

Letter of Transmittal

To whom it may concern,

This report was prepared for Joerg Rottler, a Professor of Physics and Astronomy at the University of British Columbia. It covers the research performed under Professor Rottler's supervision during my co-operative work experience term in the summer of 2022.

This report is to be used as reference for any future employees or students that may use or modify the work that I had completed during my term under the supervision of Professor Joerg Rottler. The content in this report is non-confidential.

The focus of this report is on the usage of SOMA on the LISA cluster provided by the Stuart Blusson Quantum Matter Institute for homopolymer melts in a discretized domain with and without the application of an external field, which includes thermal and electromagnetic fields. This report also includes the change in density with or without fields over time, as well as the analysis of the positions of sections and the centres of mass of molecules, both of which were done in Python.

At the end of this report are suggestions and inquiries into future work for any students and employees who are willing to work under the supervision of Professor Joerg Rottler.

Sincerely,

Matthew

UBC Science Co-op

Analysis of Homopolymer Melts Using SOMA (SOft coarse grained Monte-Carlo Acceleration)

Matthew Sahota
The University of British Columbia
MATH 499

Introduction and Background

SOMA (SOft coarse grained Monte-Carlo Acceleration) is a numerical software for simulating polymer melts with multiple, changeable conditions, such as the number of types of molecules, domain shape, the number of Monte-Carlo time steps before a file containing the positions of all molecules in the system is created, and the interaction coefficient between different types of molecules (Schneider and Müller, 2019, Daoulas and Müller, 2006). SOMA utilizes Monte-Carlo methods to simulate random movements in the given domain for a certain number of Monte-Carlo timesteps; fields can also be applied to the system to simulate thermal or electromagnetic activity. Simulations using SOMA were run on CPUs and GPUs that were available on the LISA computing cluster provided by the Stuart Blusson Quantum Matter Institute. Using the LISA computing cluster, as well as using Jupyter Notebook on my own computer, I was able to run programs to create external fields and analyze the behaviour of the molecules over time, with and without the field being applied.

Some characteristics that were analyzed were the density of the molecules along one direction, the formation of Rouse modes and their coordinates, and the effect of averaging multiple runs and analyzing how it affected the distribution of Rouse modes. Rouse modes are sections of a molecule of N chains and $N+1$ atoms that involve adding the positions of the individual atom in the molecule and applying a cosine transform of period p; this is called finding the p^{th} Rouse mode (Padding, 2005). The p^{th} Rouse mode defines the motion for a section of the molecule that contains N/p chains. When $p=0$, this gives the centre of mass of the molecule (Padding, 2005).

The distributions that result from finding the Rouse modes with no field for $p \geq 1$ are Gaussian distributions with decreasing variances as p increases, while the distribution for the centre of mass ($p = 0$) is a uniform distribution (Padding, 2005, Müller, 2022).

Experimental Design

For our simulations, we used one million molecules of one type with one hundred chains in a box centred around the origin ($[0, 0, 0]$) with vertices $[\pm 0.5, \pm 0.5, \pm 0.5]$, and we set the amount of time before the creation of files with the positions of the atoms to 100 Monte-Carlo steps, and the number of steps in space that we used was forty steps in all directions (x, y, and z). External fields were cosine functions with the equation $h \cdot \cos(q\pi x/L)$, where $q = 4$, $L = 1$, and h is the amplitude, which could be changed between runs. Notice that the field only runs in the x-direction; the y- and z-directions are not affected by the field. Python scripts were used to execute special functions on the LISA computing cluster for analyzing the data from the simulations; downloading the files containing all the positions of the atoms took up gigabytes of space on my home computer, and it would also take hours to download all the files created for an entire simulation. Simulations were run with both the field on and off with varying numbers of Monte-Carlo steps; 3000 Monte-Carlo steps were used as the benchmark for our simulations since it was the Rouse time τ_R for the simulations (the time it takes for the chains for $p = 1$ to relax); therefore, most runs were executed for either 3000, 6000, or 9000 Monte-Carlo steps (Padding, 2005).

Discussion Part I: Density Profiles

One characteristic that was analyzed in our research was the density of molecules over time along the x-direction. At rest, the density is mostly uniform, but with the application of an external field, the density takes on a different shape; removing the external field causes the density to relax back to its originally uniform shape. The way we analyzed the density of the system was to create two sums; one that added all the densities along the y- and z-directions and divided by the number of steps in the y-direction multiplied by the number of steps in the z-direction, and another that added all densities in all directions divided by the product of the number of steps in all directions; the first sum was divided by the second sum to create the relative density. The first sum remained untouched along the x-direction, which allowed us to create animated plots displaying the formation of the density profile over time.

From there, we fit a sinusoidal curve with the equation $f(x,t) = 1 + a(t) \cdot \cos(4\pi x)$ to our density profile and found our amplitude $a(t)$ through the usage of a curve fitting function provided by SciPy (Müller, 2022). We monitored the formation of our amplitude by using this curve fitting function every ten Monte-Carlo steps. For comparison between different amplitudes and between simulations with and without the external field, we divided the amplitude plots by the maximum amplitude and plotted the field on and field off example next to each other. Below are some examples of the adjusted amplitude plots and the comparison between simulations with and without an external field. Notice that the difference between field off and field on simulations becomes more obvious as the amplitude increases.

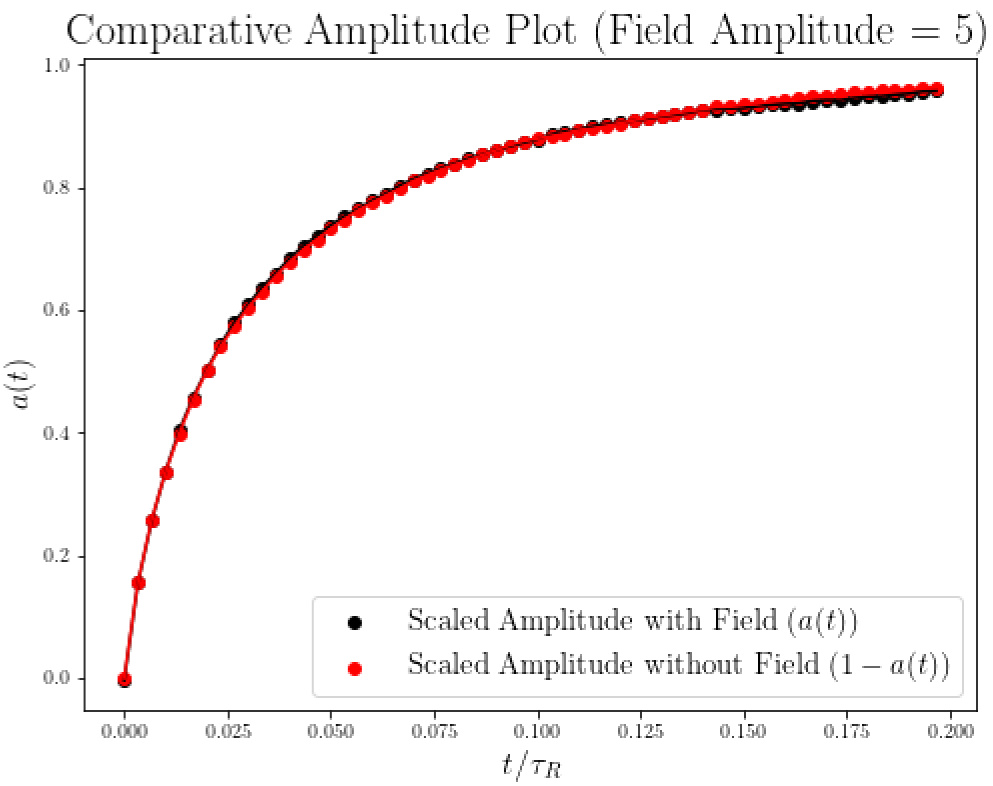

Figure 1: Simulation with $h = 5$, the maximum amplitude of the unadjusted plot was approximately 0.35. Recall that the Rouse time τ_R was approximately 3000.

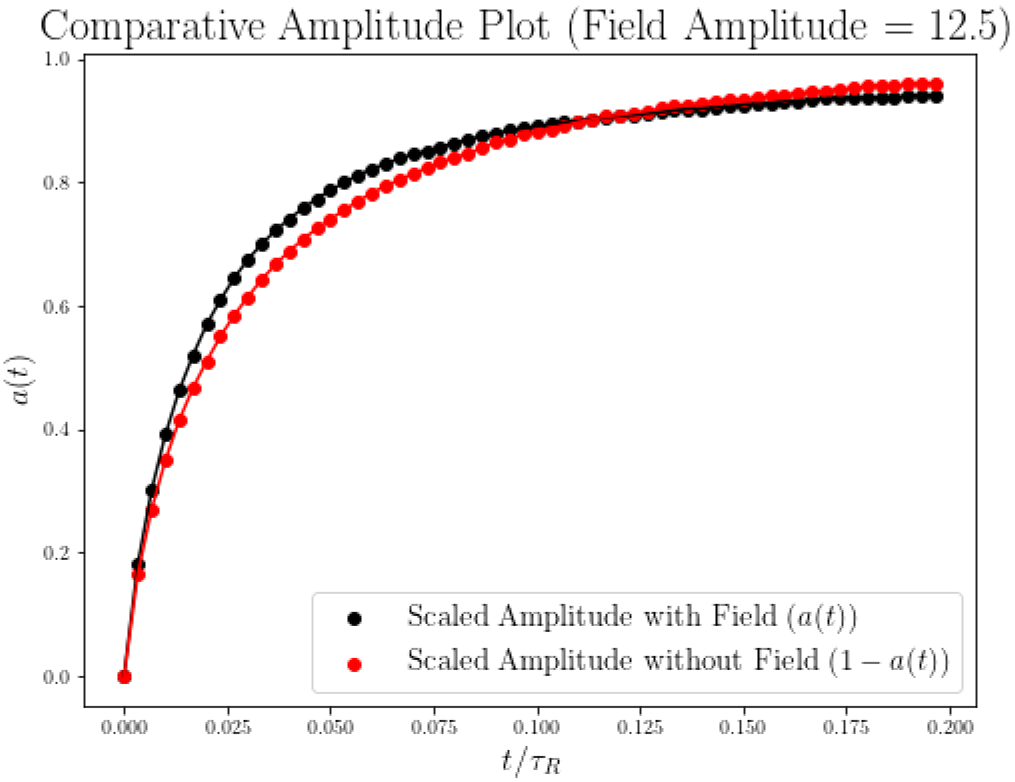

Figure 2: Simulation with $h = 12.5$, the maximum amplitude of the unadjusted plot was approximately 0.76.

Discussion Part II: Rouse Mode Analyses

Another characteristic we analyzed was the sections of our molecules called the Rouse modes; recall that the p^{th} Rouse mode for a molecule of length N gave the position for a section of the molecule of N/p length. The positions for the Rouse modes were given by,

$\mathbf{X}_p = \frac{1}{N+1}\sum_{n=0}^{N}\mathbf{R}_n \cos\left(\frac{p\pi}{N+1}(n+0.5)\right)$, where $\mathbf{R}_n$ is the position of the atoms of the molecule (Padding, 2005). All atoms that left our domain were brought back into the box, and this was carried out at every time step. For our simulations, we set our range of p-values from 0 to 3, and we found the Rouse modes for both the field on and field off simulations. Once the Rouse modes were found at each time step, we analyzed the distribution of the Rouse modes and created animated histograms displaying the distributions and plotted the maximum values of the distributions at each timestep. We also ran multiple simulations and averaged them out to reduce the deviation of the maximum values of the Rouse mode distributions, and we also performed rolling averages to accomplish this same goal. We found that running multiple simulations helped reduce the deviation in the Rouse mode distributions, and applying a rolling average alongside averaging multiple runs worked even better in reducing unwanted noise, but for higher modes, say p = 2 and 3, there would be a dip in the maximum value with the field on; for lower field strengths, the maximum values would never recover to a value equal to or higher than the initial value, but for higher field strengths, the maximum values would increase after dipping. The maximum values were shifted in a manner similar to the density profiles in the previous section. Below are some examples of maximum Rouse mode distributions with shifting.

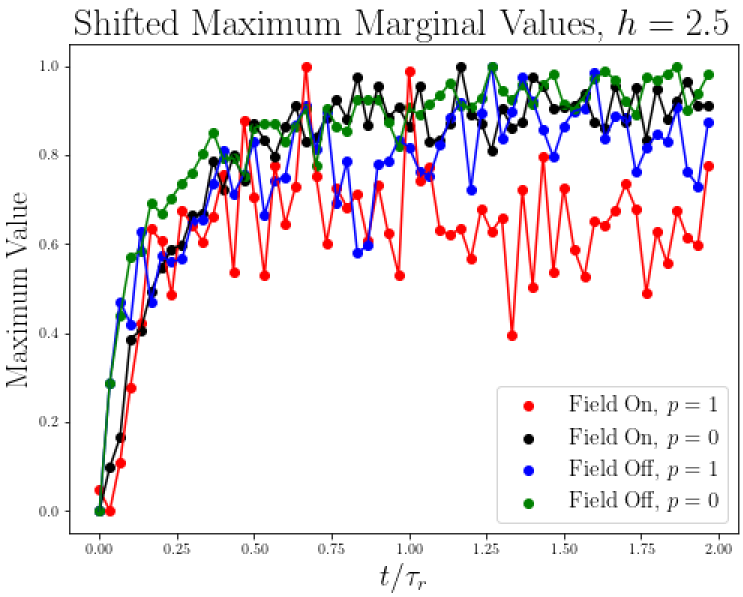

Figure 3: Shifted maximum marginal values for a field amplitude of 2.5 for p = 0, 1.

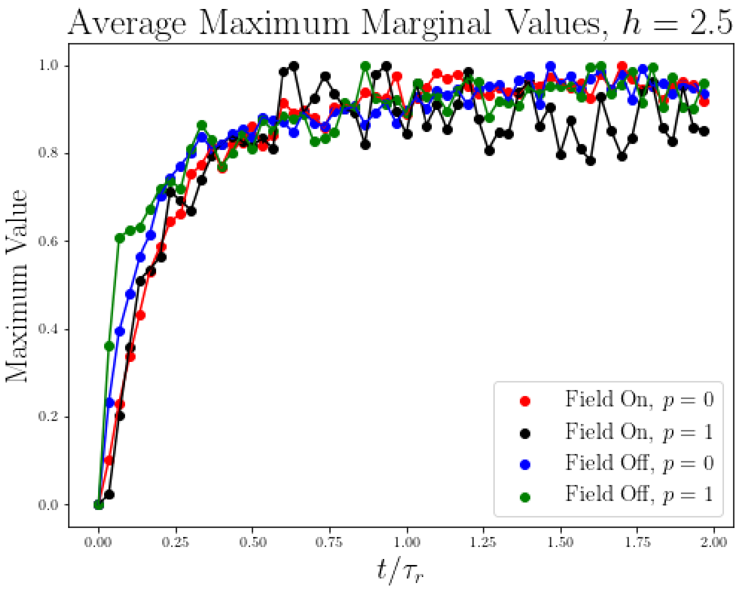

Figure 4: Shifted maximum marginal values for a field amplitude of 2.5 for p = 0, 1, averaged for five runs.

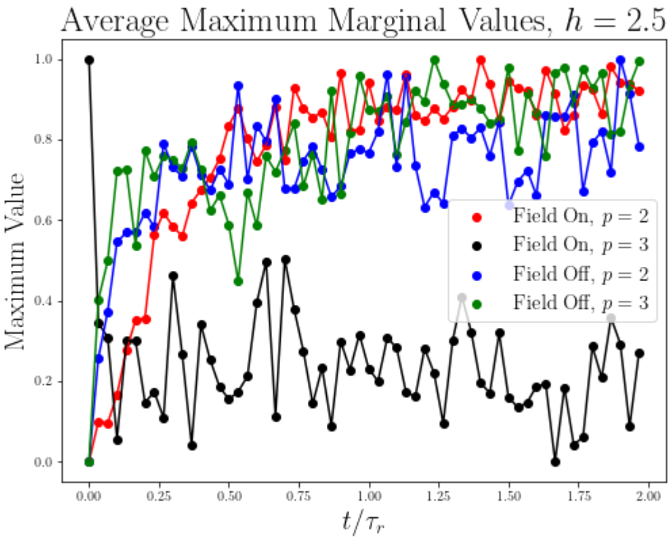

Figure 5: Shifted maximum marginal values for a field amplitude of 2.5 for p = 2, 3, averaged for five runs. Notice that the field on curve for p = 3 has a different shape.

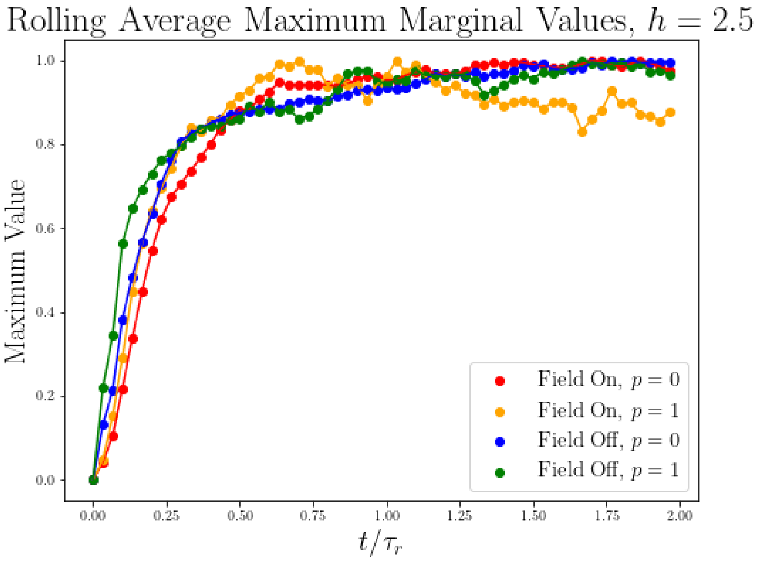

Figure 6: Shifted maximum marginal values for a field amplitude of 2.5 for p = 0, 1, averaged for ten runs and with a rolling average of spacing 3 applied.

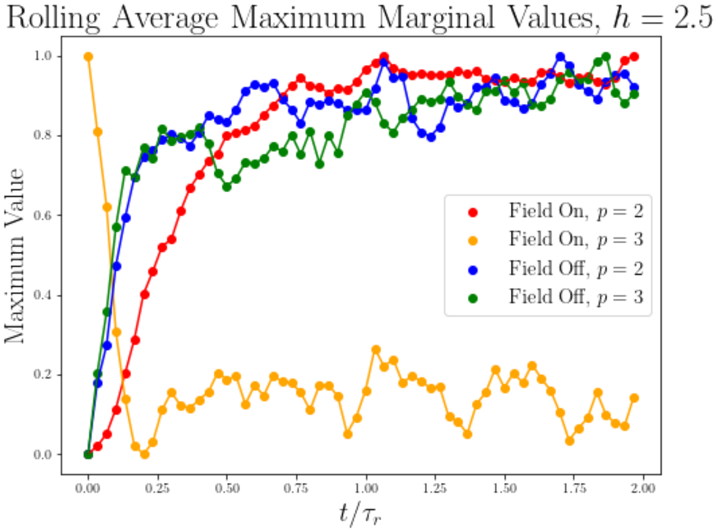

Figure 7: Shifted maximum marginal values for a field amplitude of 2.5 for p = 2, 3, averaged for five runs and with a rolling average of spacing 3 applied. Notice that the field on curve for p = 3 has a different shape.

Conclusion

For the four months spent under Professor Joerg Rottler's supervision, I have run homopolymer melt simulations with external fields and have modelled properties such as particle density, Rouse mode distributions, and smoothing methods for distributions in Python. Throughout the duration of this co-op position, I have gained many useful skills such as creating rolling averages, creating animated plots, and scripting using .py files. This project will continue under the supervision of David Steffen and Professor Marcus Müller of Georg-August Universität Göttingen alongside Professor Joerg Rottler; theory is to be created surrounding the distributions of the field on and field off densities with relation to space and time, as well as the distributions of the Rouse modes and centres of masses of the molecules.

References

Daoulas, Kostas Ch, Müller, Marcus. “Single chain in mean field simulations: Quasi-instantaneous field approximation and quantitative comparison with Monte Carlo simulations.” *J. Chem. Phys.*, 2006, https://aip.scitation.org/doi/10.1063/1.2364506

Müller, Marcus. “Memory in the relaxation of a polymer density modulation.” *J. Chem. Phys.*, 2022, https://aip.scitation.org/doi/10.1063/5.0084602

Padding, Johan T. “Theory of Polymer Dynamics.” *Advanced courses in Macroscopic Physical Chemistry (Han-sur-Lesse winterschool 2005)*, 2005

Schneider, Ludwig, Müller, Marcus. “Multi-architecture Monte-Carlo (MC) simulation of soft coarse-grained polymeric materials: SOft coarse grained Monte-Carlo Acceleration (SOMA).” *Computer Physics Simulations*, 2019, https://data.mendeley.com/datasets/j3thz43k93/1