

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: Reprocentralen, University of Oslo

Abstract

Abstract.

Acknowledgments

Acknowledgments.

Contents

List of symbols?	vii
I Background Theory	1
II Simulations	3
1 Defining the system	5
1.1 Region definitions (Sheet, pullblocks and substrate)	5
1.2 Numerical procedure	7
1.3 Creating the sheet	7
1.3.1 Graphene	7
1.3.2 Indexing	8
1.3.3 Removing atoms	9
1.4 Kirigami patterns	10
1.4.1 Tetrahedron	11
1.4.2 Honeycomb	12
1.4.3 Random walk	13
1.4.3.1 Fundamentals	14
1.4.3.2 Spacing of walking paths	14
1.4.3.3 Bias	15
1.4.3.4 Stay or break	16
1.4.3.5 Deployment schemes	17
1.4.3.6 Validity	18
1.4.3.7 Random walk examples	19
2 Pilot study	21
2.1 Friction simulation parameters	21
2.1.1 Pressure reference for normal load domain	23
2.2 Single friction simulation analysis	24
2.2.1 Force oscillations	24
2.2.2 Decompositions	25
2.2.3 Center of mass path	26
2.3 Defining metrics for dynamic and static friction	27
2.3.1 Dynamic friction	27
2.3.2 Static friction	28
2.4 Out of plane buckling	29
2.5 Investigating selected parameters	31
2.6 Normal force and stretch dependencies	33
2.6.1 Contact area	34
2.6.2 Stretch	34
2.6.3 Normal force	36
2.7 Computational cost	37

3 Generating data	39
3.1 Rupture	39
4 Training forward network (Data analysis?)	41
4.1 Dataset analysis	41
5 Accelerated Search	45
5.1 Markov-Chain Accelerated Genetic Algorithms	45
5.1.1 Talk about traditional method also?	45
5.1.2 Implementing for 1D chromosome (following article closely)	45
6 Negative friction coefficient	47
6.1 Manual coupling of normal force and stretch	47
6.2 Nanomachine coupling	48
Appendices	49
Appendix A	51

List of symbols?

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

Part I

Background Theory

Part II

Simulations

Chapter 1

Defining the system

The simulated system consists of two major parts: A 2D graphene sheet and a 3D Silicon “bulk” substrate. These parts interact with a van der Waals force (modelled by the LJ potential). We apply a normal load to the sheet inducing a normal force response between the sheet and substrate. By dragging the sheet along the substrate we measure the responding frictional forces.

Maybe include a few words about the preceding considerations between two different simulations approaches:
The one that we use now and the *graphene skin* setup, where one would consider the graphene sheet as the top part of the substrate and then measure friction with an indenting tip (AFM-style). This would probably be more directed towards grasping robot applications (with ability to stretch and relax the sheet before grasping) or creation of fixed pre-stretched surfaces in the case that certain friction coefficients of interest could be achieved here.

1.1 Region definitions (Sheet, pullblocks and substrate)

The system, sheet and substrate, is further subdivided according to functionality in the MD simulations. The sheet ends is reserved for so-called *pull blocks*, which is used application of normal load, stretching and dragging the sheet, and as a thermostat, while the remaining *inner sheet* is left as an untouched (NVE) canvas for kirigami cuts. The pull blocks are equal split between a thermostat part and a rigid part which is locked into a single rigid body after an initial relaxation period. Note that the rigid part of the pull blocks on both side is considered a single rigid object even though they are physically separated. This means that all force interactions on these parts will be applied as a common average making the move in total synchronization. The substrate is equally divided into three parts: The *upper layers* (NVE) responsible for the sheet-substrate interaction, the *middle layers* being a thermostat (NVT), and the *bottom layers* is frozen (rigid and fixed) in the initial lattice structure to ensure that the substrate stays in place. In figure 1.1 the system is displayed with colors matching the three distinct roles:

1. Red: NVE parts which is governing the frictional behaviour of interest.
2. Green: Thermostats (NVT) surrounding the NVE parts in order to modify the temperature without making disturbing changes to the interaction of the sheet and substrate.
3. Blue: Parts that is initially or eventually turned into rigid objects. For the substrate this is additionally locked off and immobile.

The total system size in terms of atom count is given in table 1.1 while the sheet length dimension is given in table ??

The total system size (without cuts in the sheet) is 27456 atoms, and the distribution into the various regions is shown in table 1.1. The length dimensions of the sheet is given in table 1.2.

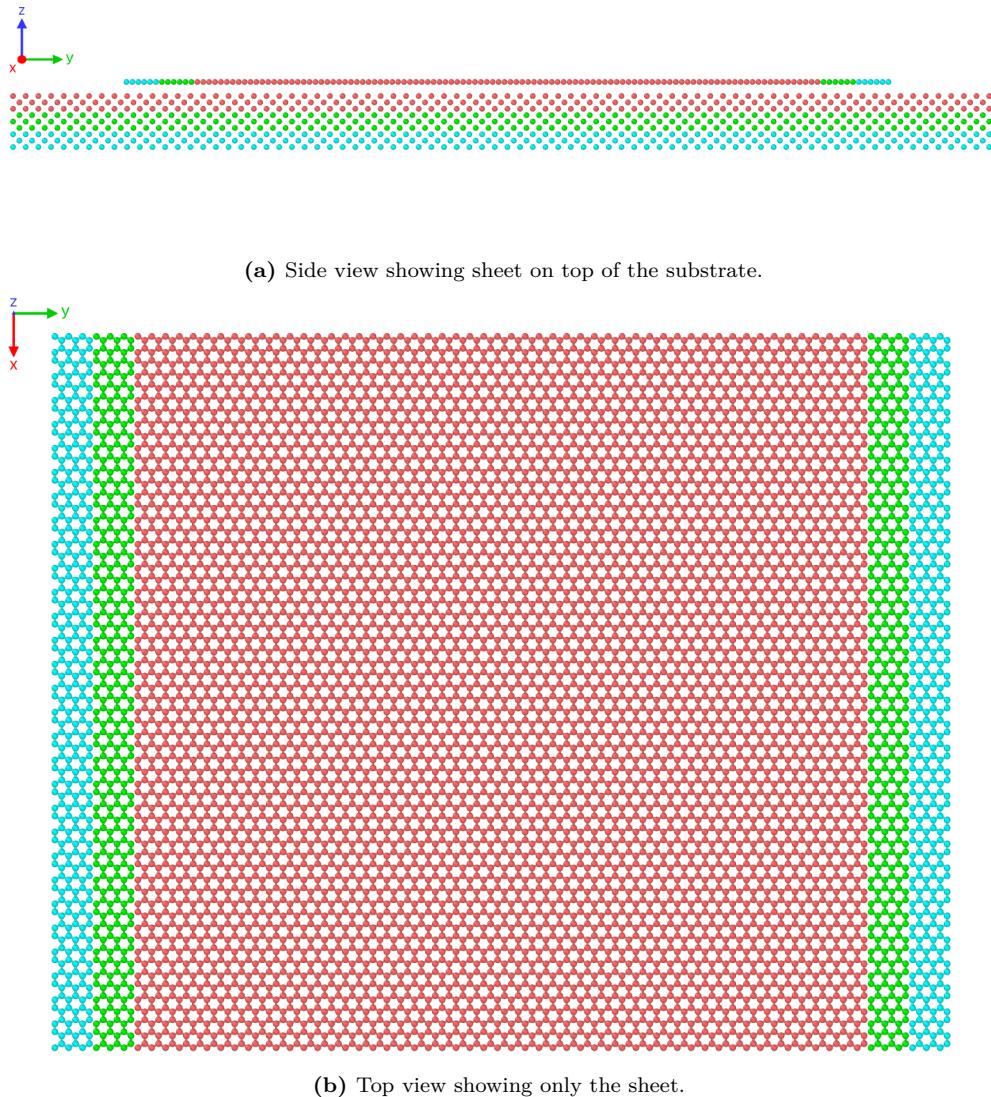


Figure 1.1: System configuration colorized to indicate NVE parts (red), thermostat parts (green) and rigid parts (blue).

Table 1.1: Amount of atoms in the various system regions in the case of no cutting applied to the sheet.

Region	Total	Sub region	Sub total	NVE	NVT	Rigid
Sheet	7800	Inner sheet	6360	6360	0	0
		Pull blocks	1440	0	720	720
Substrate	19656	Upper	6552	6552	0	0
		Middle	6552	0	6552	0
		Bottom	6552	0	0	6552
All	27456			12912	7272	7272

Table 1.2: Sheet dimensions comparing the full sheet to its subdivisions: inner sheet and pull blocks.

Group	x, y -dim	dim [Å]	Area [Å ²]
Full sheet	$x_S \times y_S$	$130.029 \times 163.219 \text{ Å}$	21, 223.203
Inner sheet	$x_S \times 81.40 \%_{y_s}$	$130.029 \times 132.853 \text{ Å}$	17, 274.743
Pull blocks	$2 \times x_S \times 9.30 \%_{y_s}$	$2 \times 130.029 \times 15.183 \text{ Å}$	$2 \times 1,974.230$

1.2 Numerical procedure

The numerical procedure for the friction simulations can be arranged as the following.

1. Relax (15 ps): The sheet and substrate is relaxed for 15 ps. They are both initially added in their crystalline form. The sheet is constrained under three hard spring forces (spring constant $10^5 \text{ eV/Å}^2 \sim 1.6 \times 10^6 \text{ N/m}$): One spring attaches the sheet center of mass (CM) to its original position preventing drift, while the remaining two are attached to the CM for the pull blocks to their initial position respectively to prevent rotation. These spring forces are immediately terminated after the relax phase. In this phase the pull blocks are only rigid with respect to the z-direction (perpendicular to the sheet). That is, all the forces in the z-direction are summed up and distributed on the pull blocks while it is free to expand and contract in the x-y-plane. This is mainly to ensure that it achieves the correct lattice spacing according to the temperature of the system. For the remaining phases the rigid parts of the pull block is in fact rigid with respect to all directions.
2. Stretch: The sheet is stretched by separating the rigid parts of the pullblock at constant velocity until the desired stretch amount is met.
3. Pause 1 (5 ps): The sheet is relaxed for 5 ps after the stretch procedure.
4. Pause 2 (Normal load): The normal load is applied to the rigid parts of the pull blocks together with a damping force to prevent hard impact between sheet and substrate as the separating distance is now reduced depending on the strength of the normal load. The damper is terminated after 0.5 ps, as this was suitable for the extreme load cases of our force range, and the system is relaxed until a total of 5 ps has passed.
5. Drag: A virtual atom is introduced into the simulation which exclusively interacts with the rigid parts of the pull through a spring force with variable spring constant K in the x-y-plane. The z-direction is not affected by the spring force and is governed by the balance between normal load and the normal force response from the sheet-substrate interaction. The virtual atom is immediately given a constant velocity corresponding to a variable *drag speed* parameter

At the initial timestep the three nearest neighbours (at distance 1.42 Å) of all graphene atoms are recorded. If these nearest neighbours exceed a threshold of 4 Å this raises a rupture flag which halts the simulation early. Thus, we effectively prevent any kind of wear on the sheet. For the substrate we do not perform such an analysis but only visually confirms that no wear is occurring under the most extreme simulation parameters.

1.3 Creating the sheet

We are going to create a 2D sheet graphene sheet.

1.3.1 Graphene

Graphene is a single layer of carbon atom, graphite is the bulk, arranged in a hexagonal lattice structure. We can describe the 2D crystal structure in terms of its primitive lattice vector and a basis. That is we populate each

lattice site by the given basis and translate it to fill the whole plane by any linear combination of the lattice vectors

$$\mathbf{T}_{mn} = m\mathbf{a}_1 + n\mathbf{a}_2, \quad m, n \in \mathbb{N}.$$

For graphene we have the primitive lattice vectors

$$\mathbf{a}_1 = a \left(\frac{\sqrt{3}}{2}, -\frac{1}{2} \right), \quad \mathbf{a}_2 = a \left(\frac{\sqrt{3}}{2}, \frac{1}{2} \right), \quad |\mathbf{a}_1| = |\mathbf{a}_2| = 2.46 \text{ \AA}.$$

Notice that we deliberately excluded the third coordinate as we only consider a single graphene layer on not the bulk graphite consisting of multiple layers stacked on top of each other. The basis is

$$\left\{ (0,0), \frac{a}{2} \left(\frac{1}{\sqrt{3}}, 1 \right) \right\}$$

It turns out that the spacing between atoms is equal for all pairs with an interatomic distance

$$\left| \frac{a}{2} \left(\frac{1}{\sqrt{3}}, 1 \right) \right| \approx 1.42 \text{ \AA}.$$

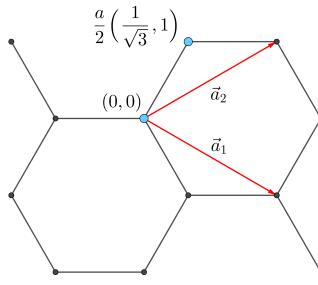
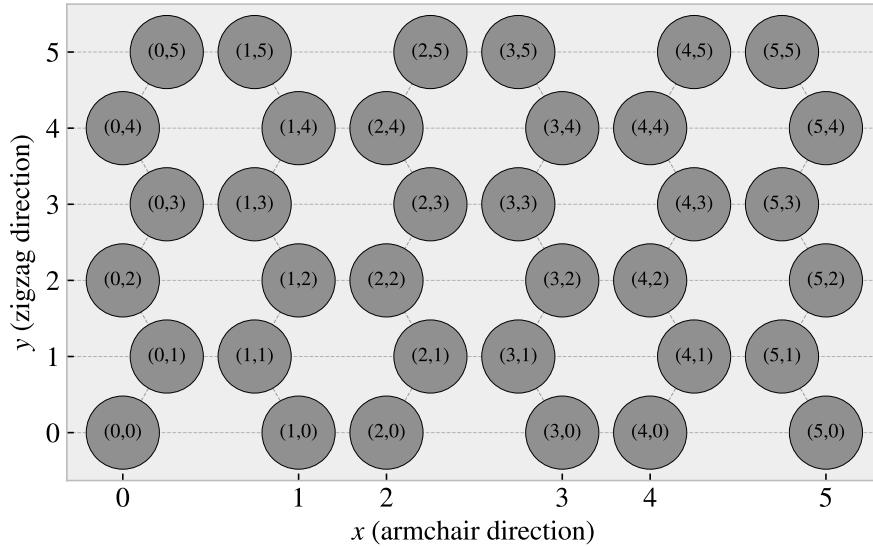


Figure 1.2: Graphene crystal structure with basis.

1.3.2 Indexing

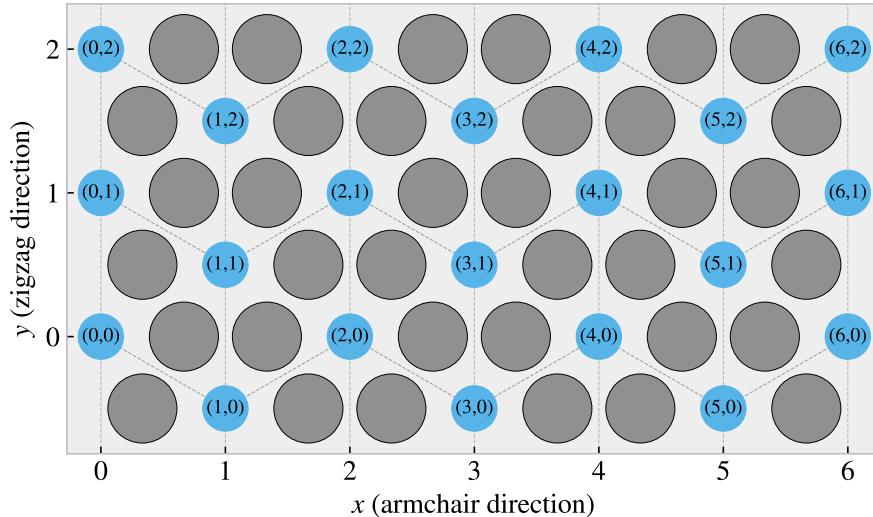
In order to define the cut patterns applied to the graphene sheet we need to define an indexing system. We must ensure that this gives an unique description of the atoms as we eventually want to pass a binary matrix, containing 0 for removed atoms and 1 for present atoms, that uniquely describes the sheet. We do this by letting the x-coordinate point to zigzag chains and the y-coordinate to the position along that chain. This is illustrated in figure 1.3. Other solutions might naturally invoke the lattice vectors, but as these are used to translate between similar basis atoms unfortunate duality is introduced as one need to include the basis atom of choice into the indexing system. Additionally, we want a system where the indexes reflect the relative physical position of neighbours . That is, atom (i, j) is in the proximity of $\{(i+1, j), (i-1, j), (i, j+1), (i, j-1)\}$. However, only three of them is categorized as nearest neighbours due to the hexagonal structure of the lattice. While $(i, j \pm 1)$ is always a nearest neighbour the neighbour in the x-direction flip sides when incrementing either x- or y-coordinate. That is the nearest neighbours (NN) is decided as

$$(i+j) \text{ is even} \rightarrow \text{NN} = \{(i-1, j), (i, j+1), (i, j-1)\}, \\ (i+j) \text{ is odd} \rightarrow \text{NN} = \{(i+1, j), (i, j+1), (i, j-1)\}. \quad (1.1)$$

**Figure 1.3:** Graphene atom indexing

1.3.3 Removing atoms

As a mean to ease the formulation of cut patterns we introduce pseudo center element in each gap of the hexagonal honeycombs, see figure 1.4.

**Figure 1.4:** Graphene center indexing

Similar to the case of the indexing for the carbon atoms themselves the nearest neighbour center elements alternate with position, this time only along the x-coordinate index. Each center element has six nearest neighbours, in clock wise direction we can denote them: “up”, “upper right”, “lower right”, “down”, “lower left”, “upper left”. The “up” and “down” is always accessed as $(i, j \pm 1)$, but for even i the $(i + 1, j)$ index corresponds to the “lower right” neighbour while for odd i this corresponds to the “upper right” neighbour. This shifting applies for all left or right neighbours and the full neighbour list is illustrated in figure 1.5.

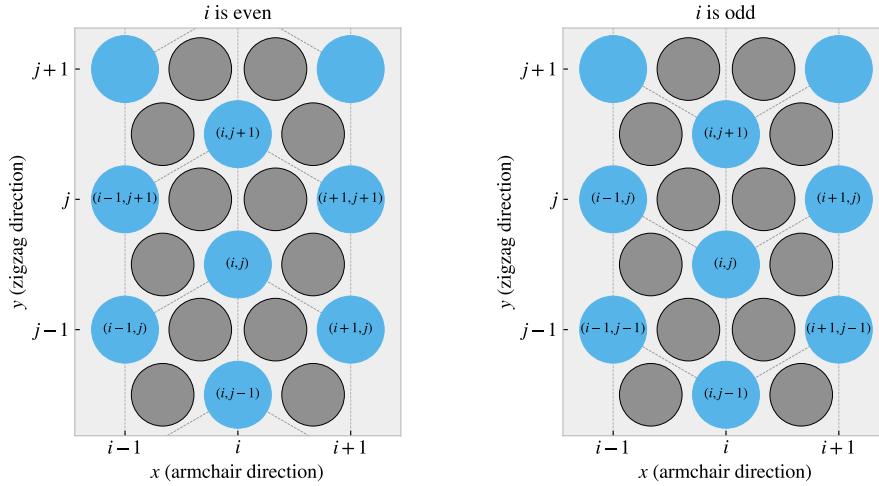


Figure 1.5: Graphene center elements directions

We define a cut pattern by connecting center elements into connected paths. As we walk element to element we remove atoms according to one of two rules

1. Remove intersection atoms: We remove the pair of atoms placed directly in the path we are walking. That is, when jumping to the “up” center element we remove the two upper atoms located in the local hexagon of atoms. This method is sensitive to the order of the center elements in the path.
2. Remove all surrounding atoms: We simply remove all atoms in the local hexagon surrounding each center element. This method is independent of the ordering of center elements in the path.

We notice that removing atoms using either of these rules will not guarantee an unique cut pattern. Rule 1 is the more sensitive to paths but we realize that, for an even i , we will remove the same five atoms following either of the following paths.

$$\begin{aligned}
 (i, j) &\rightarrow \underbrace{(i + 1, j + 1)}_{\text{upper right}} \rightarrow \underbrace{(i, j + 1)}_{\text{up}} \rightarrow \underbrace{(i + 1, j + 2)}_{\text{uppright + up}} \rightarrow \underbrace{(i + 1, j + 1)}_{\text{upper right}} \\
 (i, j) &\rightarrow \underbrace{(i + 1, j + 1)}_{\text{upper right}} \rightarrow \underbrace{(i + 1, j + 2)}_{\text{uppright + up}} \rightarrow \underbrace{(i, j + 1)}_{\text{up}}
 \end{aligned}$$

For rule 2 it is even more obvious that different paths can result in the same atoms being removed. This is the reason that we needed to define and indexing system for the atom position itself even though that all cuts generated manually will use the center element path as reference.

Illustrate some delete path?

1.4 Kirigami patterns

We propose a series of kirigami inspired cut patterns for the altering of the graphene sheet. We seek inspiration from macroscale patterns that showcases a considerable amount of out of plane buckling when stretched. We choose to imitate two different designs: 1) An alternating repeating series of perpendicular cuts as shown in figure 1.6a commonly used in studies of morphable metematerials [1]. This pattern produce surface buckling with a tetrahedron (three sided pyramid) shape when stretched. 2) A more intricate pattern shown in figure 1.6b which is used Scotch™ Cushion Lock™ [2] as protective wrap for items during shipping. This pattern buckles into a hexagonal honeycomb structure when stretched. In addition to the modeling of the so-called *tetrahedron* and *honeycomb* patterns we also create a series of random walk styled cut patterns.



Figure 1.6: Macroscale kirigami cut patterns used as inspiration for the nanoscale implementation. (a) Alternating perpendicular cuts producing a tetrahedron shaped surface buckling when stretched [1]. (b) ScotchTM Cushion LockTM [2] producing a honeycomb shaped surface buckling when stretched.

1.4.1 Tetrahedron

The *tetrahedron* pattern is defined in terms of center elements for which all atoms surrounding a given center element are removed. The pattern is characterized by two straight cuts, here denoted line 1 and line 2, arranged perpendicular to each other such that one line aligns with the center of the other line and with a given spacing in between. In order to achieve perpendicular cuts we cannot rely purely on the six principal directions corresponding to the center element neighbours which is spaced by 60° . Instead, we let line 1 run along the center elements in the direction of the “upper right” center elements (and “lower left”) while line 2 goes in the direction between the “lower” and “lower right” center elements corresponding to the direction $(1/\sqrt{3}, -1)$. We define variations of the pattern by the number of center elements L_1 and L_2 in line 1 and 2 respectively together with the spacing between the lines d as the tuple (L_1, L_2, d) . The pattern is constructed by translating the two lines to the whole sheet while according to the spacing. Due to the alignment criterias of having one line point to the center of the other line we can only have odd line length to have a clearly defined center element in the center of each line. Furthermore, in order to ensure that each translated center element stays on the same odd or even type center element we must in practice require that $|L_2 - L_1| = 2, 6, 10, \dots$. In figure 1.7 we see a visual representation of the pattern components for the $(7, 5, 2)$ pattern.

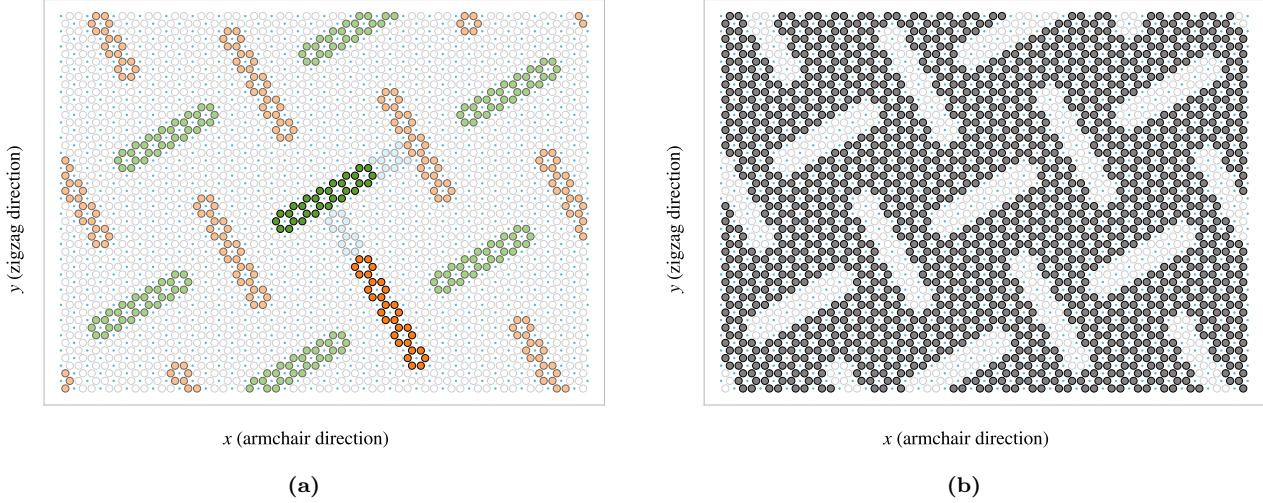


Figure 1.7: Visual representation of the tetrahedron pattern consisting of two perpendicular lines, line 1 and line 2, of length L_1 and L_2 respectively with spacing d . This example used $(L_1, L_2, d) = (7, 5, 2)$. (a) Highlight of the atoms removed. Line 1 is shown in green and line 2 in orange, with lighter colors for the translated variations, and the spacing is shown in light blue. (b) The sheet after applying the cut pattern where the grey circles denote atoms and the transparent white denotes removed atoms. The small blue circles show the center elements for reference

In addition to the three parameters L_1, L_2, d , the pattern is also anchored to a reference point which describes the position of line 1 and 2 before translating to the whole sheet. Due to the repeating structure of the pattern there exist a small finite number of unique reference positions. For the pattern $(7, 5, 2)$ used as an example in figure 1.7 there are 140¹ Some additional variation of the pattern deviation from the example in 1.7 is showcased in figure 1.8

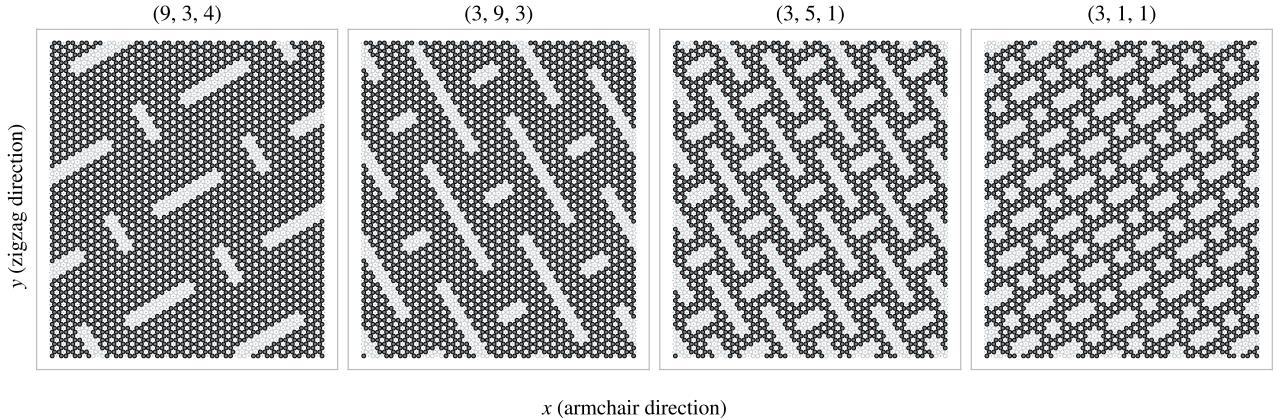


Figure 1.8

1.4.2 Honeycomb

The *honeycomb* pattern is defined, similarly to the tetrahedron pattern, in terms of center elements for which all atoms surrounding a given center element are removed. The honeycomb pattern is build from a repeating series of cuts reminiscent of a roman numeral one put on its side (I^\top). With a given spacing these are put next to each

¹The calculation of this is rather complicated in comparison of the importance in this context. Thus, we exclude the formula for this calculation as the derivation is rather handwavy and the number stated here is numerically backed for this specific parameter set.

other in the x-direction (↔↔↔) to achieve a row where only a thin *bridge* in between each cut is left to connect the sheet in the y-direction. By placing multiple rows along the y-direction with alternating x-offset we get the class of honeycomb patterns as visualized in figure 1.9. The pattern is described in terms of the parameters: (x-width, y-width, bridge thickness, bridge length) which is annotated in figure 1.9a where the parameters (2, 2, 1, 5) is used as an example. Some additional variations of the pattern class is showcased in figure 1.10

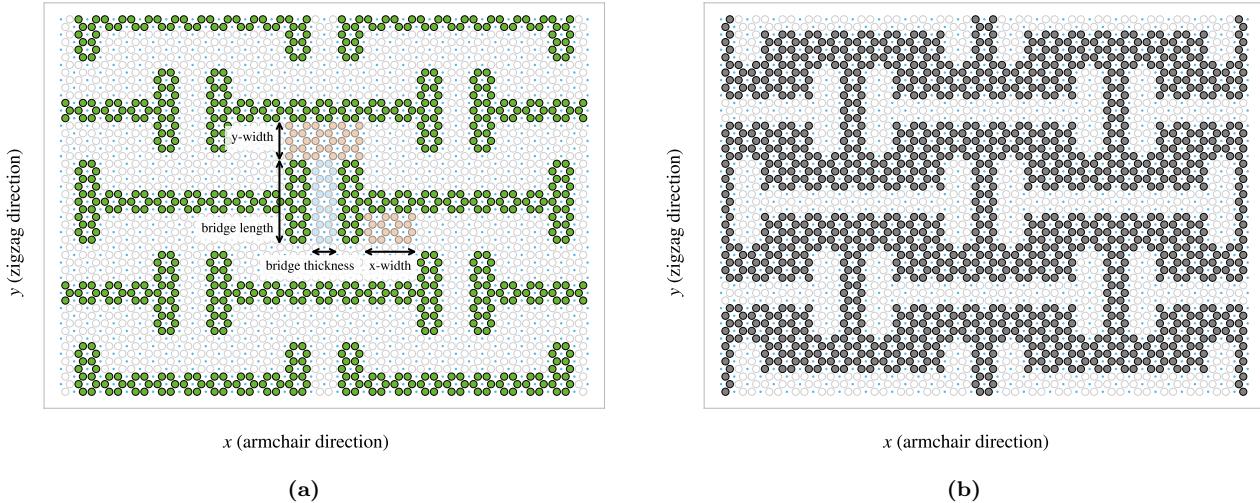


Figure 1.9

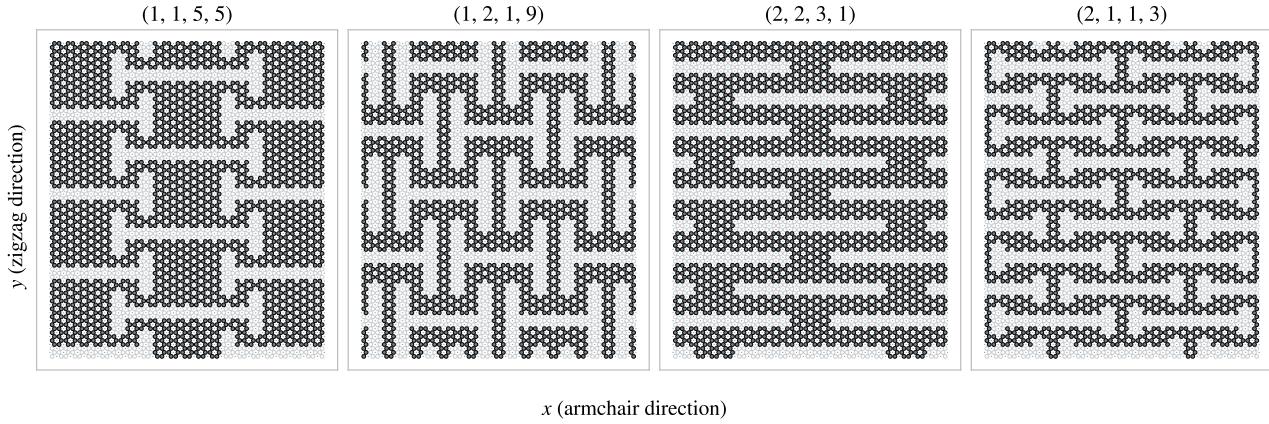


Figure 1.10

1.4.3 Random walk

The random walk serves as a method for introducing random patterns into the dataset with the scope of populating the configuration space more broadly than achieved with the more systematic patterns classes described above. By this argument a more straightforward way to create random configurations could be achieved by random noise, either uniform or gaussian. However, this would often leave the sheet detached with lots of non-connected atom clusters, and intuitively we do not find this promising for the generation of large scale organized structures which is hypothesized to be of interest. The random walk pattern generation is characterized by the parameters summarized in table 1.3 which will be introduced throughout the following paragraphs.

Table 1.3: Parameters for the random walk generator.

Parameters	Value	Description
Num. walkers (M)	Integer ≥ 1	Number of random walks to be initiated on the sheet (one at a time).
Max. steps (S)	Integer ≥ 1	The maximum steps allowed for any random walker.
Min. distance	Integer ≥ 0	The minimum distance required between any future paths and the previous paths in terms of the least walking steps in between.
Bias	Vector	Bias direction and strength defining the discrete probability for the choice of the next site.
Connection	Atoms / Center elements	Walk between atoms or between center elements (removing all adjacent atoms).
Avoid invalid	True/False	Whether to remove already visited sites from the neighbour list before picking the next site. This prevents jumping to already visited site and lowers the likelihood of early termination.
Stay or break	$p = [0, 1]$	
Periodic	True/False	Whether to use periodic boundary conditions of all four sides.
Avoid clustering	Integer ≥ 0	Amount of times to retry random walk in order to avoid detached clusters. Non-spanning clusters are removed after this amount of tries.
RN6	True/False	Randomly change the bias direction between one of the six center element directions for each random walker deployed.
Grid start	True/False	The option to have the random walkers start in an evenly spaced grid.
Centering	True/False	Relocate the path of a random walk after termination such that the path center of mass gets closer to the starting point (without violating the rules for travelling on already visited sites).

1.4.3.1 Fundamentals

For an uncut sheet we deploy M random walkers one at a time and let them walk for a maximum number of S steps. We can either let the walker travel between atom sites, removing the atoms in the path as it goes, or between center elements, removing all surrounding atoms - *Connection: Atom/Center elements*. Nonetheless, we will always remove a site once visited such that the walker itself or any other walker cannot use this site again. This corresponds with the property of a self avoiding random walk, but it furthermore constraint the walkers not to visit any path previously visited by another walker on the sheet. By default, the walker has an equal chance of choosing any of its adjacent neighbours for the next step, i.e. we draw the next step from a discrete uniform distribution. Optionally we can use periodic boundary conditions, *Periodic: True/False*, allowing neighbouring sites to be connected through the edge in both the x and y-direction. When traveling on atom sites this gives three neighbour options for the next step while traveling on the center elements gives six neighbour options. If the walker happens to arrive at an already visited site the walk is terminated early. Optionally, we can choose uniformly between the remaining options instead. This prolongs the walking distance. However, the walker is still able to find itself in a situation where no neighbouring sites is available, note that it cannot backtrack its own path either, and in such a case the walk is always terminated despite the setting of *Avoid invalid*.

1.4.3.2 Spacing of walking paths

In order to control the spacing between the paths of the various walkers we implement a so-called *minimum distance: 0, 1, ...* parameter, describing the spacing required between paths in terms of the least amount

of steps. When a walker has ended its walk, either by early termination or hitting the maximum limits of steps, all sites within a walking distance of the minimum distance is marked as visited, although they are not removed from the sheet. This prevents any subsequent walkers to visit those sites in their walk according to the general behaviour described in the previous paragraph. In practice this is done through a recursive algorithm as described in algorithm 1. For a given path the function `walk_distance()` is called with the input being a list of all sites in the given paths. The function will then for each site gather all site neighbours (regardless of their state on the sheet) and call itself using this neighbour list as input while incrementing a distance counter passed along. This will result in an expansion along all possible outgoing paths from the initial path of interest which is terminated when the distance counter hits the minimum distance limit. The function will then return the final neighbour lists which is cummulated into a final output corresponding to a list of all sites within the minimum distance to the path.

Algorithm 1 Recursive algorithm implemented as class method to mark sites within a distance of the class attribute `self.min_dis`.

```

Require: self.min_dis > 0                                ▷ This pseudocode does not handle other cases
1: function WALK_DISTANCE(self, input, dis = 0, pre = [ ])      ▷ Initialize list for new neighbours
2:     new_neigh ← []
3:     for site in input do
4:         neigh ← get_neighbouring_sites(site)                  ▷ Get sourrounding neighbours
5:         for n in neigh do
6:             if (n not in pre) and (n not in new_neigh) then          ▷ If not already added
7:                 AddItem(new_neigh, n)
8:             end if
9:         end for
10:    end for                                                 ▷ Increment distance counter
11:    dis += 1                                               ▷ Max limit hit
12:    if dis ≥ self.min_dis then
13:        return input + new_neigh
14:    else                                                 ▷ Start a new walk from each of the neighbouring sites
15:        pre ← input
16:        return pre + self.walk_distance(new_neigh, dis, pre)
17:    end if
18: end function

```

1.4.3.3 Bias

We include the option perform biased random walk through the Bias: (direction, strength) parameter option. We implement this by modelling each walking step as an analog to the canonical ensemble under the influence of an external force \mathbf{F} representing the bias. For such a system each microstate i , corresponding to the sites in the neighbour list, has the associated probability p_i given by the Gibbs–Boltzmann distribution

$$p_i = \frac{1}{Z} e^{-\beta E_i}, \quad Z = \sum_i e^{-\beta E_i},$$

where Z is the canonical partition function, $\beta = 1/k_B T$ for the boltzmann constant k_B and temperature T , and E_i the energy of site i . We model the energy of each site as the work required to move there. For a step \mathbf{s} the energy becomes $E_i = -\mathbf{s} \cdot \mathbf{F}$, where we notice that the energy is negative for alligned bias and step analogous to an energy gain by moving there. Due to the symmetry of both the atom sites and the center elements sites the step length to neighbouring sites will always be equal. By defining the bias magnitude $B = \beta|\mathbf{F}||\mathbf{s}|$ we get that the probability for jumping to site i is given by

$$p_i = \frac{1}{Z} e^{B \hat{\mathbf{s}} \cdot \hat{\mathbf{F}}} \propto e^{B \hat{\mathbf{s}} \cdot \hat{\mathbf{F}}},$$

where the hat denotes the unit direction of the vector. The bias magnitude B captures the opposing effects of the magnitude of the external force and the temperature of the system as $B \propto |\mathbf{F}|/T$. We notice that

$\hat{\mathbf{s}} \cdot \hat{\mathbf{F}} = \cos(\theta)$ for the angle θ between the step and bias direction. This shows that the bias will have the biggest positive contribution when the step direction is aligned with the bias direction ($\theta = 0$), have no contribution for orthogonal directions ($\theta = \pm\pi/2$) and biggest positive contribution when the directions are antiparallel ($\theta = \pi$). The partition function serves simply as a normalization constant which in practice is excluded ($Z = 1$) from the calculation of p_i initially and then enforced at the final stage as a division by the sum of all p_i . In the numerical implementation we then pick the step destination weighted by the discrete probability distribution p_i . In figure 1.11 we have illustrated how a bias of different strength impacts the probability distribution for a random walk between center elements. We can visually confirm that the bias will favorize the directions that lies closer to the bias direction. This favorization is more distinct at high bias strengths while at low strength $B \rightarrow 0$ we get a uniform distribution which aligns with the default unbiased random walk.

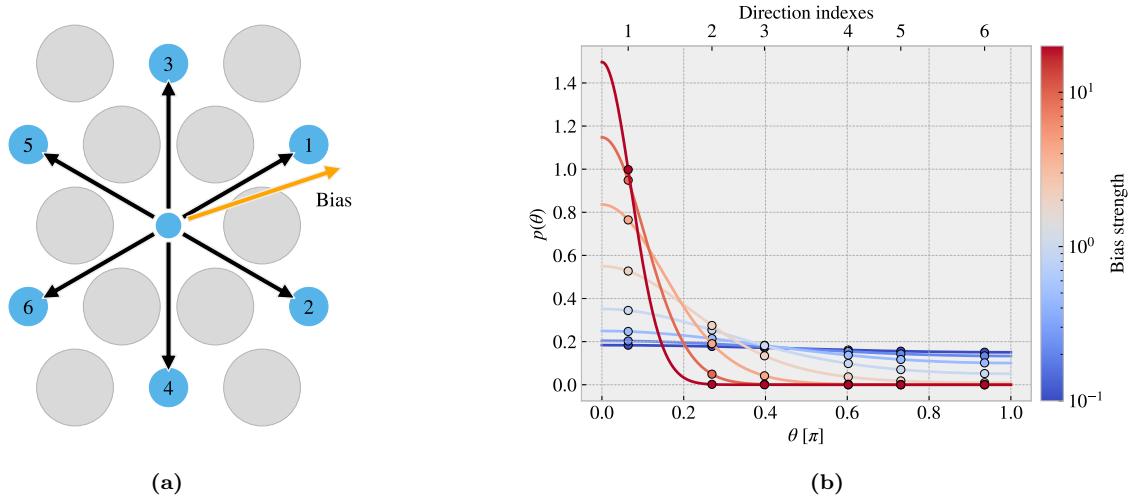


Figure 1.11: Illustration of the probability distribution for the various step direction during a bias random walk between center elements. (a) Shows the possible step directions as black arrows pointing towards the neighbouring center elements shown as blue circles. The bias direction is denoted as an orange arrow and the numbering indicates the most likely direction to take (1) towards the least likely (6). The atoms sites are marked as grey circles for reference. (b) The probability distribution as a function of angle between the direction of choice and the bias direction. The distribution is normalized according to the discrete probabilities marked with dots for which the continuous line simply highlights the curve of the distribution. The direction indexes corresponds to the numbering on figure (a). The color map indicates different strengths of the bias.

1.4.3.4 Stay or break

The *Stay or break* parameter defines the probability p_{stay} that the walker will keep its direction or otherwise break into a different direction by probability $1 - p_{\text{stay}}$. That is we manually substitute in the p_{stay} in the discrete probability for the direction corresponding to the direction of the last jump. We then shift the remaining probabilities such that the distribution sum to one again. In this way we can still perform bias random walk in combination. For the center element walk it is trivial to determine which of the neighbour directions correspond to the direction of the last jump. However, due to the layout of the atom sites, it is not possible to follow the same direction continuously when performing an atom site type walk. We recall that the nearest atom neighbour indexes alternates for each increment in x or y position ((i, j)-index) (see eq. (1.1)) which yields alternating directions as

$$(i+j) \text{ is even} \rightarrow D = \left\{ \frac{a}{2} \left(\frac{-2}{\sqrt{3}}, 0 \right), \frac{a}{2} \left(\frac{1}{\sqrt{3}}, 1 \right), \frac{a}{2} \left(\frac{1}{\sqrt{3}}, -1 \right) \right\},$$

$$(i+j) \text{ is odd} \rightarrow D = \left\{ \frac{a}{2} \left(\frac{2}{\sqrt{3}}, 0 \right), \frac{a}{2} \left(\frac{-1}{\sqrt{3}}, 1 \right), \frac{a}{2} \left(\frac{-1}{\sqrt{3}}, -1 \right) \right\}.$$

Hence, we use the six directions from the center element walk as the common direction to stay or break from. As showcased in figure 1.12, for each center element direction (black arrows) there are two possible atom directions (red and orange arrows) that are equally close to the center element direction. The red and orange arrows

represent $(i + j)$ being even or odd respectively, and we notice that these appear in pairs such that we can always determine which of the atom directions are closest to the center element direction. Following this idea we can map each center direction to an atom direction depending on the even or oddness of the position. For $p_{\text{stay}} = 1$ this results in zigzag motion along the center element direction that happens to start on.

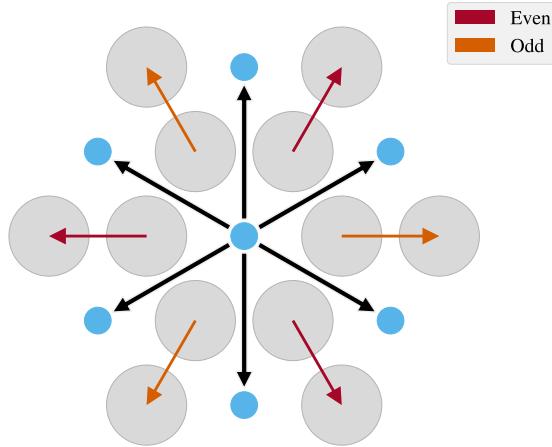


Figure 1.12: ...

In cases where the site corresponding to following the previous direction is not available we get a break of direction by definition.

1.4.3.5 Deployment schemes

By default, each random walker is given an uniform random starting point among the non-visited sites left on the sheet. This includes any modifications in relation to the minimum distance parameter. By toggling the *Grid start: True/False* parameter on the starting points are instead predefined on evenly spaced grid. That is, the sheet is subdivided into the least amount of squares that will accommodate a space for each starting point. {1} walker leads to a 1×1 partition, {2, 3, 4} walkers lead to a 2×2 partition, {5, 6, 7, 8, 9} walker lead to a 3×3 partition and so on. The lower left corner² is then chosen as a default starting place for the first walker for which the remaining sites are filled according to the order that maximizes the minimum distance between a new starting point and the ones already used. The population of the grid is visualized in figure 1.13 for 1-9 walkers in total.

²In hindsight this would have been statistically better spaced if with a random starting corner, but this is not considered to be of great importance for the way we used this feature in the dataset generation.

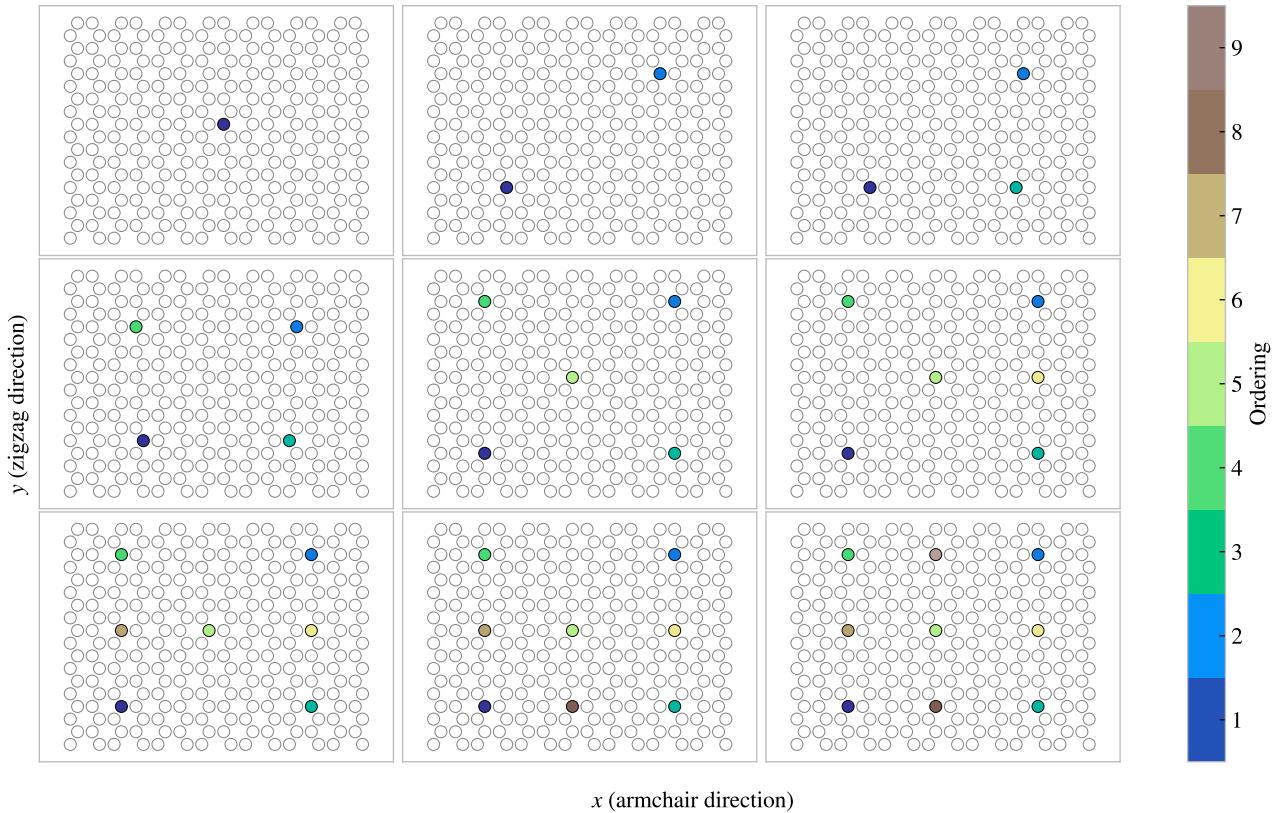


Figure 1.13: Population of starting points with the centering parameter toggled on in a 14×18 sheet. This is shown for 1-9 number of random walkers with the color map conveying the order of the population.

The *Centering: True/False* parameter let us relocate the path of the random walker such that the path center of mass alligns better with the starting point. This can be used in combination to the grid start and the bias parameter to make rather ordered configurations. In addition, the *RN6:True/False* parameter can be used change the bias direction to on of the six directions of the center element walk for each new walker. This lets us create configrations like [EXAMPLE].

1.4.3.6 Validity

The simulation procedure requires the sheet is fully attatched which can be summarized as the following requirements.

1. There exist only a single cluster on the sheet. We define a cluster as the set of atoms which can all be reached through nearest neighbour walking on the cluster.
2. The cluster of atoms is spanning the sheet in the y-direction. This means that there exist at least one path through nearest neighbour walks that connect the bottom and the top of the sheet. This is due to the reason that the sheet must be attatched to the pull blocks.

In order to accommodate these requirements we count the number of clusters and search for a spanning cluster after all walkers have terminated. If the requirements are not met we simply rerun the random walk from scratch. This is done *Avoid clustering* amount of times. If the requirements are not met during any of those reruns the non-spanning clusters are simply removed. In the case of no spanning cluster the configuration is dropped. This crude scheme was later reinvented as a more refined repair scheme which alters the sheet by the intention of performing the least amount of changes (addition of subtraction of atoms) in order to meet attachment requirements. This was done as a part of the accelerated search procedure and hence it was not utilized in the creation of the random walk dataset.

1.4.3.7 Random walk examples

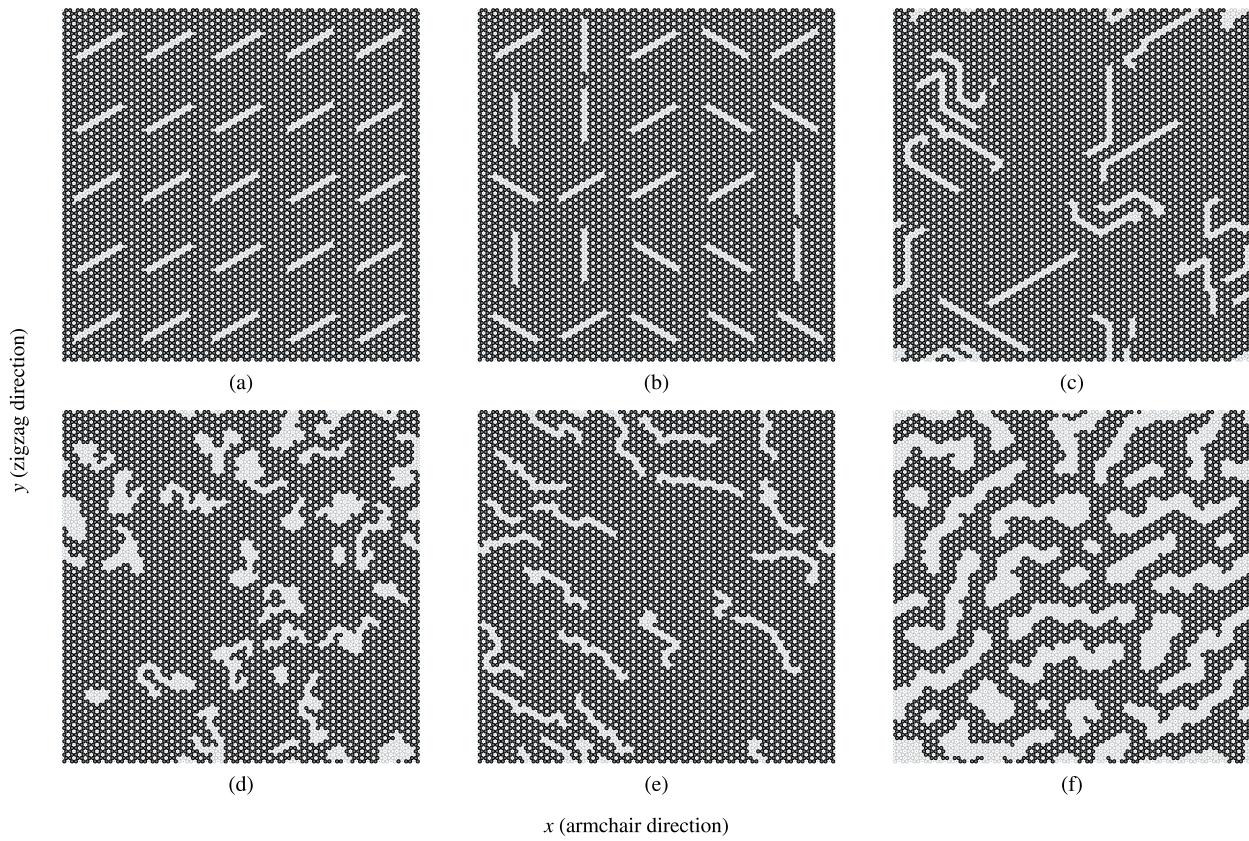


Figure 1.14: Some example uses of the random walking class.

Chapter 2

Pilot study

Consider relabeling drag length to sliding distance instead.

2.1 Friction simulation parameters

The friction simulation is governed by a set of parameters where some is kept constant while other is varied to gain insight in the frictional properties. These parameters can be categorised into three main categories of different purpose as described in table 2.1.

Table 2.1: Parameters of the numerical procedure for measuring friction.

Category	Parameter name: description	Category purpose
Physical	<ul style="list-style-type: none"> - T: Temperature for the Langevin thermostat. - v_{slide} : Sliding speed for the sheet translation. - K: Spring constant for the spring force between the virtual atom and the pull blocks responsible for translating the sheet along the substrate. An infinite spring constant is achieved by moving the pull blocks as a rigid body (Lammps: fix move). - Scan angle: The direction for which we translate the sheet. 	Parameters expected to have a physical effect on the friction properties, which is kept fixed and thus not included in the machine learning input set.
Measurement	<ul style="list-style-type: none"> - dt: Integration timestep. - t_R: Relaxation time before stretching. - Pauses between stretch and adding normal force and between dragging the sheet. - Stretch Speed: How fast to stretch the sheet. - Slide distance: How far to translate the sheet. - Sheet size: Spatial size of the 2D sheet. - Pull block size: spatial size of the pull blocks. 	Parameters influencing the simulation dynamics and being representative of the experimental procedure that we are mimicking. These parameters are chosen with the aim of getting stable parameters under small perturbations of the given parameter.
ML input	<ul style="list-style-type: none"> - Sheet configuration: A binary matrix containing information of which atoms are removed (0) and which are still present (1) in the graphene sheet. - Stretch amount: The relative sheet stretch in percentage. - F_N: Applied normal force to the pull blocks. 	The remaining parameters serve as input variables for optimization process and is thus given as input variables for the machine learning (ML).

Due to the great number of parameters, and corresponding range of reasonable numerical values they can take, it is ... to parameter search including all of these. Thus, we will to a great extent rely on a reverse engineering in order to establish a set of parameters for the *physical* and *measurement* categories along with numerical ranges for the *ML input* category which gives stable and promising results. By doing so we effectively narrow down the parameter regime for which the investigated frictional properties belong. We aim to choose the parameters in order to accommodate a balance between generalizable and stable result which is simultaneously a suitable candidate as a proof of concept for the control of friction properties using kirigami inspired cuts.

In the following we present the results of the friction simulations in parallel to the procedure of investigating the choice of different parameters.

In the following subsections (X to Y) we are going to present the friction simulation results in parallel to the presentation of the reasoning behind the parameter choices. For this we will refer to the default parameter choice showcased in table 2.2 which is representative of the final parameter choices.

Table 2.2: Final parameters for the friction simulations Probably not the neatest format for this...

Physical	Measurements	ML input
$T = 300 \text{ K}$ $v_{\text{slide}} = 20 \text{ m/s}$ $K = \text{inf}$ (LAMMPS: <i>fix move</i>) Scan angle : $(x, y) = (0, 1)$	$dt = 1 \text{ fs}$ $t_R = 15 \text{ ps}$ Pauses = 5 ps Stretch speed = 0.01 ps^{-1} Slide distance = 400 \AA Sheet size = $130.029 \times 163.219 \text{ \AA}$ Pull block size = $2 \times 130.029 \times 15.183 \text{ \AA}$	Sheet configuration = Contiguous Stretch amount = Below rupture $F_N = [0.1, 10] \text{ nN}$

Say something about how these parameters are chosen. Reference to articles for which these were mirrored from.

2.1.1 Pressure reference for normal load domain

Find place to put this.

In order to relate the magnitude of the normal force in our friction measurement we will use the pressure as a reference. We will use the pressure underneath a stiletto shoe as a worst case for human pressure execution underneath the shoe. From (source 1) it is reported that the diameter of a stiletto heeled shoe can be less than 1 cm. Hence a 80 kg man³ standing on one stiletto heel (with all the weight on the heel) will result in a pressure

$$P = \frac{F}{A} = \frac{mg}{r^2\pi} = \frac{80 \text{ kg} \cdot 9.8 \frac{\text{m}}{\text{s}^2}}{(\frac{1 \times 10^{-2} \text{ m}}{2})^2\pi} = 9.98 \text{ MPa}$$

While this is in itself a spectacular realization that is often used in introductory physics courses (source 2) to demonstrate the rather extreme pressure under a stiletto heel (greater than the foot of an elephant) (how many Atmos?) this serves as a reasonable upperbound for human executed pressure. With a full sheet area of $\sim 21 \times 10^3 \text{ \AA}^2$ we can achieve a similar pressure of $\sim 10 \text{ MPa}$ with a normal force of

$$F_N = 10 \text{ MPa} \cdot 21 \times 10^{-17} \text{ m}^2 = 2.10 \text{ nN}$$

Of course this pressure might be insufficient for various industrial purposes, but with no specific procedure in mind this serves as a decent reference point. Notice that if we consider a human foot with area 113 cm^2 the pressure drops to a mere 70 kPa corresponding to $\sim 0.01 \text{ nN}$.

³Yes, a man can certainly wear stiletto heels.

2.2 Single friction simulation analysis

We begin by assessing the raw data for a single friction simulation run with the default parameters shown in table 2.2 for a non-cut sheet, no stretch and an applied normal force of 1 nN.

2.2.1 Force oscillations

We first assess the raw data for the friction force F_{\parallel} parallel to the drag direction as seen in figure 2.1. The sample rate is $10 \text{ ps}^{-1} = 100 \text{ timesteps}^{-1}$ for which each sample is the mean value of the 100 timesteps preceding the given sample interval. We observe immediately that the data carries oscillations on different time scales. By applying a savgol filter to the data with a polyorder of 5 and window length of 150 timesteps, corresponding to a sliding distance of 3 Å or a time window of 15 ps, we can qualitatively point out at least two different frequencies of oscillation. On figure 2.1a we see roughly three waves on the savgol filter corresponding to a relative high frequency, while on 2.1b the same savgol filter reveals a lower frequency on top of the first, creating the visual pattern of a wavepacket.

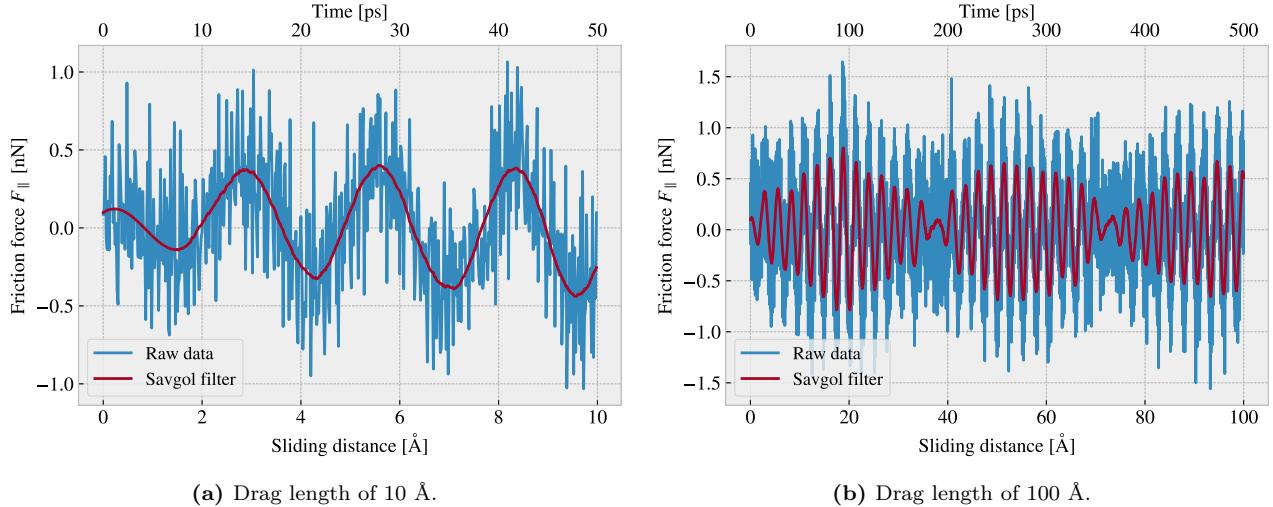


Figure 2.1: Friction force F_{\parallel} with respect to the drag direction between (full) sheet and substrate versus sliding distance. The sliding distance is measured by the constant movement of the virtual atom and not the COM of the sheet. However, we expect these measures to be fairly identical due to the fact that the pull blocks are rigidly coupled to the virtual atom. The red line represents a savgol filter with window polyorder 5 and window length of 150 timesteps (corresponding to a sliding distance of 3 Å or a time window of 15 ps).

By performing a Fourier Transform (FT) on the data we can quantify the leading frequencies as seen in figure 2.2a. By plotting the two most dominant frequencies $f_1 = 0.0074 \text{ ps}^{-1}$ and $f_2 = 0.0079 \text{ ps}^{-1}$ as $\sin(2\pi f_1) + \sin(2\pi f_2)$ we find a qualitatively convincing fit to the observed wavepacket shape as seen in figure 2.2b. By using the trigonometric identity

$$\begin{aligned}\sin(\alpha + \beta) &= \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta), \\ \sin(\alpha - \beta) &= \sin(\alpha)\cos(\beta) - \cos(\alpha)\sin(\beta),\end{aligned}$$

and decomposing $f_1 = a - b$, $f_2 = a + b$ we can rewrite the sine sum as the sinusoidal product

$$\begin{aligned}\sin(2\pi f_1)\sin(2\pi f_2) &= \sin(2\pi(a - b))\sin(2\pi(a + b)) \\ &= \sin(a)\cos(b) + \cancel{\cos(2\pi a)\sin(2\pi b)} + \sin(2\pi a)\cos(2\pi b) - \cancel{\cos(2\pi a)\sin(2\pi b)} \\ &= 2\sin(2\pi a)\cos(2\pi b),\end{aligned}$$

with

$$\begin{aligned}a &= \frac{f_1 + f_2}{2} = 0.0763 \pm 0.0005 \text{ ps}^{-1}, & b &= \frac{f_2 - f_1}{2} = 0.0028 \pm 0.0005 \text{ ps}^{-1}, \\ &= 0.381 \pm 0.003 \text{ Å}^{-1}, & &= 0.014 \pm 0.003 \text{ Å}^{-1},\end{aligned}$$

where the latter frequency is denoted with respect to the sliding distance. This makes us recognize the high oscillation frequency as a and the low frequency as b . The faster one has a period of $T_a = 2.62 \pm 0.02 \text{ \AA}^4$. This corresponds well with the magnitude of the lattice spacing and especially that of graphene at 2.46 \AA as expected theoretically (make reference to theory section?). We also take note of the longest period $T_b = 71 \pm 15 \text{ \AA}^{-1}$ which will be relevant for the evaluation of measurement uncertainty in section 2.3.

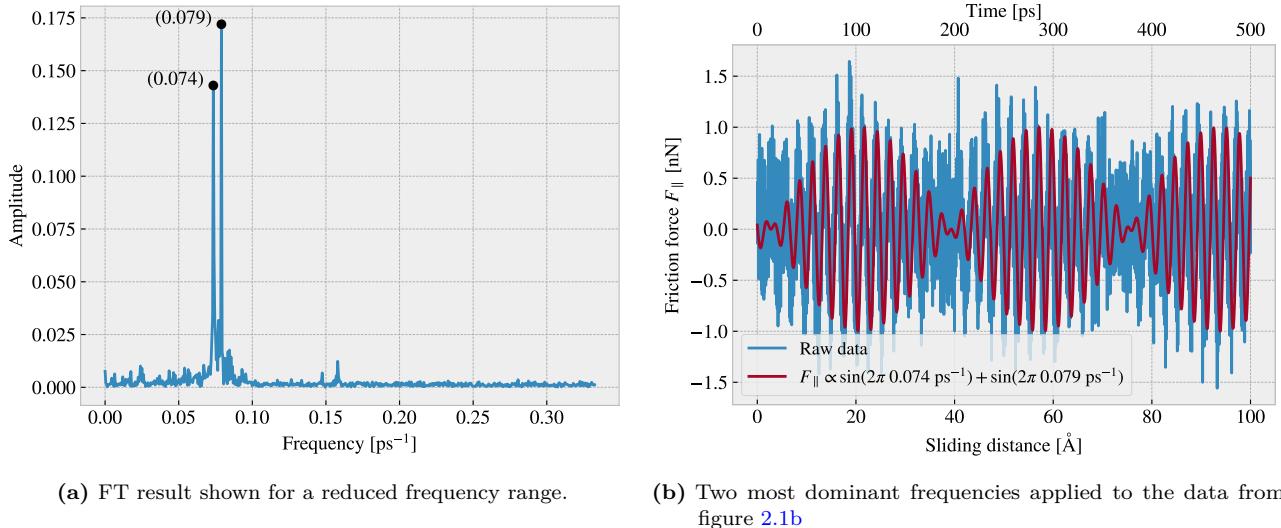


Figure 2.2: Fourier transform (FT) analysis of the full friction force data (all 400 \AA sliding distance) shown in figure 2.1. (a) shows the two most dominant frequency peaks. Note that no significant peaks were found in a higher frequency than included here. (b) shows a comparison between the raw data and the wavefunction corresponding to the two peaks in figure (a).

2.2.2 Decompositions

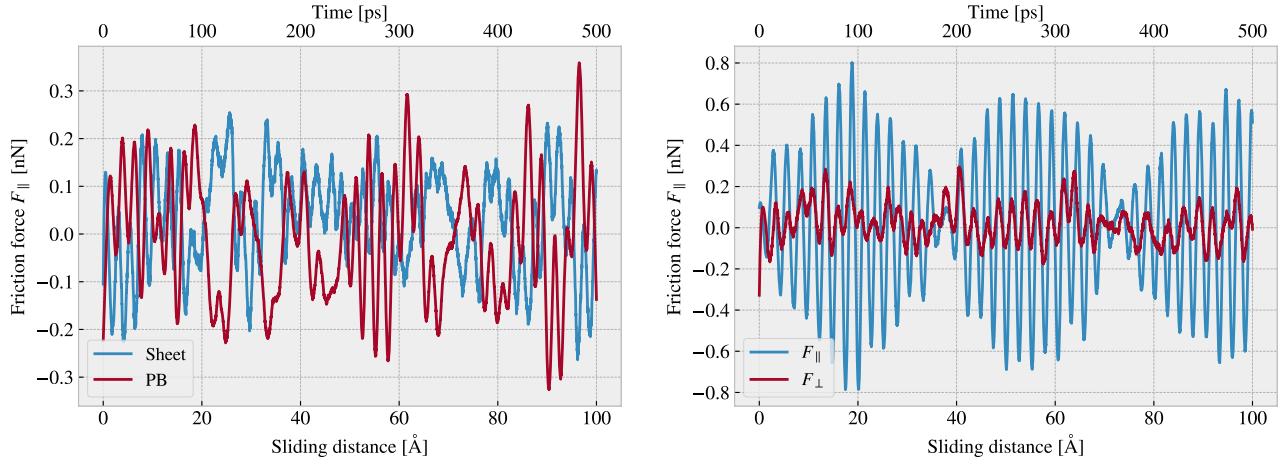
In the previous analysis we have looked only at the friction force for the full sheet, including the pull blocks which is locked off during sliding, and with respect to the drag direction. This represents our choice measurement which we will address in the following.

Due to the fact that we are only applying cuts to the inner sheet (excluding the pull blocks), it might seem more natural to only consider the friction on that part. If the desired frictional properties can be achieved by altering the inner sheet one can argue that any opposing effects from the pull blocks can be mitigated by scaling the relative size between the inner sheet and the pull blocks. However, when looking at the time series of the friction force decomposed with respect to the inner sheet and pull block region (see figure 2.3a), we observe the friction force arising from those parts are seemingly antisymmetric. That is, the distribution of the frictional pull from the substrate on the sheet is oscillating between the inner sheet and the pull block. Keeping in mind that normal force is only applied to the pull blocks we might take this as an integrated feature of the system which does not necessarily disappear when changing the spatial ratio between inner sheet and pull block. Any interesting friction properties might depend on this internal distribution of forces. Hence, we hedge our bets and use the full sheet friction force as a holistic approach to this measurement problem.

Similar we might question the decision of only considering the frictional force projected onto the sliding direction as we are neglecting the “side shift” induced during the slide phase. In figure 2.3b we see the decomposition into force components parallel F_{\parallel} and perpendicular F_{\perp} to the slide direction respectively. We see that the most dominant trend is projected into the parallel component. If we want to include the perpendicular component as well we would have to evaluate the friction as the length of the force vector for which we would lose the sign of the force direction. Hence, we would only get a positive contribution which would not be able to capture the change between resisting and assisting the sliding during stick-slip motion. One option to accommodate this is by using the vector length but keeping the sign from the projection parallel to the sliding direction. However,

⁴The uncertainty Δy is calculated as $\Delta y = \left| \frac{\partial y}{\partial x} \Delta x \right|$ for uncertainty Δx and $y(x)$

we omit such compromises as this might make analysis interpretation more difficult, and we use only the parallel component going forward.



(a) Decomposition into group inner sheet (Sheet) and pull blocks (PB). (b) Decomposition into parallel (F_{\parallel}) and perpendicular (F_{\perp}) to drag sliding direction.

Figure 2.3: Friction force decomposition on the data shown in figure 2.1 with applied savgol filters similar to that of figure 2.1b with window polyorder 5 and window length of 150 timesteps (corresponding to a sliding distance of 3 Å or a time window of 15 ps).

2.2.3 Center of mass path

From the previous observations of the friction force time series we see evidence of a stick-slip behaviour. Specially, we see in figure 2.3b that this might be the case both parallel and perpendicular to the sliding direction. By looking at the x, y -position for the sheet center of mass (COM) we observe the stick-slip motion manifested as a variation in COM speed combined with a side to side motion as shown in figure 2.4a. In an attempt to increase the magnitude of the slips we evaluate a similar simulation with spring constant $K = 30 \text{ N/m}$ (see figure 2.4b) in contrast to that of an infinite spring constant. While the maximum slip speed stays within a similar order of magnitude the slip length in the sliding direction is increased along with the side to side motion. Note that the axis scale is different between figure 2.4a and 2.4a. However, in both cases we observe that the side to side motion is associated with a low speed, meaning that is more reminiscent of a “slow” creep alignment with the substrate than a slip.

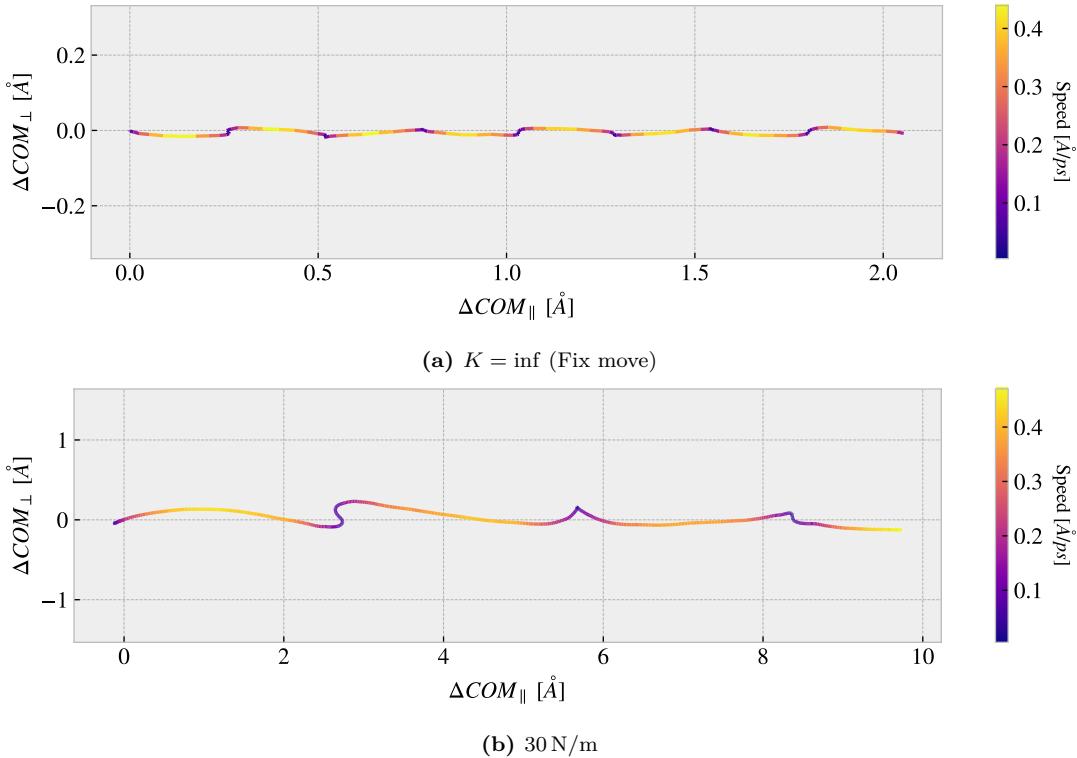


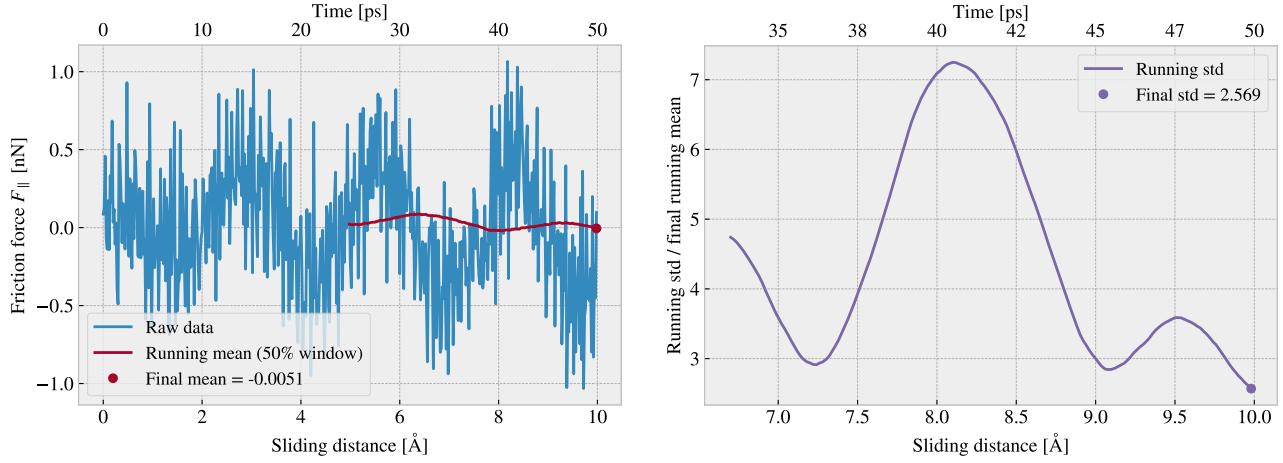
Figure 2.4: Center of mass position relative to the start of the sliding phase in terms of the direction parallel to the sliding direction ΔCOM_{\parallel} and the axis perpendicular to the sliding direction ΔCOM_{\perp} . The colorbar denotes the absolute speed of the COM.

2.3 Defining metrics for dynamic and static friction

In order to evaluate the frictional properties of the sheet we reduce the comprehensive friction force time series addressed in section 2.2 into single metrics describing the dynamic and static friction. The natural choice is to use the mean and max values of the time series.

2.3.1 Dynamic friction

For the dynamic friction measurement we take the mean value of the latter half of the dataset to ensure that we are sample from a stable system. For a full sliding simulation of 400 Å we thus base our mean value on the latter 200 Å of sliding. In figure 2.5a we have shown the friction force of the first 10 Å of sliding together with a running mean with window length of 5 Å corresponding to 50% the data length. This is merely done to illustrate the samplig procedure and by only using a 10 Å sliding distance the final mean estimate (indicated with a dot) takes a negative value due to the specific cut-off of the few osciallition captured here. Nonetheless, one approach to quanify the uncertainty of the final mean estimate is to consider the variation of the running mean preceeding the final mean value. The more the running mean fluctuates the more uncertainty associated with the final estimate. However, only the running mean “close” to the ending should be considered, since the first part will rely on data from the beginning of the simulation. From the Fourier analyse in section 2.2.1 we found the longest significant oscillation period to be $\sim 71 \text{ \AA}^{-1}$ corresponding to $\sim 35\%$ of the running mean window consisting of 200 Å of slifing when including all the data. Hence, we use the standard deviation of the final 35% of the running mean to approximate the uncertainty of the final mean value, and we estimate the relative error by dividing the standard deviation by the final mean value. In figure 2.5b we showcase a running standard deviation of a window length 35% the running mean window in figure 2.5a for the illustrative case of a total 10 Å slide. The final uncertainty value is marked by a dot, and we see as expected that we get a high relative error of $\sim 257\%$ which corresponds well with the short sampling period and the mean value taking an unphysical negative value.



(a) Running mean with window length 5 Å (50% the data length). (b) Running std with window length 1.75 Å (35% the mean window length.)

Figure 2.5: Running mean and running standard deviation (std) on the friction force data from a 10 Å of sliding simulation. The running mean window is 50% the data length while the running std window is 35% the running mean window length.

When including the full dataset of 400 Å of sliding, such that std window actually matches with the longest period of oscillations expected from the data, we get a final relative error of $\sim 12\%$ as shown in fig 2.6. This is arguable just at the limit for an acceptable error, but as we shall see later (Make a reference to fig or sec) this high relative error is mainly connected to the cases of low friction. When changing the simulation parameters, such that the mean friction evaluates to considerable higher values, the relative error drops to the order (put in numbers). One interpretation of this finding is simply that the oscillations in the running mean is somewhat independent of the magnitude of the friction. In that case, the relative error will spike for the low friction cases.

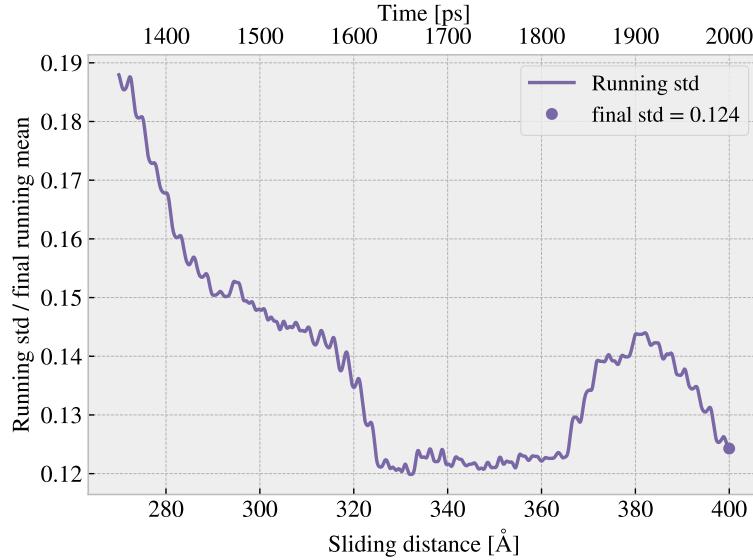


Figure 2.6: Running standard deviation (std) for a full 400 Å sliding simulation. The running std window is 70 Å (35% the running mean window of 50% the data length).

2.3.2 Static friction

The max value is the most obvious choice for addressing the static friction, even though that the definition of the static friction is a bit vague. When considering the friction force time series in figure 2.1 we observe that

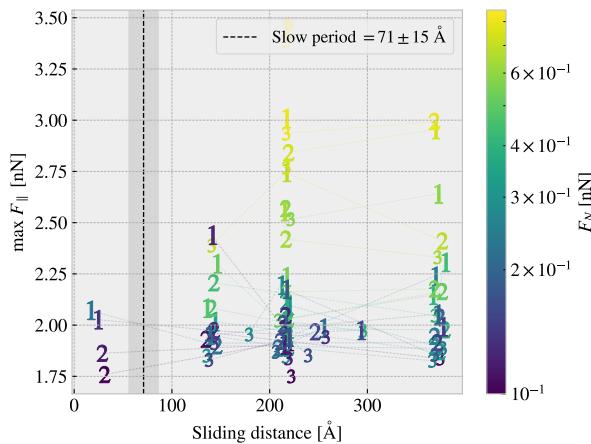


Figure 2.7: Distribution of top three max friction force peaks for 30 uniformly sampled normal forces $F_N \in [0.1, 10]$ nN. The dotted line and the grey area marks the slowest significant oscillation period found in the data and thus marking a dividing line for whether a peak falls within the “beginning” of the sliding simulation.

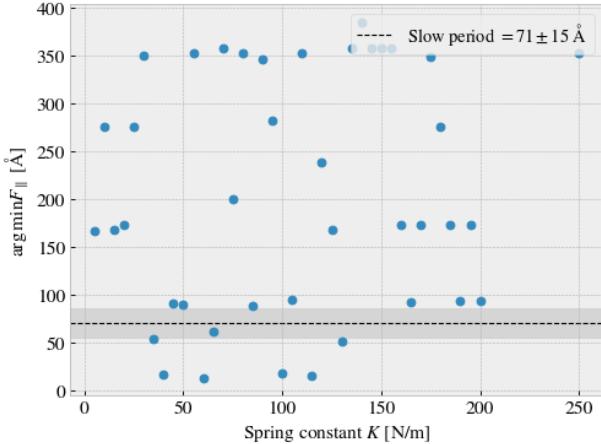


Figure 2.8: Sliding displacement for the max friction peak to appear as a function of spring constant. Fixmove is tmp mapped to $K = 200$ here without any discontinuous lines.

the stick-slip oscillations increase in magnitude toward a global peak at ~ 20 Å. Thus, we could identify this peak as the static friction force, but the global max does in fact rarely fall within the first part of the sliding. In figure 2.7 we investigate the top three max value, at which sliding distance they occur and at what magnitude, for 30 uniformly sampled normal forces in the interval [0.1, 10] nN. It is immediately clear that only few of the peaks fall within the “beginning” of the simulation defined by the slowest significant oscillation period of 71 ± 15 Å. In fact only 2/30 global values and 4/90 top three values can be associated to the start of the sliding by this definition. Thus, this result suggests that the max value cannot be used as a reliable measure for the static friction either due to its lack of presence or due to the simulation setup procedure. For a more typical evaluation of the static friction force one would increase force slowly until the first slip significant slip is recorded (a series of precursors is expected to precede this). In our simulations we drag the sheet relatively fast in a rigid manner which might be the reason for the lacking the static friction. Bonelli et al. [3] reported that the stick-slip behaviour was only presented when using a relatively soft spring. Thus, by changing the spring constant we investigate possibility to observe a static friction (**I kind of interchanged stick-slip and static friction int his argument, but I still think it can be used to argue for doing the test...**) response within the framework of our simulation procedure as shown in figure 2.8. However, the results do not indicate any implications that a recognizable domain exist for which the static friction response would be reliable. Hence, we will base the final assessment on frictional properties purely on the dynamic friction force.

2.4 Out of plane buckling

The out of plane buckling is the main motivation for applying the kirigami inspired cuts to the sheet. Thus, we perform a stretch simulation in a low temperature $T = 5$ K vacuum in order to verify that the chosen cut configurations do in fact contribute to a significant out of plane buckling when stretched. For the non-cut, popup and honeycomb configuration we assess the movement in the z-direction (perpendicular to the plane) during the stretch, which we visualize by the min and max z-value along with the atom count quartiles 1%, 10%, 25%, 50% (median), 75%, 90% and 99% as shown in figure 2.9. We observe that the popup and honeycomb pattern buckle considerably out of plane during the stretch in comparison to the non-cut sheet which only exhibit minor buckling of ~ 2 Å which is on the same order as the atomic spacing in the sheet. We also notice that the popup pattern buckles more in consideration to the min and max peaks while the 1%, 99% quartiles is on the same

magnitude as the honeycomb. By looking at the simulation visualization (**include OVITO figures for vacuum stretch as well?**) we can conclude that this is mainly due to the fringes of the sheet “flapping” around.

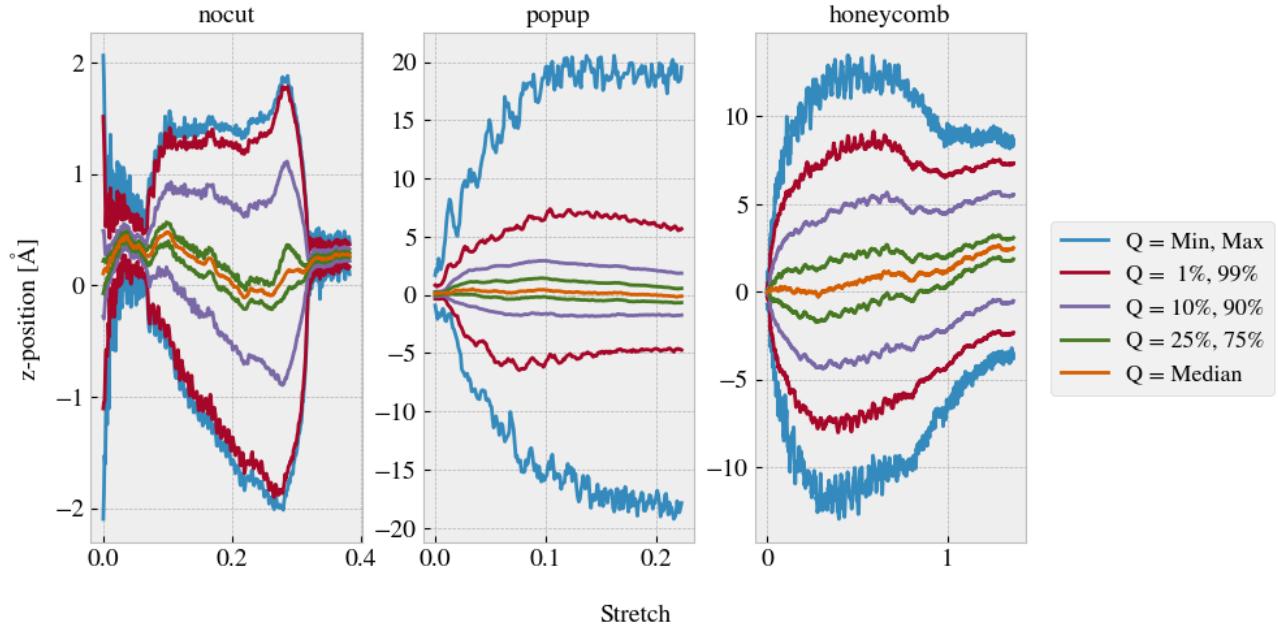


Figure 2.9: Out of plane buckling during stretch of sheets in vacuum at $T = 5$ K. Reading from left to right the vacuum rupture stretch are 0.38, 0.22 and 1.37. **perhaps use a color scale instead of the standard color cycles here.**

The next step is to verify that the buckling will lead to a significant altering of the contact area when the sheet is in put in contact with the substrate. We investigate this by simulating the stretch at the default temperature $T = 300$ K with the presence of contact forces between the sheet and substrate. Note that no normal load is applied as the sheet and substrate is sufficiently attracted by the LJ potential. Selected frames from the simulation is shown in appendix ???. We assess the contact area by the relative amount of atoms in the sheet within chemical range of the substrate. The cut-off for this interaction is 4 Å corresponding to $\sim 120\%$ the LJ equilibrium distance. Since the contact area is usually calculated as the amount of atoms in contact multiplied with an associated area for each contact this feature is taken to be proportional to the contact area. The relative amount of bonds as a function of stretch for the various configurations is shown in figure 2.10 which clearly indicates a drop in contact area as the cutted sheets are stretched.

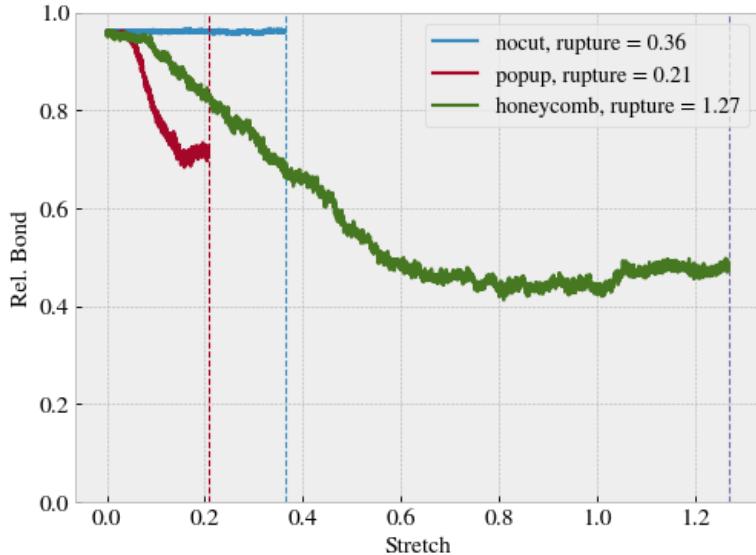


Figure 2.10: Contact vs. stretching of the sheet, where the contact is measured by the relative amount atoms in the sheet within chemical interaction range to the substrate. The cut-off for this interaction range is 4 Å corresponding to $\sim 120\%$ the LJ equilibrium distance. $T = 300$ K

Compare figure 2.10 to that of figure 2.15 where multiple simulations constitute the stretch-contact curve.

2.5 Investigating selected parameters

We investigate the importance of the physical variables T , v_{slide} and K (make plots for scan angle as well?) and the choice of timestep dt . This is done partly understand how the dependencies relate to theoretical, numerical and experimental results, and partly to understand how these parameter choices defines the regime for our multi configurational search. We use the default parameters in table 2.2 with exception of the single parameter of interest which is varied in a reasonable range of the default choice. In figure 2.11-2.14 the dynamic friction estimate and the max friction force is shown as a function of T , v_{slide} , K and dt respectively. For the dynamic friction estimate the absolute error is denoted by a shaded error which linearly connects the points.

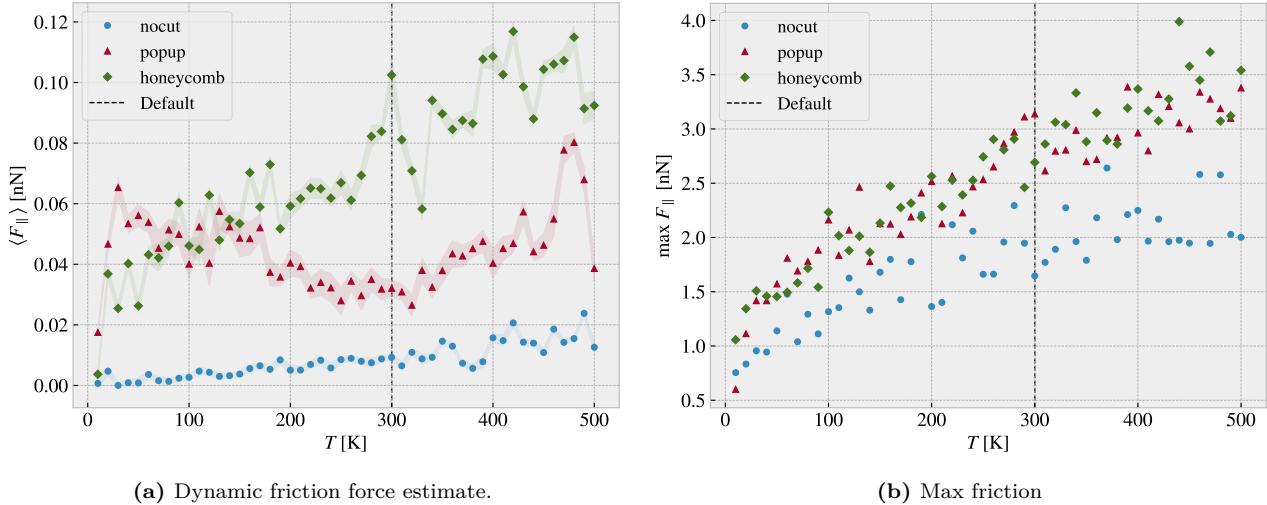


Figure 2.11: Temperature.

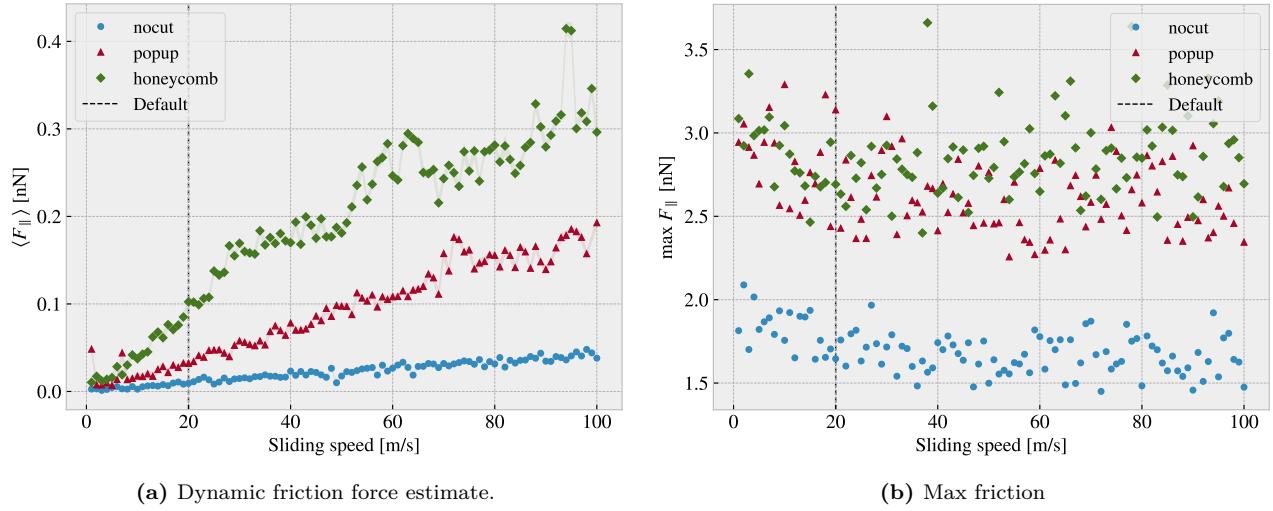


Figure 2.12: Sliding speed

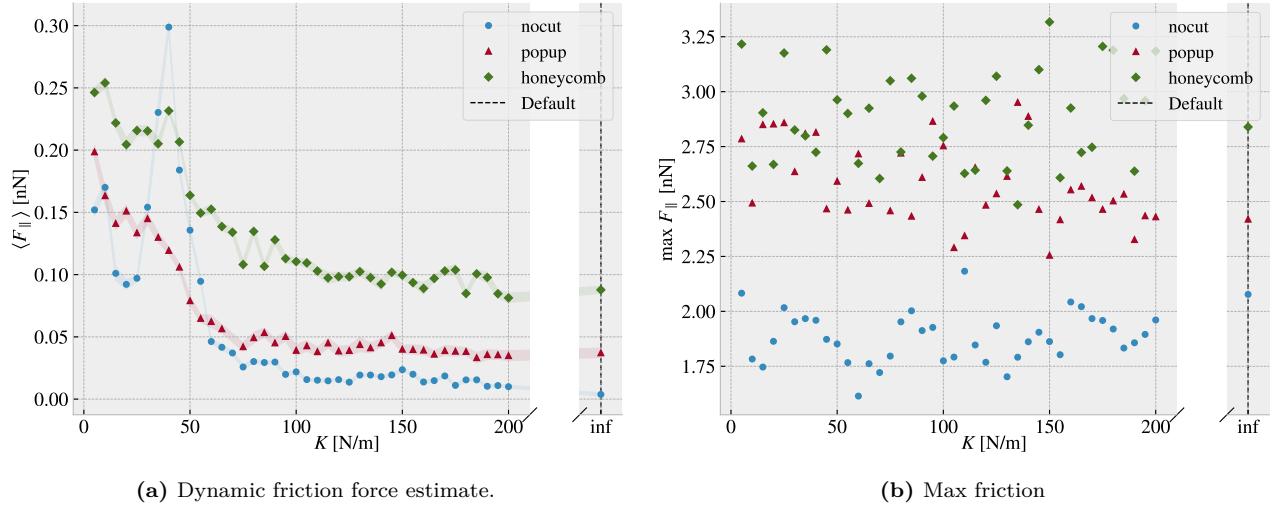


Figure 2.13: Spring constant

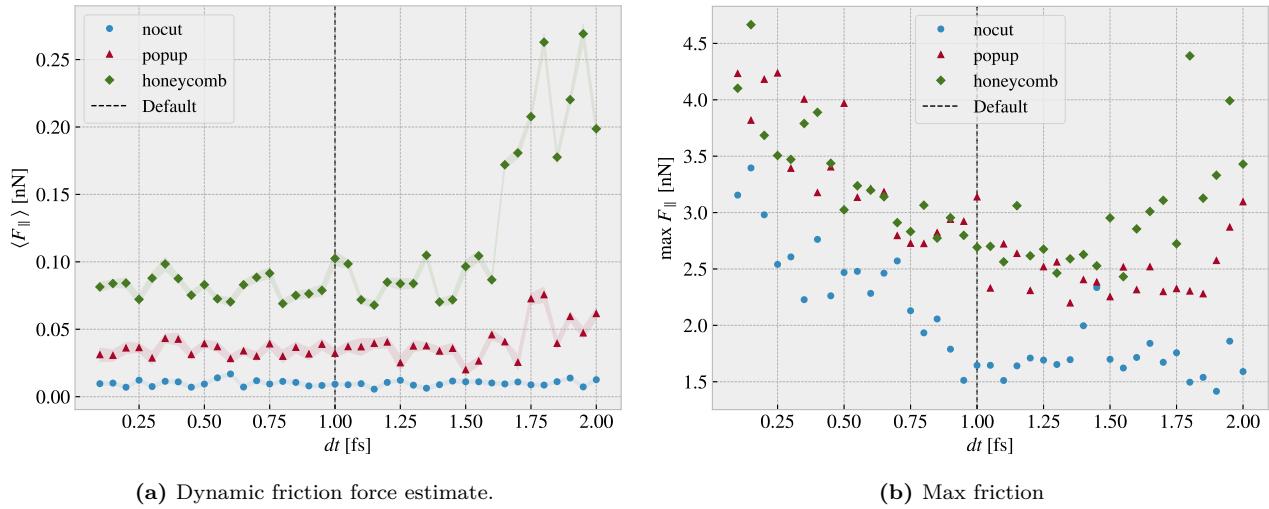


Figure 2.14: Timestep

Quick thoughts:

- Temperature: We do clearly not see the $1/T$ temperature decrease. The non-cut sheet seems to showcase a linear relationship which is also somewhat present for the honeycomb which matches some of the findings in other MD simulations. For the popup we do see a local decrease at low temperatures which flip at around the default $T = 300$ K temperature. The max friction peaks seem to increase with temperature as well indicating that the peaks might be associated with thermal fluctuations rather than actual stick-slip behaviour. This supports the finding that the static friction response is not significantly present in these simulations.
- Velocity: Considering the non-cut sheet first the velocity dependency is seemingly linear which deviates from the expected logarithmic trend. For the cutted configurations we find some peaks which might indicate the presence of resonance frequencies. The cutted sheet might be closer to a logarithmic trend, but this is not spot on either. The max friction seems to decrease slightly with small velocities and then stay rather constant. This can probably be explained by the reduced time to stick between stick slip.
- Spring constant: On all three configurations the dynamic friction decreases with an increasing spring constant. The best explanations might be due to the lack of freedom to “get stuck” in incommensurable configurations. We also notice that the friction varies a lot at lower spring constants supporting the choice of having a stiff spring for stability reasons. Especially the non-cut sheet peaks at $K = 40$ N/m. The max friction seem to be constant with K .
- dt : The dynamic friction is relatively stable around the default choice of $dt = 1$ fs. However, the fluctuations with respect to dt is more significant for popup pattern and even more for the honeycomb pattern. This indicates that the more complex dynamics of the simulation is more sensitive to the timestep. We might interpret this information as an additional measure of uncertainty. The maximum friction decreases with increasing timestep which can be asserted a statistical interpretation: Higher peaks will be captured by the high resolution of a low dt and vice versa. The high max values towards the point of $dt = 2$ fs is most likely due to the approach of instability in the simulation as seen more clearly for the dynamic friction evaluation.

2.6 Normal force and stretch dependencies

Till this point we have only changed variables one by one to investigate single dependencies. We now advance the study to a simultaneous variation of stretch and normal force.

Explain how the stretch is uniformly sampled within equally divided intervals and the normal force is actually uniformly sampled in a given range. Argue that the first might be approximately uniformly distributed for large numbers.

Talk about rupture test also. Maybe in the theory/method section under numerical procedure: Before simulating a rupture test is perform to determine under what stretch the sheet ruptures. This is a slightly higher threshold than when applied normal load and sliding along the substrate.

2.6.1 Contact area

??

We reproduce the contact area investigation of figure 2.10 with the modification that the contact count is measured as an average of the latter 50% of the sliding simulation at a non-zero applied normal load. The results are shown in figure 2.15 with 30 attempted (some rupture) stretch (pseudo) uniformly distributed stretch between 0 and the rupture point and 3 uniform distributed normal loads in the interval [0.1, 10] nN.

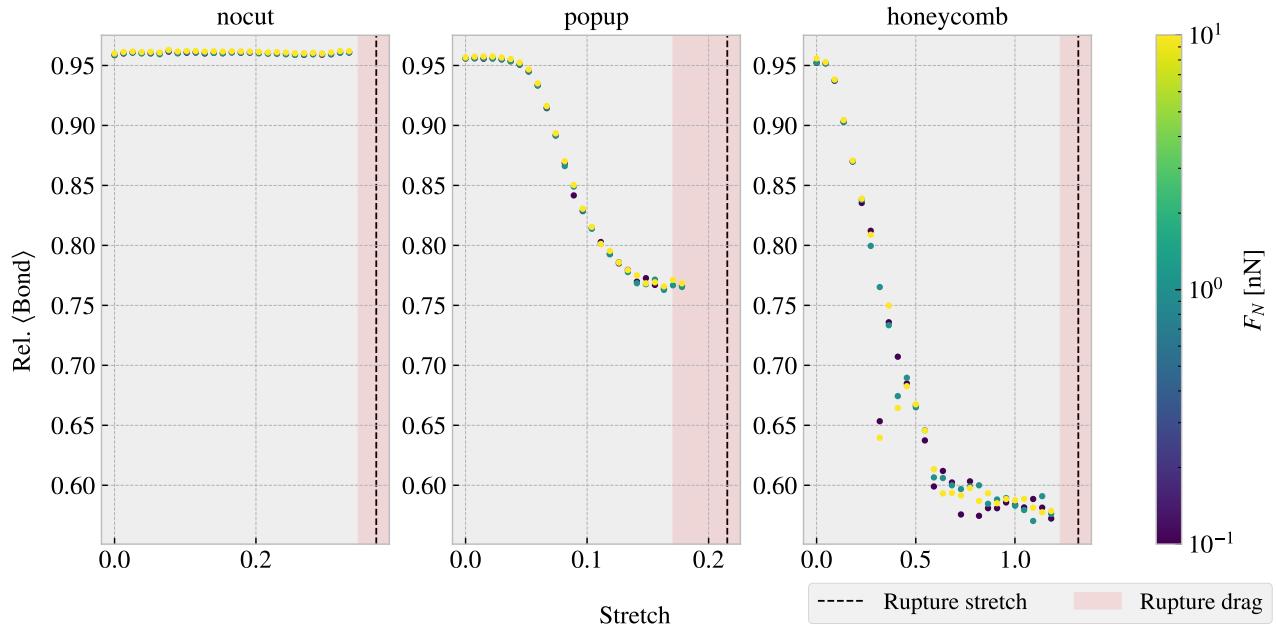


Figure 2.15: Average relative amount of bonds between the sheet and the substrate defined by the cut-off distance of 4 Å. The average is taken over the latter half of the sliding phase. The red shade denotes the stretch range where ruptures occur at certain normal loads under sliding while the black-dotted line represents the rupture point due to stretching (rupture test)

From figure 2.10 we observe a significant decrease in the contact due to stretching of the cut configurations in contrast to the non-cut which stays roughly constant. This is reminiscent of the non-sliding stretch vs. contact curve shown in figure 2.10. Given these results, theoretically one would expect the dynamic friction to decrease with stretch for the cut configurations.

2.6.2 Stretch

We make a similar analysis as done in the previous section ?? with the substitution of friction force instead of contact (The data is taken from the same simulations runs). The dynamic friction force (put uncertainty here even though that it is quite low?) and the max friction is shown in figure 2.16a and 2.16 respectively.

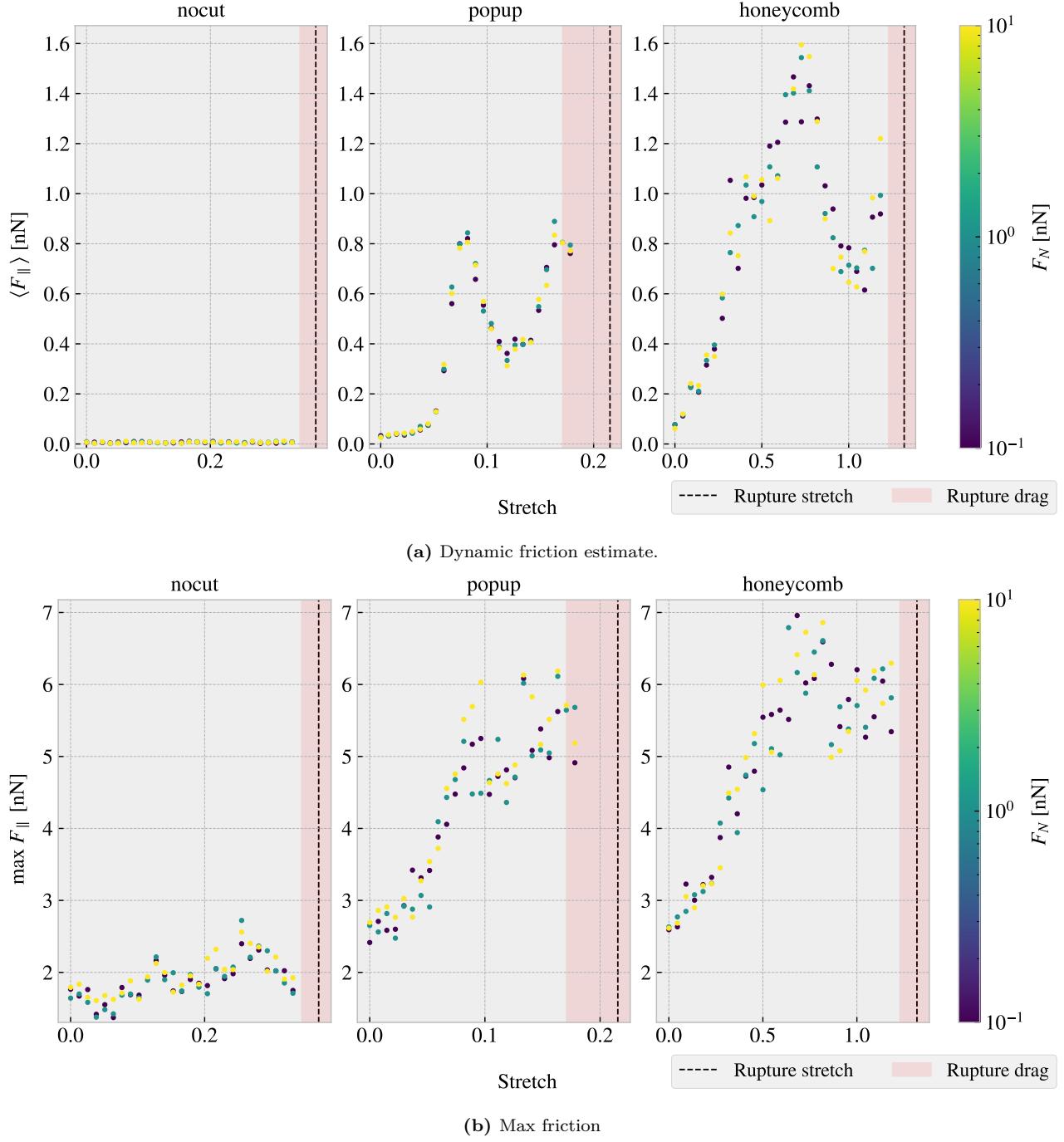


Figure 2.16: CAPTION

From figure 2.16a we find to our surprise that the dynamic friction increase with stretch for the cut configurations despite a simultaneous decrease in contact area as shown in figure 2.15. This suggests that the amount of chemical bonding atoms is not the dominant mechanism for the friction of this system. Instead, we might point to a mechanism more mechanical of nature associated to phonon excitations. When the cut sheet is stretched the stress (show stress maps somewhere or not necessary?) might induce a certain distribution and magnitude of point pressures to favor energy dissipation. Nonetheless, the results showcase a strong coupling between stretch and friction force, also for the max friction force, which is beyond the expectations at this stage of the study. The non-cut configuration does not show significant dependency on the stretch which reveal that this effect is only present when combining cut and stretch and not purely by stretching the sheet.

By considering the increase in dynamic friction towards the first peak we get a relative friction increase and increase vs. stretch ratios as described in table 2.3. While the honeycomb force increase towards the first peak is approximately linear the popup exhibits seemingly exponential growth which yield a slope on the order ~ 30 nN.

Table 2.3: (stretch, dynamic friction) coordinates from figure 2.16a at start and the first peak respectively used to approximate the relative increase in friction force and the ratio for friction increase vs. stretch for sait range. In practice the latter ratio denotes the slope of a forced linear trend.

Configuration	Start	First peak	Relative increase	Friction force vs. stretch ratio [nN]
Popup	$\sim (0, 0.03)$	$\sim (0.082, 0.83)$	27.7	9.76
Honeycomb	$\sim (0, 0.07)$	$\sim (0.728, 1.57)$	22.4	2.06

Additionally, we notice that both the popup and honeycomb also exhibits stretch ranges where the dynamic friction force decrease with increasing stretch. Qualitatively we assign the slope to be on the same order of magnitude as those towards the first peak. This is useful for the prospect of taking advantage of this phenomena as we can essentially achieve both higher and lower friction for increasing stretch for different starting points.

2.6.3 Normal force

Main take away from this section should be that the normal force does not really change the friction much; The friction coefficient is extremely low, but I'm not sure how well the linear fits are (whether they are linear or sublinear). Not sure if I should do a linearly increasing normal force for better linear plots?

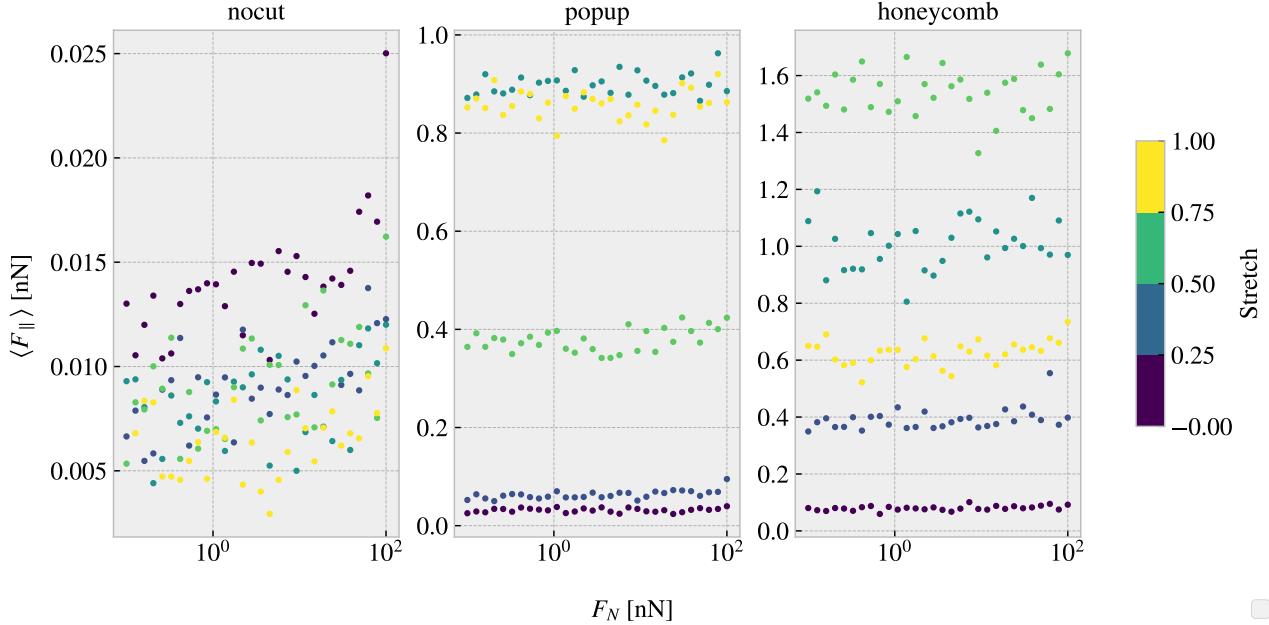


Figure 2.17: ...

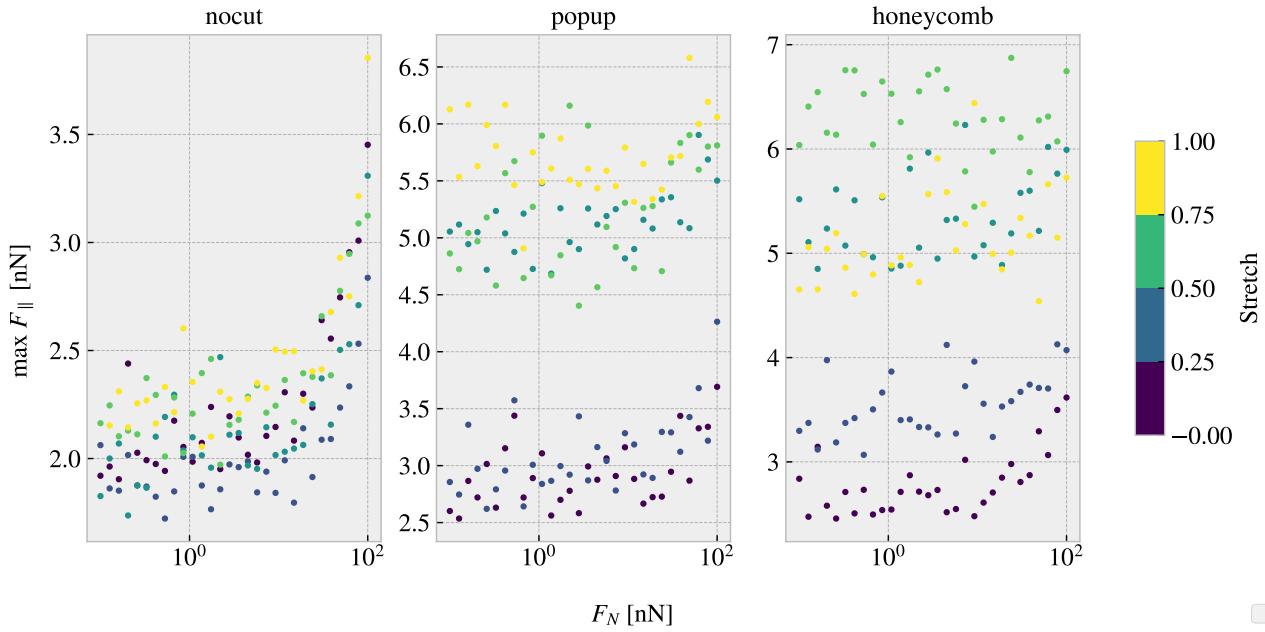


Figure 2.18: Colorbar is only fitted for the right plot (honeycomb)... this should be fixed. Should I have run a linear distribution of FN so I could plot it linear here also...?

Table 2.4: Mean friction coeff

nocut	$0.00009 \pm 1 \times 10^{-5}$	$0.00005 \pm 1 \times 10^{-5}$	$0.00004 \pm 1 \times 10^{-5}$	$0.00005 \pm 2 \times 10^{-5}$	
popup	$0.00005 \pm 3 \times 10^{-5}$	$0.00024 \pm 5 \times 10^{-5}$	$0.0002 \pm 2 \times 10^{-4}$	$0.0005 \pm 1 \times 10^{-4}$	$0.0003 \pm 2 \times 10^{-4}$
honeycomb	$0.00013 \pm 6 \times 10^{-5}$	$0.0006 \pm 3 \times 10^{-4}$	$0.0004 \pm 6 \times 10^{-4}$	$0.0007 \pm 6 \times 10^{-4}$	$0.0009 \pm 3 \times 10^{-4}$

Table 2.5: Max friciton coeff

nocut	$0.0139 \pm 9 \times 10^{-4}$	$0.0083 \pm 7 \times 10^{-4}$	$0.010 \pm 1 \times 10^{-3}$	$0.0105 \pm 9 \times 10^{-4}$	
popup	$0.007 \pm 2 \times 10^{-3}$	$0.010 \pm 2 \times 10^{-3}$	$0.007 \pm 2 \times 10^{-3}$	$0.009 \pm 3 \times 10^{-3}$	$0.006 \pm 2 \times 10^{-3}$
honeycomb	$0.010 \pm 1 \times 10^{-3}$	$0.007 \pm 2 \times 10^{-3}$	$0.007 \pm 3 \times 10^{-3}$	$0.000 \pm 3 \times 10^{-3}$	$0.004 \pm 3 \times 10^{-3}$

One theory for the low friction coefficient might depend on the fact that the normal force is only applied on the pull blocks. Especially with the cut sheet the tension drops such that the effective normal force on the inner sheet is not changing very much. By this theory the friction force vs. normal force on the pull blocks should look a bit more like expected and we might make some plots of those to check

When looking at the graphs for the PB the max friction is visually textbook linear, while the mean friction is a bit more linear but also with negative coefficients...

2.7 Computational cost

Talk about the computational cost of different choices. How does computation time scale with drag speed, dt and maybe T and K as well. One could also mention scaling with system size.

Show how the number of cores per simulation scale to argue that running on just one core (maybe 4) is smart for the next step of many simulations.

Mention the trouble with GPU to show that this was considered, and in fact this was the reason for choosing the Tersoff potential over the AIREBO which is perhaps more common these days...

Chapter 3

Generating data

Present the configuration and variable choices for the generated dataset. Perhaps include appendix with all the configurations shown in a grid

3.1 Rupture

Better word might be detached and nondetached as used by Hanakate in Accelerated search article.

Chapter 4

Training forward network (Data analysis?)

4.1 Dataset analysis

A summary of the data points is given in table ??

Table 4.1: Summary of the number of generated data points used for machine learning. Due to random stretch values not all submitted simulations makes it through the framework to become a data point ([explain this somewhere](#)) Notice that the Tetrahedon (7, 5, 2) and Honeycomb (2, 2, 1, 5) from the pilot study is rerun as a part fo the Tetrahedon and Honeycomb dataset seperately. In the latter the reference point for the pattern is randomized and thus theese configurations is included twice. This is the reason for the total configuration count summing up to two less than otherwise expected.

Type	Configurations	Submitted data points	Final data points
Pilot study	3	270	261
Tetrahedon	68	3060	3015
Honeycomb	45	2025	1983
Random walk	100	4500	4401
Total	214 (216)	9855	9660

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, μ the mean value and σ 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 4.1

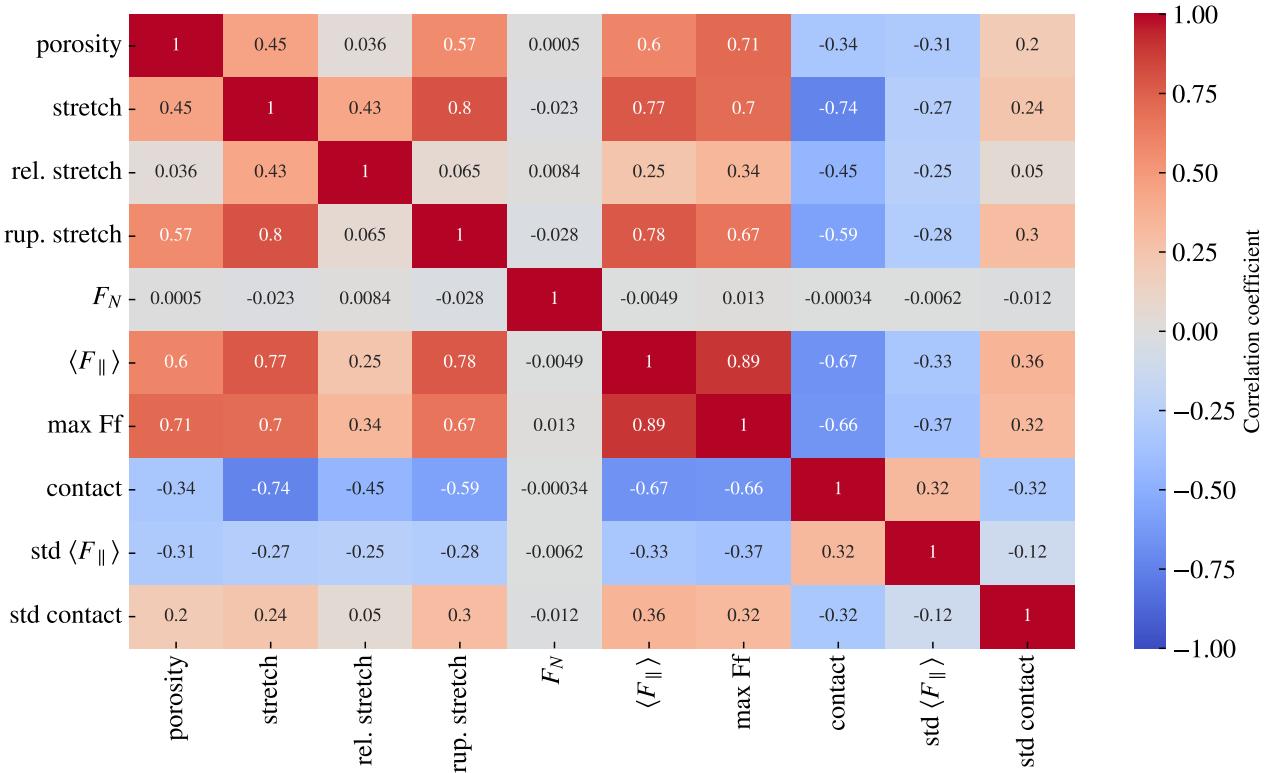


Figure 4.1: Pearson product-moment correlation coefficients for the full dataset (see table 4.1).

From figure 4.1 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 4.1 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 2.15 and 2.16a) 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 2.16a) where the relationship was partwise characterized by a positive 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 4.1.

In figure 4.2 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.

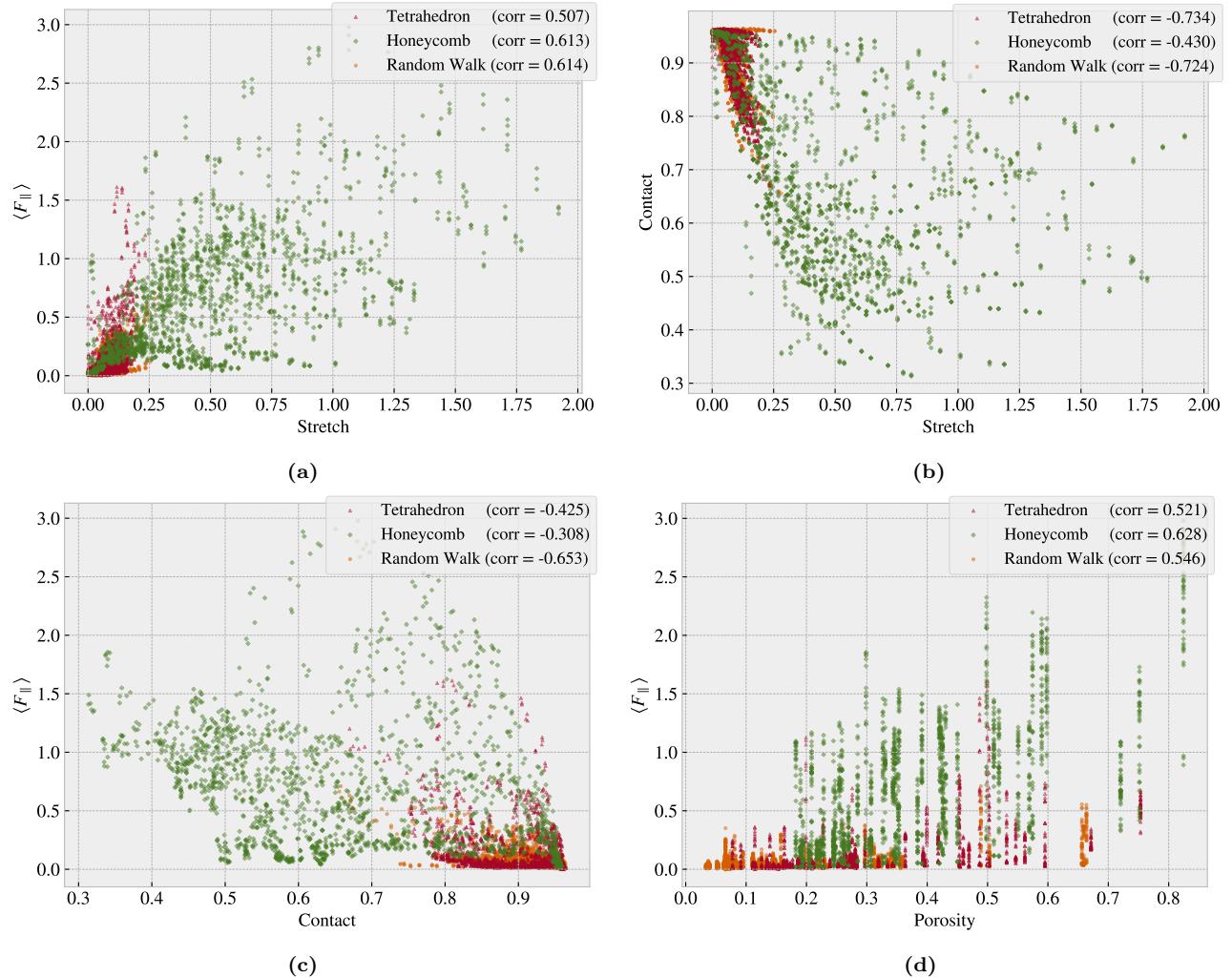


Figure 4.2: Scatter plot of the data sets Tetrahedron, Honeycomb and Random Walk (excluding the pilot study) for various variable combinations in order to visualize some chosen correlations of interest and distributions in the data

Chapter 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

5.1 Markov-Chain Accelerated Genetic Algorithms

5.1.1 Talk about traditional method also?

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 this 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 $P_{00}(j, t) = n_0(j, t)$

Chapter 6

Negative friction coefficient

6.1 Manual coupling of normal force and stretch

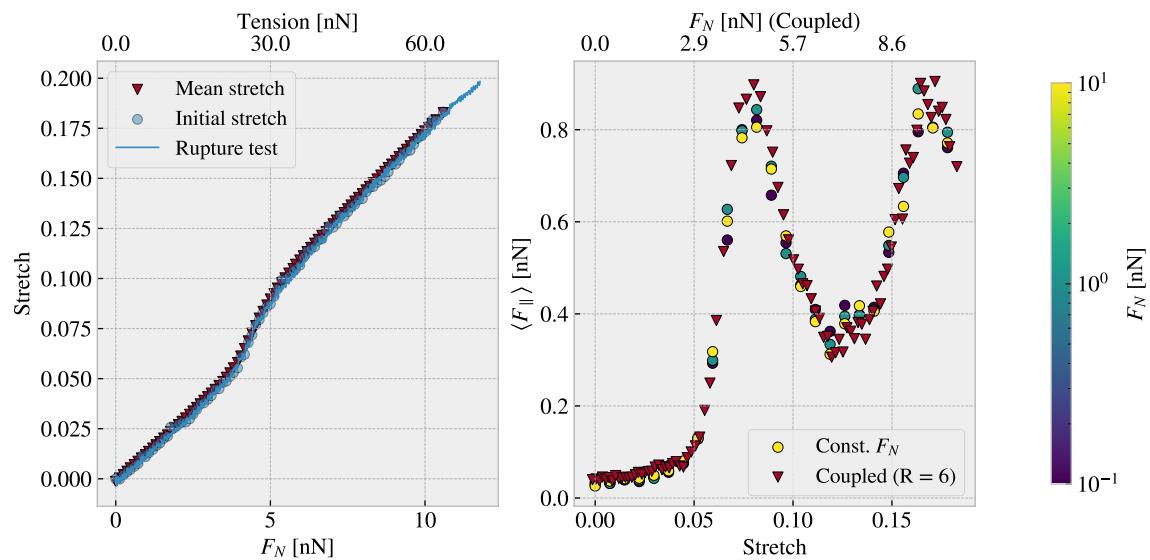


Figure 6.1

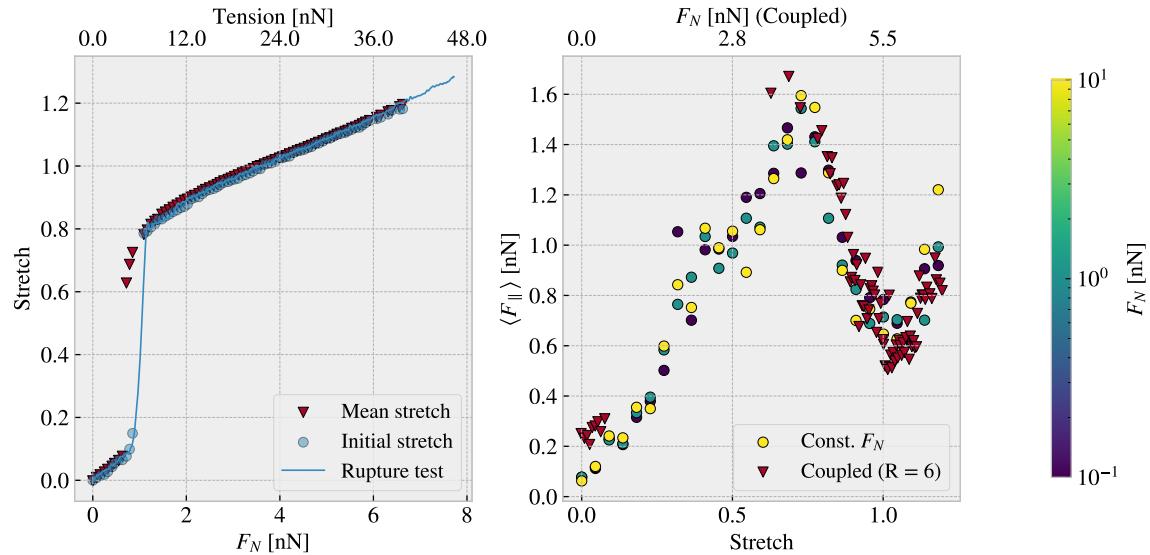


Figure 6.2: Fetch last data and update figure. Can we got more points in the sparse area?

6.2 Nanomachine coupling

Attempt to couple normal force and stretch by crossed carbon nanotube (CNT) contraption [6.3](#).

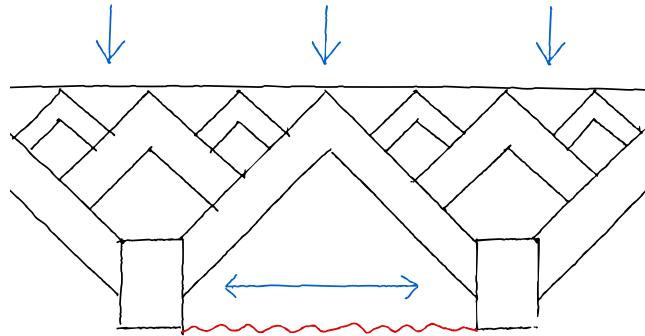


Figure 6.3: Working sketch for nanomachine

Appendices

Appendix A

Bibliography

- [1] L. Burrows, “New pop-up strategy inspired by cuts, not folds.”
- [2] “Scotch cushion lock protective wrap.”
- [3] F. Bonelli, N. Manini, E. Cadelano and L. Colombo, *Atomistic simulations of the sliding friction of graphene flakes*, .