

Title

Subtitle

by

Mikkel Metzsch Jensen

THESIS

for the degree of

MASTER OF SCIENCE



Faculty of Mathematics and Natural Sciences  
University of Oslo

Spring 2023



Title

Subtitle

Mikkel Metzsch Jensen



© 2023 Mikkel Metzsch Jensen

Title

<http://www.duo.uio.no/>

Printed: Reprosentralen, University of Oslo



# Abstract

Abstract.



# Acknowledgments

Acknowledgments.



# Contents

|  |            |
|--|------------|
| <b>List of symbols?</b>  | <b>vii</b> |
| <b>I Background Theory</b>   | <b>1</b>   |
| <b>II Simulations</b>  | <b>3</b>   |
| <b>1 Dataset study</b>   | <b>5</b>   |
| 1.1 Generating data  | 5          |
| 1.2 Data analysis  | 5          |
| 1.3 Properties of interest / Stretch profiles                      | 7          |
| 1.4 Machine learning   | 10         |
| 1.5 Accelerated Search   | 10         |
| 1.5.1 Markov-Chain Accelerated Genetic Algorithms                  | 10         |
| 1.5.1.1 Talk about traditional method also?                        | 10         |
| 1.5.1.2 Implementing for 1D chromosome (following article closely) | 10         |
| <b>Summary</b>   | <b>13</b>  |
| 1.6 Summary and conclusion   | 13         |
| 1.7 Outlook / Perspective  | 13         |
| <b>Appendices</b>  | <b>15</b>  |
| <b>Appendix A</b>  | <b>17</b>  |
| <b>Appendix B</b>  | <b>19</b>  |
| <b>Appendix C</b>  | <b>21</b>  |



# List of symbols?

Maybe add list of symbols and where they are used like Trømborg.



# Part I

# Background Theory



## **Part II**

# **Simulations**



# Chapter 1

## Dataset study

### 1.1 Generating data

We generate the dataset by simulating various sheet cut configurations under multiple combinations of normal load and stretch. For each configuration we sample 15 pseudo uniform (refer to relevant section here) stretch values between zero and the rupture stretch found in the rupture test. The normal force is uniformly sampled in the range  $[0.1, 10]$  nN. In total this gives  $3 \times 15$  data points for each configuration. For the remaining parameters we use the values presented in the pilot study (see table ??). We generate 68 configurations of the Tetrahedron pattern type, 45 of the Honeycomb type and 100 of the Random walk type which is shown in Appendix C?. A summary of the data points is given in table 1.1. The table shows that not all submitted data points “makes it” to the final dataset. This is due to the combination of our numerical procedure and small variations in the rupture stretch point. After performing the rupture test the simulation is restarted with a new substrate size corresponding to the measured rupture stretch limit and also with new random velocity and thermostat initializations values. The sheet is then stretched and checkpoints of the simulation state (LAMMPS restart files) are saved for each of the targeted stretch samples. However, if the rupture points arrives slightly early than suggested by the rupture test, some sampled stretch values might not get a corresponding checkpoint file. Thus, these data points are not included in the data set even though they ideally should have been noted as a rupture event. This could quite easily have been mitigated by a rewrite of that part of the code, but it was first discovered after the dataset had been created. However, the dataset still includes 11.57 % rupture events and it is most likely that the most cases with a lost rupturer event have a rupture event stored for the preceding stretch value instead which captures the information of the sheet stretch limit on its own.

**Table 1.1:** Summary of the number of generated data points in the dataset. Due to slight deviations in the rupture stretch and the specific numerical procedure not all submitted simulations “makes it” to the final dataset. Notice that the Tetrahedon (7, 5, 2) and Honeycomb (2, 2, 1, 5) from the pilot study is rerun as a part of the Tetrahedon and the Honeycomb datasets separately. In the latter datasets the reference point for the pattern is randomized and thus these configurations is not fully identical. This is the idea behind the difference of 2 in the total sum.

| Type        | Configurations | Submitted data points | Final data points | Ruptures       |
|-------------|----------------|-----------------------|-------------------|----------------|
| Pilot study | 3              | 270                   | 261               | 25 (9.58 %)    |
| Tetrahedon  | 68             | 3060                  | 3015              | 391 (12.97 %)  |
| Honeycomb   | 45             | 2025                  | 1983              | 80 (4.03 %)    |
| Random walk | 100            | 4500                  | 4401              | 622 (14.13 %)  |
| Total       | 214 (216)      | 9855                  | 9660              | 1118 (11.57 %) |

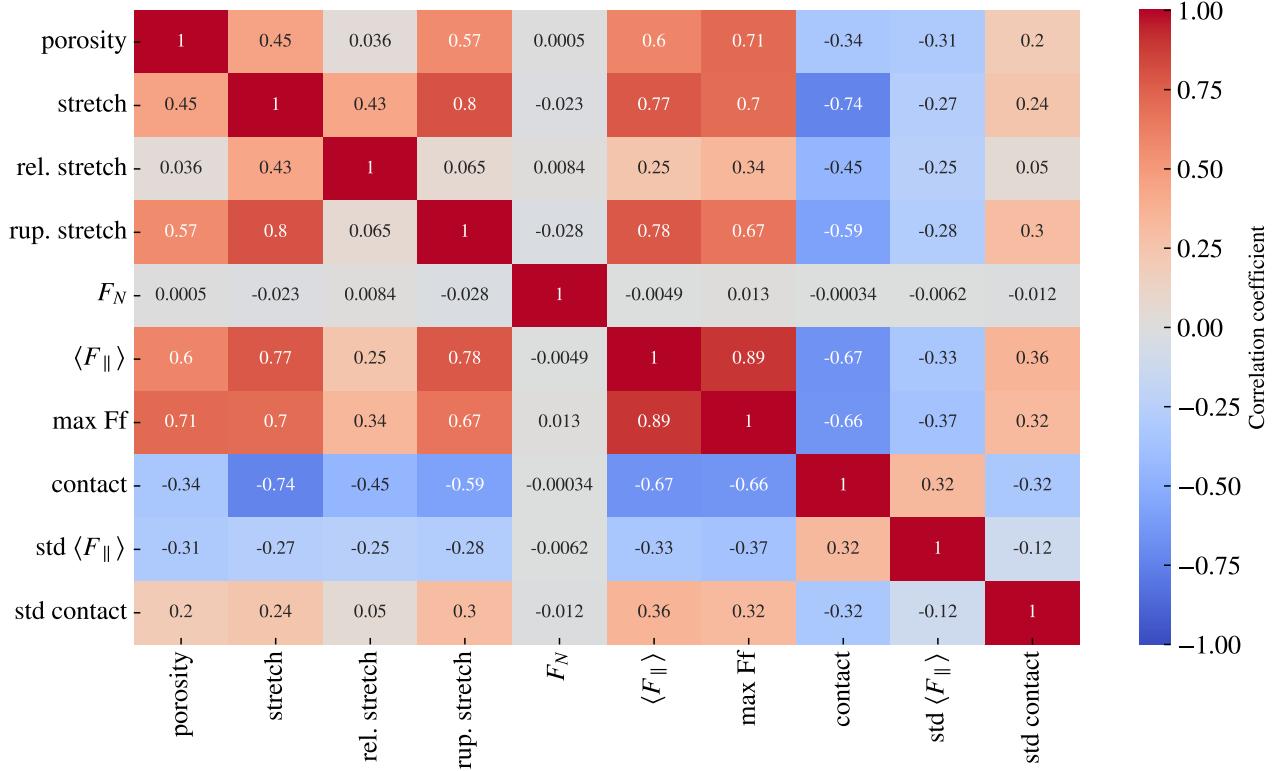
### 1.2 Data analysis

In order to gain insight into the correlations between variables associated to the simulations we calculate the correlations coefficients between all variable combinations. More specific, we are going to calculate the Pearson

product-moment correlation coefficient (PPMCC) for which is defined, between data set  $X$  and  $Y$ , as

$$\text{corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\langle (X - \mu_X)(Y - \mu_Y) \rangle}{\sigma_X \sigma_Y} \in [-1, 1]$$

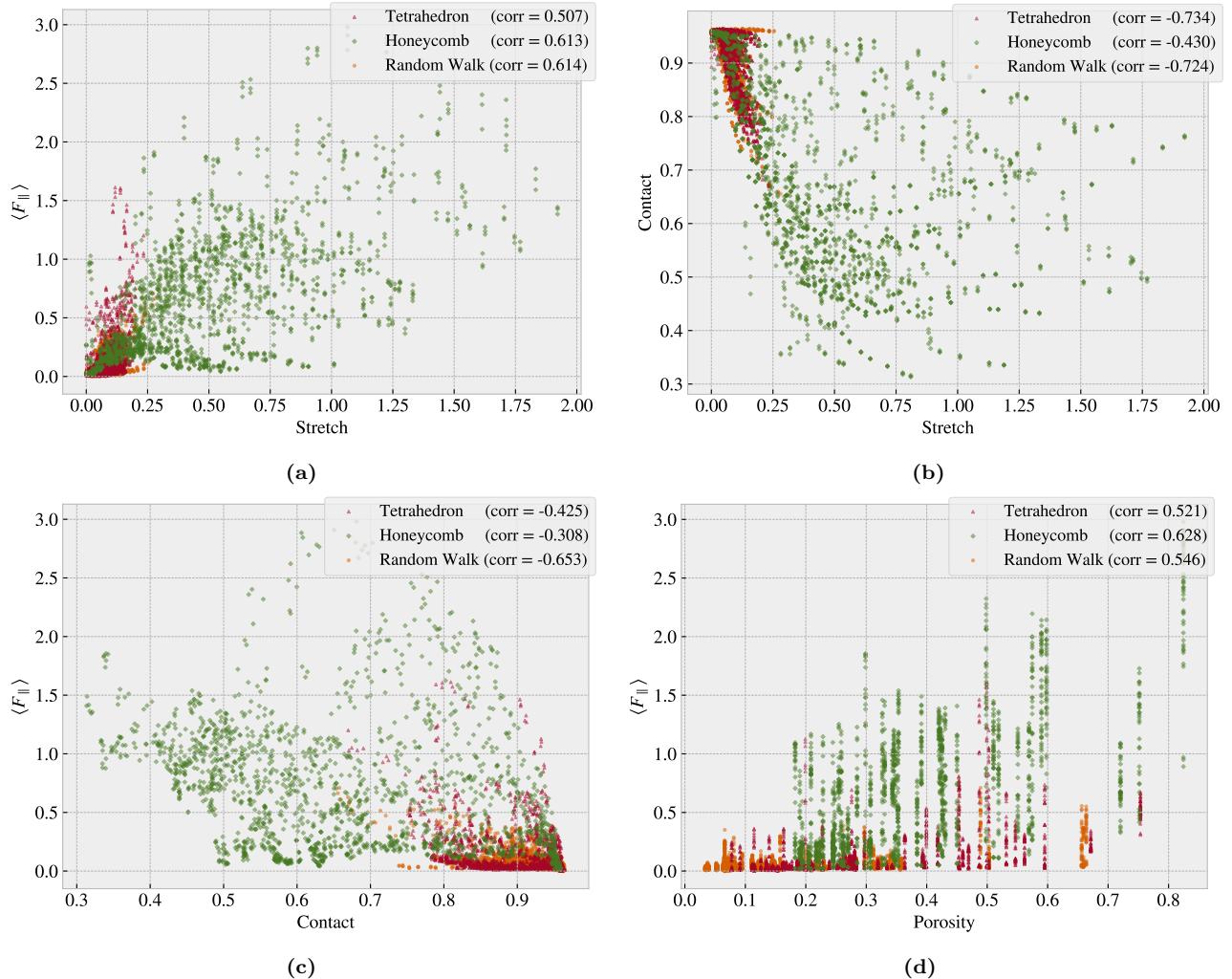
where  $\text{Cov}(X, Y)$  is the covariance,  $\mu$  the mean value and  $\sigma$  the standard deviation. The correlation coefficients ranges from perfect negative correlation ( $-1$ ) through no correlation ( $0$ ) to a perfect positive correlation ( $1$ ). The correlation coefficients is shown in figure 1.1



**Figure 1.1:** Pearson product-moment correlation coefficients for the full datset (see table 1.1).

From figure 1.1 we especially notice that the mean friction force  $\langle F_{\parallel} \rangle$  has a significant positively correlation with stretch (0.77) and porosity (0.60) (void fraction). However, the relative stretch, which is scaled by the rupture stretch, has a weaker correlation of only 0.25 which indicates that it is the absolute stretch value that has the most significant impact on the friction force increase during stretching. This is further supported by the fact that the mean friction and the rupture stretch is also strongly positively correlated (0.78). From figure 1.1 we also observe that the contact bond count is negatively correlated with the mean friction ( $-0.67$ ) and the stretch value ( $-0.74$ ) which is consistent with the trend observed in the pilot study (figure ?? and ??) of the contact decreasing with increasing stretch and mean friction. However, we must take note that the correlation coefficients is a measure of the strength and slope of a forced linear fit on the data. We clearly observed a non-linear relationship between stretch and mean friction for the tetrahedron and honeycomb pattern used in the pilot study (figure ??) where the relationship was partwise characterized by a postive correlation for some stretch ranges and partwise negative correlation for other stretch ranges. Hence, interesting strong regime-specific correlations might not be accurately highlighted by the correlation coefficients shown in figure 1.1.

In figure 1.2 we have visualized the data (excluding the pilot study) for chosen pairs of variables on the axes. In addition to a visual confirmation of how the given correlations look in a 2D plot we also get a feeling for the coverage in various areas of the parameter space that we are eventually going to feed the neural network. The honeycomb pattern is spanning a significant larger range of stretch, contact and mean friction makes the data rather biased towards the Honeycomb pattern in those areas.



**Figure 1.2:** Scatter plot of the data sets Tetrahedron, Honeycomb and Random Walk (excluding the pilot study) for various variable combinations in order to visualize some chosen correlations of interest and distributions in the data

### 1.3 Properties of interest / Stretch profiles

Define somewhere that we will look at low friction, high friction and the biggest (forward) drop in friction corresponding to a significant negative friction coefficient.

**Table 1.2:** Interesting properties

| <b>Tetrahedron</b>           | Configuration | Stretch          | Value [nN] |
|------------------------------|---------------|------------------|------------|
| Min $F_{\text{fric}}$        | (3, 9, 4)     | 0.0296           | 0.0067     |
| Max $F_{\text{fric}}$        | (5, 3, 1)     | 0.1391           | 1.5875     |
| Max $\Delta F_{\text{fric}}$ | (5, 3, 1)     | [0.0239, 0.1391] | 1.5529     |
| Max drop                     | (5, 3, 1)     | [0.1391, 0.1999] | 0.8841     |

| <b>Honeycomb</b>             | Configuration | Stretch          | Value [nN] |
|------------------------------|---------------|------------------|------------|
| Min $F_{\text{fric}}$        | (2, 5, 1, 1)  | 0.0267           | 0.0177     |
| Max $F_{\text{fric}}$        | (2, 1, 1, 1)  | 1.0654           | 2.8903     |
| Max $\Delta F_{\text{fric}}$ | (2, 1, 5, 3)  | [0.0856, 1.4760] | 2.0234     |
| Max drop                     | (2, 3, 3, 3)  | [0.5410, 1.0100] | 1.2785     |

| <b>Random walk</b>           | Configuration | Stretch          | Value [nN] |
|------------------------------|---------------|------------------|------------|
| Min $F_{\text{fric}}$        | 12            | 0.0562           | 0.0024     |
| Max $F_{\text{fric}}$        | 96            | 0.2375           | 0.5758     |
| Max $\Delta F_{\text{fric}}$ | 96            | [0.0364, 0.2375] | 0.5448     |
| Max drop                     | 01            | [0.0592, 0.1127] | 0.1818     |

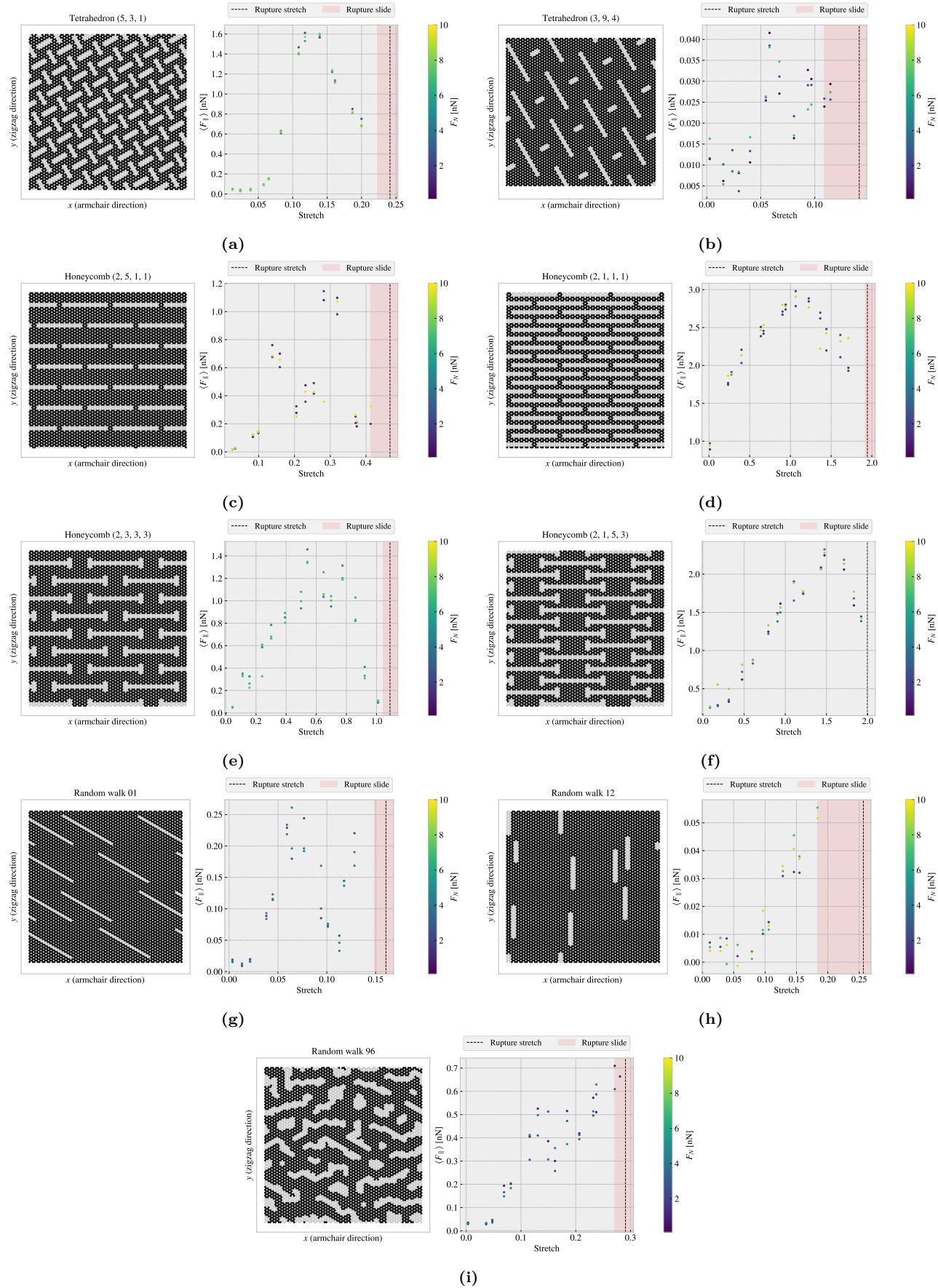


Figure 1.3

The stretch profiles for all the configurations are shown in appendix ??.

## 1.4 Machine learning

Staircase architecture tuning.

## 1.5 Accelerated Search

Having a network model that can predict friction force for a given configuration are able to search for some desired properties. Low and high friction and maximal negative friction coefficients

Here we pursue two different approaches for finding

1. Generate an enlarged dataset and run it through the ML model
2. Genetic algorithm

### 1.5.1 Markov-Chain Accelerated Genetic Algorithms

#### 1.5.1.1 Talk about traditional method also?

#### 1.5.1.2 Implementing for 1D chromosome (following article closely)

We have the binary population matrix  $A(t)$  at time (generation)  $t$  consisting of  $N$  rows denoting chromosomes and with  $L$  columns denoting the so-called locus (fixed position on a chromosome where a particular gene or genetic marker is located, wiki). We sort the matrix rowwise by the fitness of each chromosome evaluated by a fitness function  $f$  such that  $f_i(t) \leq f_k(t)$  for  $i \geq k$ . We assume that there is a transition probability between the current state  $A(t)$  and the next state  $A(t+1)$ . We consider this transition probability only to take into account mutation process (mutation only updating scheme). During each generation chromosomes are sorted from most to least fitted. The chromosome at the  $i$ -th fitted place is assigned a row mutation probability  $a_i(t)$  by some monotonic increasing function. This is taken to be

$$a_i(t) = \begin{cases} (i-1)/N', & i-1 < N' \\ 1, & \text{else} \end{cases}$$

for some limit  $N'$  (refer to first part of article talking about this). We use  $N' = N/2$ . We also define the survival probability  $s_i = 1 - a_i$ . In thus way  $a_i$  and  $s_i$  decide together whether to mutate to the other state (flip binary) or to remain in the current state. We use  $s_i$  as the statistical weight for the  $i$ -th chromosome given it a weight  $w_i = s_i$ .

Now the column mutation. For each locus  $j$  we define the count of 0's and 1's as  $C_0(j)$  and  $C_1(j)$  respectively. These are normalized as

$$n_0(j, t) = \frac{C_0(j)}{C_0(j) + C_1(j)}, \quad n_1(j, t) = \frac{C_1(j)}{C_0(j) + C_1(j)}.$$

These are gathered into the vector  $\mathbf{n}(j, t) = (n_0(j, t), n_1(j, t))$  which characterizes the state distribution of  $j$ -th locus. In order to direct the current population to a preferred state for locus  $j$  we look at the highest weight of row  $i$  for locus  $j$  taking the value 0 and 1 respectively.

$$\begin{aligned} C'_0(j) &= \max\{W_i | A_{ij} = 0; i = 1, \dots, N\} \\ C'_1(j) &= \max\{W_i | A_{ij} = 1; i = 1, \dots, N\} \end{aligned}$$

which is normalized again

$$n_0(j, t+1) = \frac{C'_0(j)}{C'_0(j) + C'_1(j)}, \quad n_1(j, t+1) = \frac{C'_1(j)}{C'_0(j) + C'_1(j)}.$$

The vector  $\mathbf{n}(j, t + 1) = (n_0(j, t + 1), n_1(j, t + 1))$  then provides a direction for the population to evolve against. This characterizes the target state distribution of the locus  $j$  among all the chromosomes in the next generation. We have

$$\begin{bmatrix} n_0(j, t + 1) \\ n_1(j, t + 1) \end{bmatrix} = \begin{bmatrix} P_{00}(j, t) & P_{10}(j, t) \\ P_{01}(j, t) & P_{11}(j, t) \end{bmatrix} \begin{bmatrix} n_0(j, t) \\ n_1(j, t) \end{bmatrix}$$

Since the probability must sum to one for the rows in the P-matrix we have

$$P_{00}(j, t) = 1 - P_{01}(j, t), \quad P_{11}(j, t) = 1 - P_{10}(j, t)$$

These conditions allow us to solve for the transition probability  $P_{10}(j, t)$  in terms of the single variable  $P_{00}j, t$ .

$$\begin{aligned} P_{10}(j, t) &= \frac{n_0(j, t + 1) - P_{00}(j, t)n_0(j, t)}{n_1(j, t)} \\ P_{01}(j, t) &= 1 - P_{00}(j, t) \\ P_{11}(j, t) &= 1 - P_{10}(j, t) \end{aligned}$$

We just need to know  $P_{00}(j, t)$ . We start from  $P_{00}(j, t = 0) = 0.5$  and then choose  $P_{00}(j, t) = n_0(j, t)$



# Summary

## 1.6 Summary and conclusion

## 1.7 Outlook / Perspective

- What did we not cover?
- What kind of further investigations does this study invite?

Things to include here

- Could be valuable to spend more time on the validation of the MD simulations. How does material choice and potential effects the results. How realistic is the simulations?
- Are there any interesting approaches for compressed kirigami structures?
- How does these results scale? I imagined that the nanomachine systems should be applied in small units to avoid scaling problems, but in general I could spend way more time on the scaling investigation.
- Since the normal force is applied at the pull blocks the normal force distribution changes from the sides more towards and even distribution as the sheet is put under tension (stretched). If we imagined a sheet for which the center part was either a different material or had some kind of pre-placed asperity on it, could we then exploit this force distribution to get exotic properties as well? By studying this we might get a clearer understanding of what is the cause of my results.
- Possibility to study hysteresis effects. Maybe the frictional behaviours change significantly through repeated cycles of stretch and relax.



# Appendices



# Appendix A



# Appendix B



# Appendix C



# Bibliography