

Syracuse University

**SURFACE**

---

Dissertations - ALL

SURFACE

---

August 2018

## Statistical Mechanics of thermalized ribbons and sheets

SOURAV SANJUKTA BHABESH

*Syracuse University*

Follow this and additional works at: <https://surface.syr.edu/etd>



Part of the [Physical Sciences and Mathematics Commons](#)

---

### Recommended Citation

BHABESH, SOURAV SANJUKTA, "Statistical Mechanics of thermalized ribbons and sheets" (2018).

*Dissertations - ALL*. 910.

<https://surface.syr.edu/etd/910>

This Dissertation is brought to you for free and open access by the SURFACE at SURFACE. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE. For more information, please contact [surface@syr.edu](mailto:surface@syr.edu).

# Abstract

In this dissertation, we study the thermal behavior of two-dimensional sheets and ribbons. We study the effects of thermal fluctuations on the crumpling transition of elastic sheets. Existing two-dimensional sheets have a crumpling temperature which is very high and the crumpling transition has not been observed experimentally. We propose a mechanism using which one can tune this crumpling transition by changing the shape and geometry of the sheet. We perform extensive molecular dynamics simulations by perforating the sheet with a dense array of holes and find that the critical temperature is a function of the removed area. Lastly, we look at clamped thermalized ribbons and study thermalized Euler buckling. Again, we perform molecular dynamics simulations by clamping one end and allowing the other end to slide to get the projected thermal length of the ribbon. We compress this system and observe a distinct two-state dynamics of the center of mass along with thermalized Euler buckling.

# Statistical Mechanics of thermalized ribbons and sheets

by

Sourav Sanjukta Bhabesh

B.Tech. Indian Institute of Technology Madras, 2010

MS Physics, Syracuse University, 2015

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy in Physics

Syracuse University

August 2018

Copyright © Sourav Sanjukta Bhabesh, August 2018

All rights reserved

*Dedicated to my parents and Deepika*

## Acknowledgements

I would like to take this opportunity to express my gratitude and thank all professors, colleagues, and friends whose support and guidance helped me throughout my Ph.D. First and foremost I would like to thank my research advisor Prof. Mark Bowick. Without his guidance and encouragement, research done by me here would not have been possible. I am indebted to him for allowing me the freedom to pursue my passion for computational work and advising me throughout my research work.

I would also like to thank other members of my defense committee - Cristina Marchetti, Carl Rosenzweig, Jennifer Schwarz, Joseph Paulsen and Teng Zhang - for reading and critiquing my work.

I am also grateful to David Yllanes, for helping me with a lot of what I learned during my research work at Syracuse. His mentorship and advice was valuable in shaping my research work. To my group members Suraj Shankar, Francesco Serafin, and Michael Moshe with whom I had many fruitful discussions. I am also thankful to Prof. David Nelson, whose insights into my research work helped me immensely.

I am grateful to Syracuse University faculty who taught me courses which helped me conduct my research. I would also like to thank the Syracuse Soft and Living Matter Group for the support and travel grants which helped me present my work at conferences. I am also thankful to Eric Sedore and Larne Pekowsky who helped me with computing resources.

Also, none of this would have been possible without the blessings and support of my parents and family. I would like to thank my wife Deepika, who has been patient with me and supported me throughout the last year that we have been together. To Raghav Govind Jha, Prashant Mishra, Nouman Butt, Prayush Kumar, Swetha Bhagwat, Michael Czajkowski and Adam Patch who have become

valuable friends and helped me innumerable times. To Patricia Whitmore whose help was always there with regards to academic paperwork and TA duties.

Finally, I am also indebted to the support from my professors in my undergraduate at IIT Madras - Neelima Gupte, Suresh Govindarajan, and Lakshmi Bala, who encouraged me to pursue my Ph.D. in the first place.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Thermal crumpling of perforated two dimensional sheets</b>	<b>13</b>
2.1	Introduction . . . . .	13
2.2	Model . . . . .	13
2.2.1	Dense arrays of holes . . . . .	16
2.3	Finite-size scaling . . . . .	22
2.4	Methods . . . . .	28
2.5	Conclusion and Discussion . . . . .	28
<b>3</b>	<b>Anharmonic effects and buckling in thermalized ribbons</b>	<b>30</b>
3.1	Introduction . . . . .	30
3.2	Model . . . . .	30
3.3	Tension free clamping of ribbons . . . . .	33
3.4	Effects of double clamping . . . . .	34
3.4.1	Bending Rigidity renormalization . . . . .	37
3.4.2	Two State System . . . . .	42
3.4.3	Time Autocorrelation function . . . . .	42
3.4.4	Integrated Survival Time . . . . .	45
3.4.5	Thermalized version of Euler buckling . . . . .	47
3.5	Conclusion and Discussion . . . . .	53
3.6	Appendix A . . . . .	53



## CONTENTS

3.7	Appendix B . . . . .	55
<b>4</b>	<b>Simulation methods</b>	<b>56</b>
4.1	Introduction . . . . .	56
4.2	Molecular Dynamics . . . . .	56
4.2.1	Derivation of Bending Force . . . . .	57
4.2.2	Derivation of bond harmonic force . . . . .	64
4.2.3	Noose Hoover Algorithms . . . . .	64
4.3	CUDA C GPU implementation . . . . .	65
4.3.1	Kernels . . . . .	65
4.4	HOOMD Blue package . . . . .	66
4.5	Visual Molecular Dynamics package . . . . .	67
4.6	Monte Carlo Implementation . . . . .	67
4.6.1	Metropolis method . . . . .	68
<b>5</b>	<b>Simulation Data Analysis</b>	<b>70</b>
5.1	Introduction . . . . .	70
5.2	System Equilibration time . . . . .	70
5.2.1	Integrated autocorrelation time . . . . .	71
5.2.2	Exponential time . . . . .	71
5.3	Error Estimation . . . . .	72
5.3.1	Jack Knife . . . . .	72
5.3.2	Bootstrap . . . . .	73
	<b>Bibliography</b>	<b>74</b>

# List of Figures

1.1	An electron micrograph of a portion of erthrocyte cytoskeleton. The skeleton has been spread to a surface area nine to ten times as large as in its natural state [24]. . . . .	3
1.2	Isolated RBC skeletons viewed under video-enhanced differential interference contrast microscopy. A and B show different closeups [25]. . . . .	4
1.3	A planar section of Boron Oxide $B_2O_3$ [30]. . . . .	7
1.4	Graphene is an atomic-scale hexagonal lattice made of carbon atoms - Wikipedia. . . . .	8
1.5	a. Paper model of spring. b and c. Grapehene spring [34]. . . . .	10
1.6	Graphene rectangular spring [34]. . . . .	11
2.1	Close ups of sections of the membrane for two different arrays of perforations. We begin by considering a full triangulated sheet and then remove all the nodes in a radius $R$ around its center. This central hole is then repeated periodically throughout the membrane. In the figure the removed nodes are represented by blue dots ( <i>Left: <math>R = 1</math>, Right: <math>R = 2</math></i> ). In the rest of the paper we will consider patterns of perforations with $R = 1, 2$ and varying spacing between holes (see Figure 8 in the Supplementary Material for a full description of all our perforation patterns). . . . .	15

## LIST OF FIGURES

2.2	Snapshots of thermalized configurations. We superimpose the configurations of a pristine sheet (blue) and a perforated sheet with holes of size $R = 2$ , in the pattern of Fig. 2.1-b (red) for two values of the temperature (in units of the bending rigidity $\tilde{\kappa}$ ). In both cases the full sheet is well into the flat phase, the thermal fluctuations causing just some wrinkling and oscillation. This 10% increase in temperature, however, triggers a crumpling of the perforated sheet. Both systems have size $L = 100a$ . . . . .	16
2.3	Radius of gyration as a function of $\tilde{\kappa}/kT$ for several patterns of perforation. The black curve is the baseline unperforated membrane, which crumples at the highest temperature, $\tilde{\kappa}/kT_c \approx 0.18$ . The red curves (A to D) are for triangular arrays of perforations of radius $R = 2$ with decreasing spacing between individual holes. The blue curves (1 to 3) are arrays of perforations with radius $R = 1$ . As the spacing between holes is reduced, the crumpling temperature decreases. The full description of the perforation patterns A–D and 1–3 is given in Supplementary Fig. 1. Data for systems of size $L = 100a$ . . . . .	17
2.4	Location of the crumpling temperature. We plot the specific heat $C_V$ (right axis) and $\tilde{\kappa}$ -derivative of the radius of gyration (left axis) as a function of $\tilde{\kappa}/kT$ for our most perforated system (corresponding to curve D in Fig. 2.3). The inset shows the analogous plot for a less perforated sheet (corresponding to curve B in Fig. 2.3), with a much sharper transition (note the different vertical scales of the axes). All error bars represent the standard error of the mean. . . . .	19

LIST OF FIGURES

2.5 Crumpling temperature  $kT_c$  as a function of the fraction  $s$  of removed area. When plotted against this parameter, the values of  $kT_c$  for all eight curves in Fig. 2.3 collapse to a single smooth function, independent of the size of the individual holes or their precise geometrical arrangement. The curve is a fit to  $f(s) = A(1 - s)^c$ , with  $c = 1.93(4)$  and a goodness-of-fit estimator  $\chi^2/\text{d.o.f.} = 8.72/6$  (d.o.f. = degrees of freedom). On the right-hand vertical axis we also plot the zero-temperature effective bending rigidity in units of  $\kappa_0$  (red dotted line), which is simply linear in  $(1 - s)$ . The error bars represent an estimate of our systematic error, as explained in the text. . . . . 20

2.6 Peak of  $d\mathcal{R}_g^2/d\tilde{\kappa}$  for our most perforated sheet and six system sizes  $L$ . Inset: scaling of the height of the peak with an exponent  $\theta = 4/d_H + 1/\nu = 2.88(7)$ , from a fit with  $\chi^2/\text{d.o.f.} = 4.67/4$ . The expected value for the crumpling transition [5] is  $\theta \approx 2.82$ . All error bars represent the standard error of the mean. . . . . 21

2.7 Crumpling of a thin frame. The top row shows the initial ( $T = 0$ ) configuration for frames of  $L = 100a$  and  $W = 24a, 12a, 6a, 3a$  (left to right). The bottom row shows thermalized configurations (for  $\tilde{\kappa} = 1.25kT$  and  $\epsilon = 1800kT/a^2$ ) for each of these geometries, showing a clear crumpling as the frame width  $W$  is reduced. Points  $A$  and  $B$  of the leftmost frame are used to define an order parameter for crumpling in the text. . . . . 23

2.8 Scaling in thin-frame crumpling. We plot the radius of gyration for frames of different  $L$  and  $W$  against  $\ell_p/L$ , where the persistence length  $\ell_p = 2W\kappa^R(W)/kT$  and the renormalized bending rigidity  $\kappa^R(W)$  are defined in (2.11). The curves for different system sizes collapse when plotted against this scaling variable. The inset shows that neglecting thermal renormalization of the bending rigidity, that is, considering  $\ell_p^0 = 2W\kappa_0/kT$ , leads to a poorer collapse. In all these simulations we have used  $\tilde{\kappa} = 1.25kT$  and  $\epsilon = 1800kT/a^2$ . All error bars (not visible at this scale in most cases) represent the standard error of the mean. . . . . 25

LIST OF FIGURES

2.9 Two-step crumpling transition in an anisotropic sheet. The zig-zag pattern of approximately vertical struts reinforced by edge-sharing triangles make this structure more rigid in the vertical than in the horizontal direction (see highlighted example in the figure). We show snapshots of thermalized configurations for several temperatures. As  $T$  increases, the anisotropy in the pattern of perforations makes the membrane first fold into a tight cylinder, before crumpling completely. This geometry corresponds to the system labeled Pattern 3 in Fig. 2.3 and in Supplementary Fig. 1. The  $T = 0$  snapshot (a) is a close-up to a  $30a \times 30a$  section of the lattice, while the finite-temperature snapshots (b) show the full  $100a \times 100a$  system. . . . . 27

3.1 Ribbon of dimension  $L = 70a$ ,  $W = 17.3a$  discretized using a triangular lattice with each edge length  $a = 1$ . The light green line segment depicts the backbone of the ribbon. Two columns of nodes marked Red on the left and Violet on the right, denote the nodes which are clamped. Initially, we clamp the Red nodes and allow the violet nodes to slide along the x-direction in the  $z = 0$  plane. All other nodes are free to move in the 3D Euclidean space. The x-position of violet nodes is then averaged over several runs to get the projected thermalized length  $L_{\bar{\kappa}}$  of the ribbon at this temperature. Once we have  $L_{\bar{\kappa}}$  we clamp the Violet nodes at this new position and rerun the simulations, to study long wavelength behavior of thermalized ribbons in tension-free bending dominated regime. . . . . 32

3.2 Projected Thermalized Length  $L_{\bar{\kappa}}$  for ribbons versus rest length in the range  $L = 30a$  to  $L = 100a$ . Clamping both ends of the ribbon with distance  $L_{\bar{\kappa}}$  between them ensures that there is no tension in the system.  $L_{\bar{\kappa}}$  is plotted for two aspect ratio's AR=0.25 and AR=0.375. A straight line fit for the two sets of data shows that they are parallel(see inset) and lower AR has a shorter  $L_{\bar{\kappa}}$  for the same rest length of ribbon. . . . . 33

LIST OF FIGURES

3.3 Top: Snapshots of the ribbon in Up and Down state. Bottom: Height of the ribbon is averaged along the width (average height of nodes in a column in the y-direction) to get  $h(x)$ . Fourier transform of  $h(x)$  gives  $\tilde{h}(q)$  and this activity is done for all snapshots over all runs, to evaluate  $|\tilde{h}(q)|^2$  with errors. 35

3.4 Fitting  $|\tilde{h}(q)|^2$  obtained from clamping ribbon at its projected thermalized length  $L_{\tilde{\kappa}}$  (violet) and rest length (ribbon length when  $T=0$  in blue) to the function  $1/(Sq^2 + Bq^4)$ . Rest length clamping curve is dominated by  $q^{-2}$  stretching term whereas  $L_{\tilde{\kappa}}$  clamping curve is dominated by  $q^{-4}$  bending term. By clamping the ribbon at its  $L_{\tilde{\kappa}}$  length we have successfully tuned any induced tension out of the system. ( $S = \sigma/kT$  and  $B = \kappa/kT$ ) . . . . . 36

3.5 Ribbon of rest length  $L = 100a$  and  $W = 34.6a$  is gradually compressed and clamped at different compressed lengths. Stretching coefficient  $S$  is evaluated for each and plotted against the compressed length (right end of the ribbon), left end is clamped at  $x = 0$ . Stretching coefficient becomes negative when compression crosses the projected thermal length of this system (marked by green point). The inset shows a zoomed section of the plot near  $S = 0$  which clearly shows the cross-over from positive  $S$  (stretched case) to negative (compressed case). The green dot is obtained from the sliding procedure explained in Fig. 3.1. . . . . . 38

3.6  $\langle h_{CM}^2 \rangle$  is normalized using  $L^2$  versus  $l_p^0(W)/L$ . As one can see the plot doesn't collapse for longer ribbons. . . . . 40

3.7  $\langle h_{CM}^2 \rangle$  is normalized using  $W^{2+2\eta}/l_{th}^{2\eta}$  versus  $l_p(W)/L$ . The plot collapses best when renormalization of the width is taken into account. . . . . 41

3.8 Top: Height time series of CM of ribbon  $h_{CM}(t)$  ( $L = 70a$ ,  $W = 34.6a$ ). First 10% of data is discarded to allow for equilibration. A filter is applied to extract the time spent by CM in the Up/Down state. Bottom: CM height time series for ( $L = 70a$ ,  $W = 69a$ ) corresponding  $AR \sim 1$ . Two states are distinctly visible in this case. Both ribbons are clamped at their projected thermalized length  $L_{\tilde{\kappa}}$ . . . . . 43

LIST OF FIGURES

3.9 Time autocorrelation function for ribbon length  $L = 70a$  with different widths. As the width increases the survival time increases drastically. . . . 44

3.10 Post filtering the CM height time series into Up and Down survival times from multiple runs (10 runs for each parameter set),probability distribution function (PDF) is generated. Top: Shows the PDF to be a Poisson distribution for ribbon of dimensions  $L = 70a$  and  $W = 34.6a$  for one run. Middle: Cumulative Distribution Function (CDF) for the above PDF. Bottom: Complementary Cumulative Distribution Function(CCDF=1-CDF) is generated from CDF. Area under this curve is defined as the Integrated Survival Time, since  $\int_0^\infty \exp(-x/\tau)dx = \tau$ . Note that the error bars, in this case, are generated using bootstrap method [52] since the underlying distribution is not gaussian. . . . . 46

3.11 Integrated survival time  $\tau_{survival}$  is studied for different aspect ratio's over a range of ribbon lengths by varying width. Integrated survival time for larger system sizes ( $L = 60a$  and  $L = 70a$ ) jumps sharply around  $AR \sim 1$ . Inset depicts  $\tau_{survival}$  values for lower aspect ratio's showing clearly that at a given aspect ratio  $\tau_{survival}$  increases for larger system sizes. . . . . 48

3.12 The average squared height of the ribbon CM is plotted against relative compression for different ribbon geometries. In the stretched case (negative compressions) the CM height is almost zero. Compressing the ribbon beyond it's  $L_{\tilde{\kappa}}$  causes the ribbon CM to jump immediately, which increases with compression and at a fixed relative compression is greater for the longer ribbon. The ribbons studied here all had fixed aspect ratio of 0.25. . . . . 49

3.13  $\tau_{survival}$  as a function of relative compression but fixed  $AR=0.25$ . Notice, that  $\tau_{survival}$  takes off for cases where the ribbon is compressed beyond its  $L_{\tilde{\kappa}}$ . Looking at relative compression values at 0.5,  $\tau_{survival}$  for longer ribbons is higher, indicating  $\tau_{survival}$  may diverge for sufficiently longer ribbons. . . . 50

## LIST OF FIGURES

3.14 Comparing integrated survival time for ribbons with AR 0.25 and 0.375. Clearly, AR=0.375 ribbons exhibit persist in Up/Down state longer than ribbons with lower AR. . . . .	51
3.15 Ribbon of rest length $L = 70a$ and $W = 34.6$ is clamped at its projected thermalized length $L_{\bar{r}}$ and its Up/Down survival time evaluated using the autocorrelation of height time series. In this semi-log plot, we fit a straight line and survival time in, this case is, 25.837 MD time steps. . . . .	54
3.16 Comparing the two methods for survival time estimation . . . . .	55
4.1 Neighboring triangular plaquettes with indexed lattices . . . . .	57
4.2 Neighboring triangular plaquettes with indexed lattices . . . . .	61



# Chapter 1

## Introduction

In this dissertation, we will deal with thin sheets. The mechanics of solid bodies under the action of external forces is the subject of study in theory of elasticity [1]. The external force applied to a specific area is called stress. The resulting deformation in the solid is called strain. In the classical picture, where thermal fluctuations are ignored, a thin elastic plate has two contributions to its energy, one due to bending and the other due to stretching. The bending energy of the plate is characterized using the bending rigidity ( $\kappa$ ) of the material and the stretching energy by the Lamé coefficients ( $\lambda, \mu$ ) or more commonly by the Young's modulus ( $Y$ ) of the material. One can write the energy of the plate in the continuum limit and then use the nonlinear strain matrix, along with external boundary conditions, and the Airy stress function to obtain the Föppl-von Kármán equations which gives the shape of the deformation of the thin plates under the action of external forces. However, in this Föppl-von Kármán theory, the elastic constants of the material, namely  $\kappa$ ,  $\lambda$  and  $\mu$ , are constant and do not change as a function of the length scale.

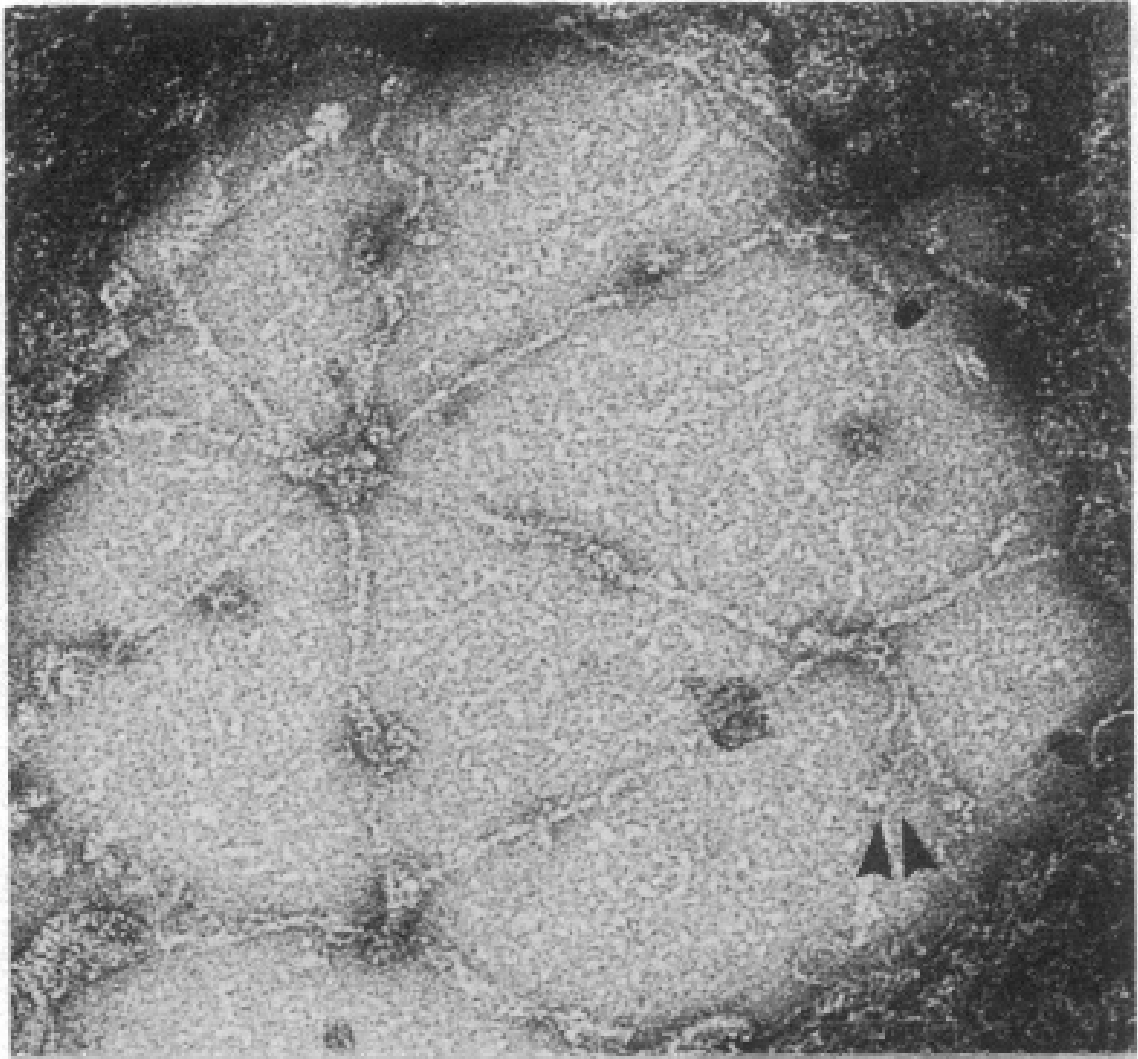
An interesting thing happens when contributions from thermal fluctuations are taken into account. The elastic parameters mentioned above start running as a function of the length scale  $l$ , or equivalently momentum  $q$ :  $\kappa(q)$ ,  $\lambda(q)$  and  $\mu(q)$ . This particular behavior, attributed solely to thermal fluctuations, has been the basis of many theoretical and numerical predictions from 1987 until today [2–19].

The free energy has contributions from a pure energy term and an entropy term. Thermal

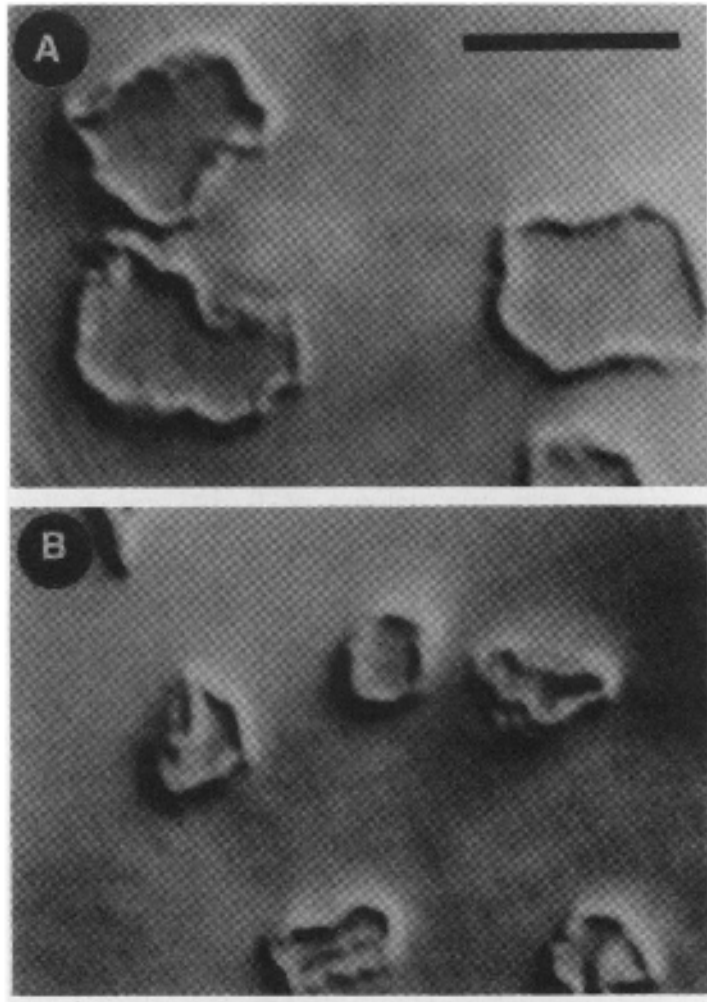
fluctuations become crucial when the entropy term dominates over the energy term. An easy way to look at this is to imagine a two-dimensional lattice of connected nodes kept at room temperature ( $kT \sim 1/40eV$ ). The Boltzmann factor  $e^{-\epsilon/kT}$  comes out to be of the order of  $10^{-18}$ , which is exceedingly small. Note that scale of  $\epsilon$  here is  $\epsilon \sim eV$  for soft materials like lipid bilayers. One might then argue that thermal fluctuations are of no significance. But this deduction is not correct, since we only have half the picture. The effect of thermal fluctuations is determined by the Boltzmann factor multiplied by the density of states. The density of states for a fluctuating sheet is exponentially large since there are many nearly degenerate configurations available. Hence, thermal fluctuations cannot be ignored.

Before we dive into discussing these results, a cursory look at polymers will help us get a quick idea of the kind of order we might be interested to look at in a two-dimensional material. A polymer is modeled as a one-dimensional chain of nodes. If we look at a polymer subject to thermal fluctuations, the normal-normal correlation function which measures the long range order in normals at large scales goes to zero [20, 21] as one increases temperature. So at sufficiently large separations, this system is always disordered. An immediate question one can ask is whether we can see an ordered phase if we go from a polymer to a two-dimensional sheet or membrane? An ordered phase here would be a flat phase with long range order in the normals, as opposed to a disordered crumpled phase with no long range normal-normal order. Interestingly, there are different kinds of two-dimensional sheets or membranes example liquid membranes [22], and elastic membranes [2, 23]. Liquid membranes are governed by bending energy alone since they have vanishing elastic moduli. Elastic membranes have both bending and elastic contributions to their free energy.

We first look at examples of liquid membranes like red blood cells and microemulsions [26]. Fig. 1.1 and Fig. 1.2 shows electron micrograph images of an erythrocyte, or red blood cell, cytoskeleton. The cell or plasma membrane of a red blood cell is composed of a bilayer of amphiphilic molecules. Each amphiphilic molecule has one or more hydrophobic hydrocarbon tails and a polar head. The topology of the cell is spherical and these cells have very small shear modulus at the temperature levels of a living cell. Hence, these cells



**Figure 1.1:** An electron micrograph of a portion of erythrocyte cytoskeleton. The skeleton has been spread to a surface area nine to ten times as large as in its natural state [24].



**Figure 1.2:** Isolated RBC skeletons viewed under video-enhanced differential interference contrast microscopy. A and B show different closeups [25].

are ideal for studying a system which is purely bending driven as any shearing force will tear the cell. These systems are studied by modeling the membrane as a triangular lattice and using the Monge representation for the position. Normals to each triangular lattice can be erected and the bending energy can be written simply by taking the dot product of neighboring normals over the entire membrane, with bending rigidity acting as the coupling constant. One can now estimate fluctuations in the normals by taking the dot product of normals with the z-axis and passing to Fourier space. As shown in [27], the normal-normal correlation in these liquid membranes depends logarithmically on the system size, and hence diverges (goes to infinity) with the system size  $L$ . So, like polymers, one doesn't find long range order in normals. Peliti and Leibler [28] calculated the renormalized wave-vector-dependent bending rigidity ( $\kappa_R(q)$ ) and found that the thermal fluctuations soften these liquid membranes and in fact, make the normal-normal divergence more severe. The correction to the bare bending rigidity is negative. The bending rigidity decreases as a result of fluctuations. Even though we moved from a one-dimensional polymer to a two-dimensional liquid membrane, we still do not get an ordered phase and one sees crumpling at finite temperature.

We next consider elastic membranes with bending energy as well as a tethering potential between nearest-neighbor vertices [2, 29]. Again one works with a triangular lattice to discretize the membrane with each vertex having a fixed connectivity (six in this case). The tethering potential essentially introduces stretching in the system, governed by the Lamé coefficients  $\lambda$  and  $\mu$  mentioned earlier. We now have two competing terms in the free energy and the system tries to minimize the total energy by trading one for the other. Examples of such systems are Boron Oxide  $B_2O_3$  [30] and graphene [31]. Fig. 1.3 shows a planar section of a  $B_2O_3$  sheet and Fig. 1.4 shows an artist's impression of single layer graphene. In this case, thermal fluctuations will renormalize both the bending rigidity and the two dimensional Young's modulus. The Young's modulus penalizes any membrane distortions that cause non-zero Gaussian curvature and hence the system searches for low energy configurations which are isometric. One can crumple a paper piece, which has almost infinite in-plane elastic constants, to see that there are many low energy configurations available to the

system. This particular assertion is important because graphene sheets, which are the core of research presented in this dissertation, have a very high Young's modulus. An important dimensionless parameter in the context of two-dimensional sheets is the Föppl-von Kármán number, which is the ratio of the stretching energy for a side ( $L$ ) square sheet to the bending energy of the sheet. The Föppl-von Kármán number for a  $10\mu\text{m} \times 10\mu\text{m}$  graphene sheet is of the  $\mathcal{O}(10^{12})$ . This high number only exemplifies the idea that these sheets bend much more easily than they stretch. Nelson and Kantor [2, 29] calculated the running of the bending rigidity and found the corrections to the bare bending rigidity to be positive. So, the presence of stretching energy has the effect of stiffening the membrane due to thermal undulations. When this new renormalized bending energy is used to calculate the normal-normal correlation one finds that the fluctuations are finite now. This is a clear indication that the presence of stretching along with bending has a stabilizing effect on the long range order of the normals. As opposed to liquid membranes described earlier, we now have a two-dimensional system which shows evidence of long range order in normals at a low temperature. But we know that entropy still favors a crumpled state at high temperature. The finite normal-normal correlation indicates the presence of a low temperature flat phase. Therefore, there should be a finite-temperature crumpling transition present in these elastic systems. Subsequent computer simulations [7, 8, 32, 33] supported this result for elastic membranes without self avoidance.

The crumpling transition has not been experimentally observed yet. In many materials, the crumpling transition is very far from the experimentally accessible regime. Let's look at graphene, for example, which is an atomically thin sheet of connected carbon atoms and provides a plausible testing bed for these experiments. The bending rigidity of graphene  $\kappa \approx 48kT_R$ , where  $T_R$  is the room temperature [9]. The crumpling temperature of graphene comes out to be of order  $10^4 - 10^5 K$ , which is well beyond the melting point of graphene. One can think of using a softer material to work around this problem. But unfortunately, this doesn't work either. Thermal fluctuations become important on scales  $q^{-1}$  larger than the thermal length which is inversely proportional to the square root of Young's modulus. Softer materials have a low Young's modulus which makes the thermal length for such

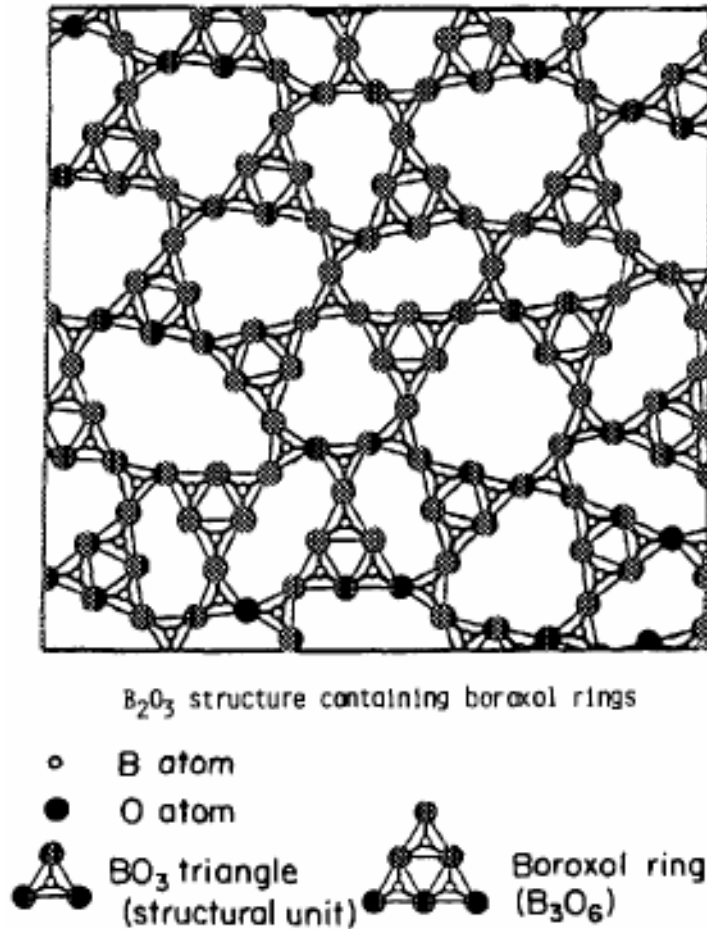
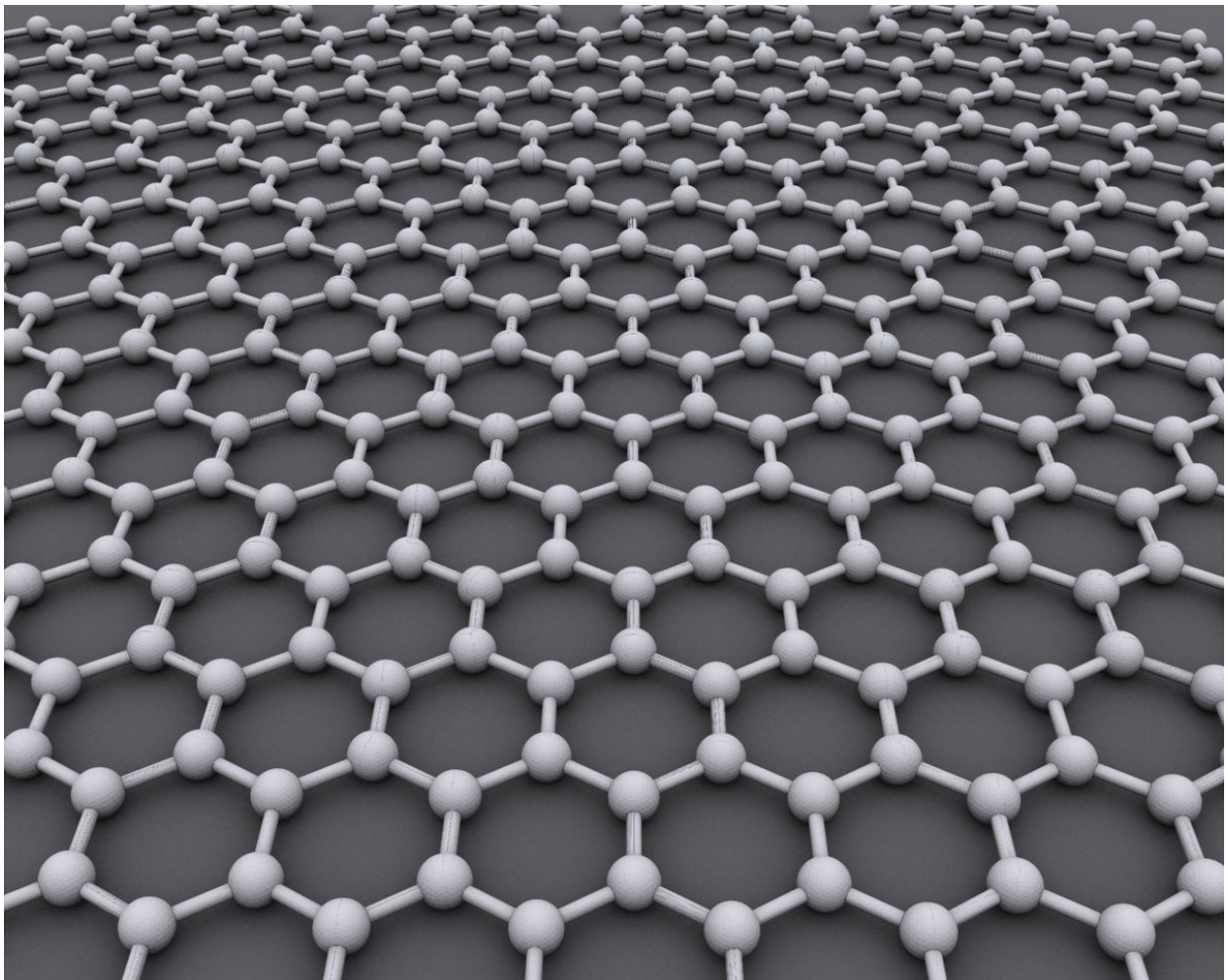


Figure 1.3: A planar section of Boron Oxide  $B_2O_3$  [30].

systems very large, they don't see effects of thermal fluctuations. So, one is forced to work with a material with high Young's modulus, which in turn will give a small thermal length and thermal fluctuations will set in quickly. For graphene, the thermal length is of the order of 1Å and thermal fluctuations become important quickly but still one has to deal with the very high crumpling temperature. Clearly, this situation requires a means of lowering the crumpling temperature to an experimentally viable regime.

Chapter 2 in this dissertation attempts to answer this long-standing problem by providing a mechanism for lowering the crumpling temperature of existing two-dimensional membranes. Ideas from Origami(ori means "folding" and gami means "paper") and kirigami(kiri means "cut") techniques traditionally used in making complex shapes and structure with





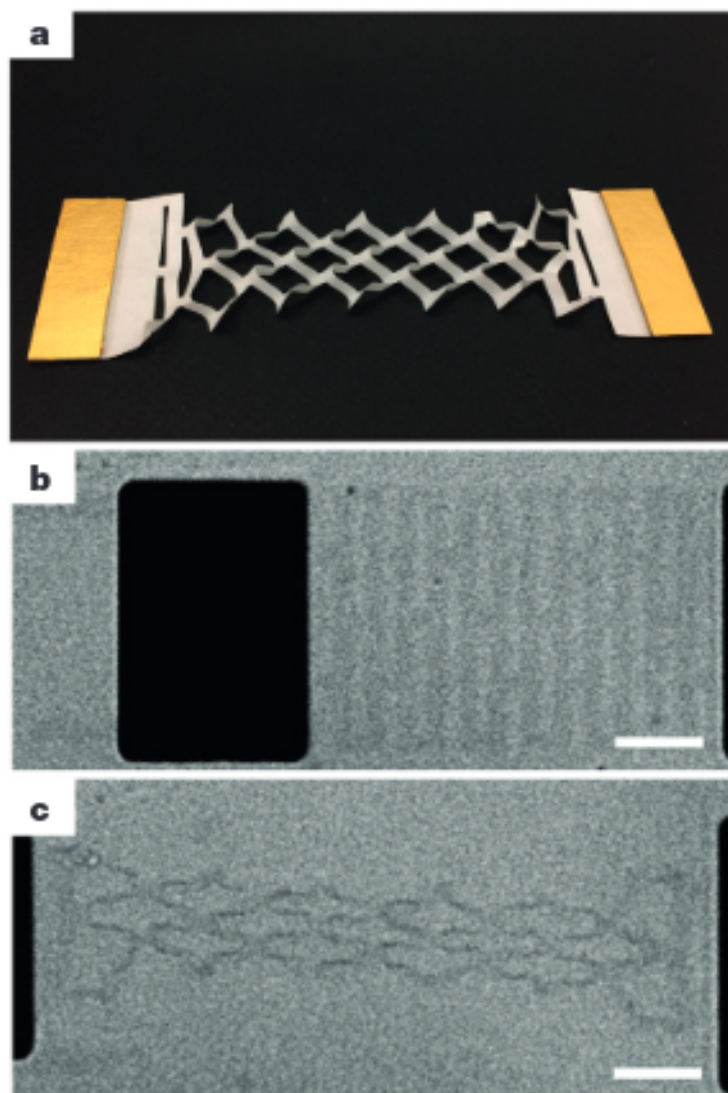
**Figure 1.4:** Graphene is an atomic-scale hexagonal lattice made of carbon atoms - Wikipedia.



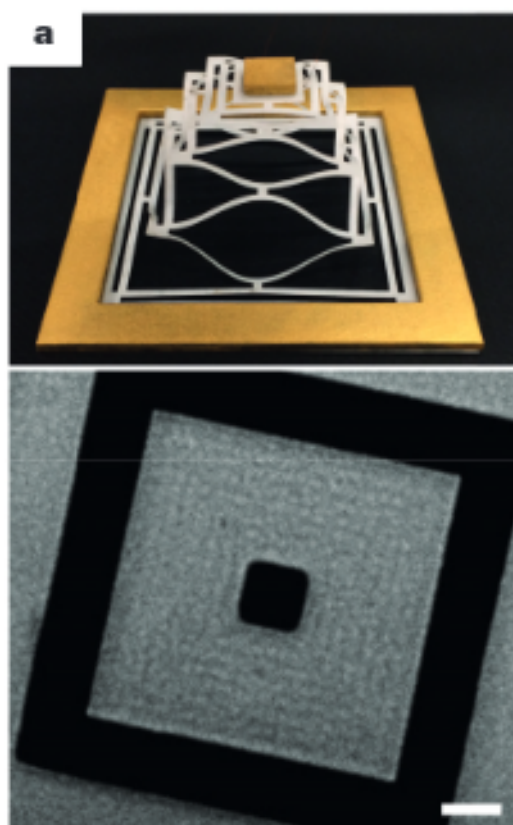
paper are used to design new metamaterials from these two-dimensional membranes. Metamaterials are materials which are engineered to have properties that are not found in nature. Specifically, these techniques alter the shape, geometry, size, and orientation of the material to engineer the elastic parameters (bending rigidity and Young's modulus). Both mechanical and electrical properties can be modified using this approach, though we will concentrate only on the mechanical aspects. Recent work by Blees et al. [34], in addition to demonstrating a 4000-fold enhancement of the bending rigidity relative to its  $T = 0$  value has shown the potential of graphene as the raw ingredient of microscopic mechanical metamaterials. Fig. 1.5 and Fig. 1.6 gives examples of such metamaterials which have been fabricated using graphene and have generated renewed interest in looking at graphene as a base material for designing metamaterials with engineered properties. Employing the principles of kirigami, one can construct robust microstructures, thus providing an alternative route to tune mechanical properties, leaving graphene's remarkable electrical properties essentially intact. Similarly, it has been shown numerically that a simple cut in a ribbon causes thermal stiffening [35]. Using these ideas as inspiration, we will show that perforating a two-dimensional sheet paves the way for lowering the crumpling temperature in these systems.

Cantilevers are long projecting beams fixed at one end used in the construction of bridges and buildings. Recently, at the microscale cantilevers are being made by cutting graphene sheets. These have now been studied experimentally in great detail with application in microelectromechanical (MEMS) and nanoelectromechanical (NEMS) devices [36]. Also, micro and nano-scale oscillators have been studied for a long time [37, 38] to explore their mechanical response and energy dissipation mechanism. Given that these system sizes are in the microscale the classical approach to bending is not adequate in understanding their thermal response as thermal fluctuations will inadvertently play a significant role in changing the bending rigidity of the beams as explained earlier. One other aspect which has not been explored completely is characterizing the response of these cantilevers when geometric changes are made to it.

In chapter 3, we explore the system dynamics of these microscale cantilevers as a function of its aspect ratio (width/length). We find interesting results pertaining to the oscillatory



**Figure 1.5:** a. Paper model of spring.  
b and c. Graphene spring [34].



**Figure 1.6:** Graphene rectangular spring [34].

motion of these systems, which can be tuned just by playing with the dimensions of the cantilever beam. We also find evidence of thermal renormalization in these systems. Incidentally, no thermal analogue of Euler buckling has yet been proposed where out of plane buckling can be controlled by temperature alone. In this dissertation we will show thermalized Euler buckling in these cantilever devices.

Finally, chapter 4 and 5 elaborate on the simulation and analysis methods used.

## Chapter 2

# Thermal crumpling of perforated two dimensional sheets

### 2.1 Introduction

In this chapter we propose a mechanism using which one can lower the crumpling transition temperature for two dimensional sheets to an experimentally accessible regime. We successfully show that the crumpling temperature can be significantly lowered by altering the geometry and topology of the sheet. In particular, we perform extensive molecular dynamics simulations of crystalline membranes with dense periodic arrays of holes and determine the dependence of the onset of crumpling on the degree of perforation. This work was done in collaboration with David Yllanes, Mark Bowick and David Nelson which has been published in Nature Communications [39]. My contribution was running simulations, writing code to analyze the raw simulation data and plotting figures.

### 2.2 Model

We consider square sheets of size  $L \times L$ , which for the purposes of computer simulation we discretize with a tiling of equilateral triangles of side  $a = 1$ . We use a standard coarse-grained model [40] to compute the elastic energy in the sheet, which is composed of a

stretching and a bending term

$$\mathcal{H} = \mathcal{H}_{\text{stretch}} + \mathcal{H}_{\text{bend}}. \quad (2.1)$$

Stretching is modeled by considering each triangle side as a spring of elastic constant  $\epsilon$  and rest length  $a$ :

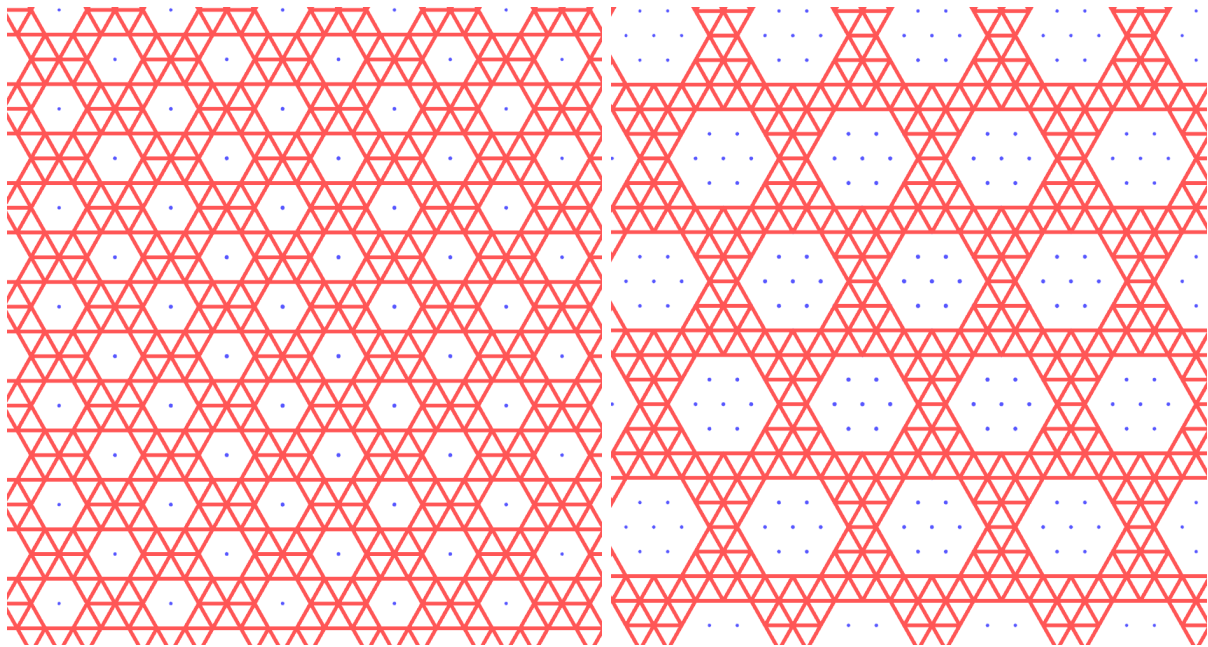
$$\mathcal{H}_{\text{stretch}} = \frac{1}{2}\epsilon \sum_{\langle i,j \rangle} (r_{ij} - a)^2, \quad (2.2)$$

where the sum is over all pairs of vertices joined by a triangle edge. The bending energy is represented by a standard dihedral interaction between normals,

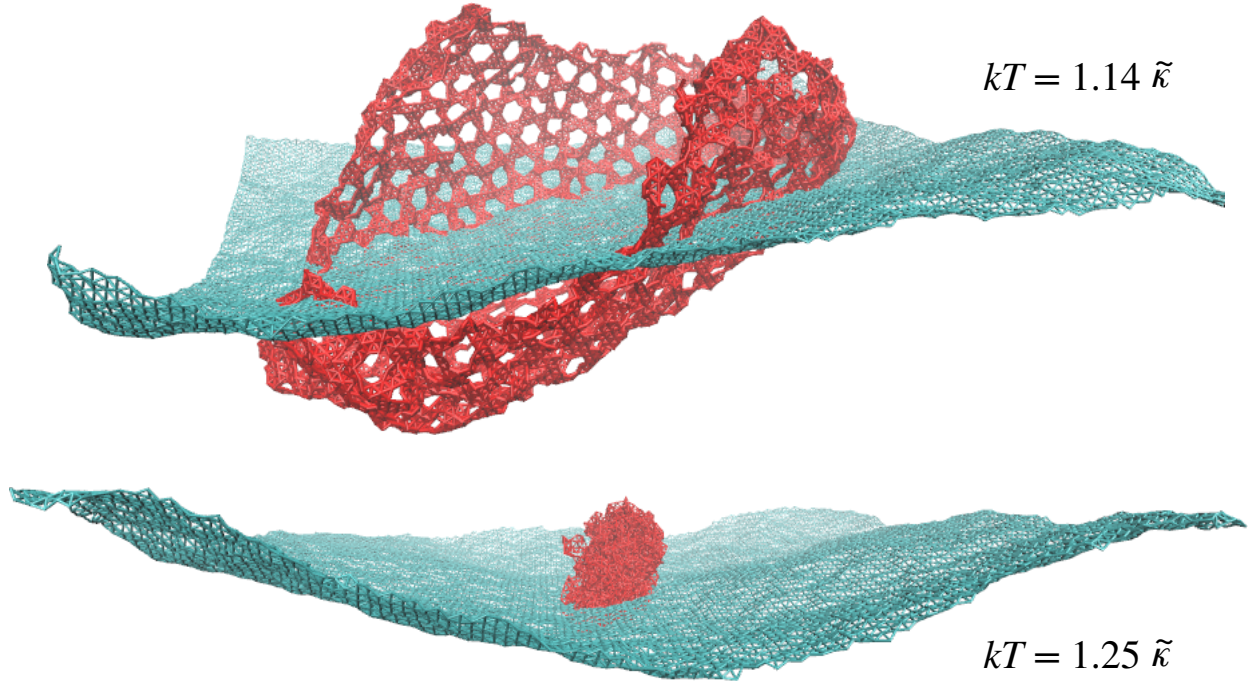
$$\mathcal{H}_{\text{bend}} = \tilde{\kappa} \sum_{\langle \alpha, \beta \rangle} (1 - \hat{\mathbf{n}}_\alpha \cdot \hat{\mathbf{n}}_\beta). \quad (2.3)$$

Here the sum is over all the pairs of triangles that share a side and  $\hat{\mathbf{n}}_\alpha$  is the unit normal to triangle  $\alpha$ . Note that placing a carbon atom at the center of each triangle provides an approximate atomic model for the elastic modes of graphene on a dual lattice, as long as we choose the bending rigidity and Young's modulus correctly.

The elastic parameters  $\epsilon$  and  $\tilde{\kappa}$  are directly related to the continuum Young's modulus ( $Y_0 = 2\epsilon/\sqrt{3}$ ) and bare bending rigidity ( $\kappa_0 = \sqrt{3}\tilde{\kappa}/2$ ). Normally, when performing a numerical study (see, e.g., [7, 41]) one chooses natural units where  $\epsilon = 1$  and  $\tilde{\kappa}$  is varied. To better approximate the behavior of materials such as graphene or MoS<sub>2</sub>, however, we will instead fix the ratio  $\epsilon/\tilde{\kappa}$  and vary temperature by changing the  $\tilde{\kappa}/kT$  ratio. For graphene at room temperature  $\tilde{\kappa}/kT \approx 48$  [9]. Following [35, 42], we will use  $\epsilon/\tilde{\kappa} = 1440/a^2$ . This corresponds to a Young's modulus about an order of magnitude lower than for real graphene [43, 44], in order to facilitate equilibration in our computer simulations. We note, however, that this choice should have only a minor effect on the onset of crumpling, since the degree of order in the normals only depends on  $\epsilon/\tilde{\kappa}$  logarithmically [19]. For a sheet of length  $L/a = \mathcal{O}(10^2)$ , corresponding to a patch of freely suspended graphene roughly 30nm on a side, these pa-



**Figure 2.1:** Close ups of sections of the membrane for two different arrays of perforations. We begin by considering a full triangulated sheet and then remove all the nodes in a radius  $R$  around its center. This central hole is then repeated periodically throughout the membrane. In the figure the removed nodes are represented by blue dots (*Left:  $R = 1$ , Right:  $R = 2$* ). In the rest of the paper we will consider patterns of perforations with  $R = 1, 2$  and varying spacing between holes (see Figure 8 in the Supplementary Material for a full description of all our perforation patterns).



**Figure 2.2:** Snapshots of thermalized configurations. We superimpose the configurations of a pristine sheet (blue) and a perforated sheet with holes of size  $R = 2$ , in the pattern of Fig. 2.1-b (red) for two values of the temperature (in units of the bending rigidity  $\tilde{\kappa}$ ). In both cases the full sheet is well into the flat phase, the thermal fluctuations causing just some wrinkling and oscillation. This 10% increase in temperature, however, triggers a crumpling of the perforated sheet. Both systems have size  $L = 100a$ .

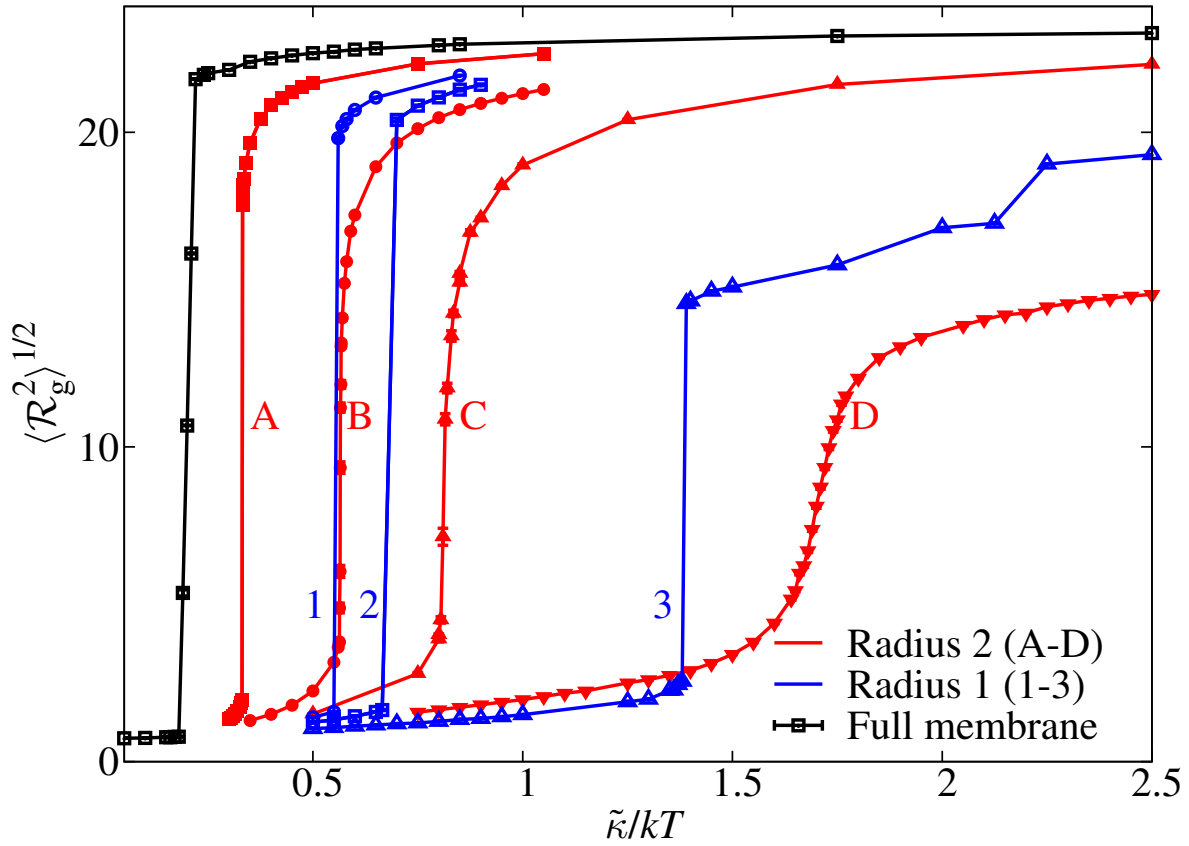
rameters result in a Föppl von Kármán number of  $\nu K = Y_0 L^2 / \kappa_0 \sim \mathcal{O}(10^7)$ , similar to that of a standard A4 sheet of paper.

### 2.2.1 Dense arrays of holes

We are interested in exploring the effect of a perforated geometry on the rigidity of elastic membranes. To this end we shall compare the physics of the “full” or unperforated sheet described above with that of a sheet with a dense array of holes. We begin by removing the node  $i = 0$  situated in the center of the sheet and all the nodes  $j$  such that  $r_{0j} = |\mathbf{r}_j| < R$ . We then repeat this operation periodically throughout the lattice to create a dense lattice of perforations (see Fig. 2.1 and Supplementary Fig. 1). In this paper we consider arrays of holes of size  $R = 1, 2$  with varying spacing. We kept the radius of the hole small relative to the length of the sheet to minimize finite-size effects.

As a first demonstration of the dramatic effect of these perforations, consider the pattern





**Figure 2.3:** Radius of gyration as a function of  $\tilde{\kappa}/kT$  for several patterns of perforation. The black curve is the baseline unperforated membrane, which crumples at the highest temperature,  $\tilde{\kappa}/kT_c \approx 0.18$ . The red curves (A to D) are for triangular arrays of perforations of radius  $R = 2$  with decreasing spacing between individual holes. The blue curves (1 to 3) are arrays of perforations with radius  $R = 1$ . As the spacing between holes is reduced, the crumpling temperature decreases. The full description of the perforation patterns A–D and 1–3 is given in Supplementary Fig. 1. Data for systems of size  $L = 100a$ .

of holes depicted in Fig. 2.1–right. Fig. 2.2 compares the equilibrium configurations of this perforated sheet and those of the full membrane for two values of temperature that differ by only 10%. In both cases, the full membrane is deep in the flat phase and exhibits smooth, approximately flat configurations. The perforated sheet, on the other hand, experiences a crumpling transition.

To characterize this transition it will be useful to consider the radius of gyration of the sheet

$$\mathcal{R}_g^2 = \frac{1}{3N} \sum_{i=1}^N \langle \mathbf{R}_i \cdot \mathbf{R}_i \rangle, \quad \mathbf{R}_i = \mathbf{r}_i - \mathbf{r}_{\text{CM}}, \quad (2.4)$$

where  $\mathbf{r}_{\text{CM}}$  is the position of the center of mass and  $\langle O \rangle$  represents a thermal average. In the flat phase,  $\mathcal{R}_g^2 \sim L^{4/d_H}$ , with Hausdorff dimension  $d_H = 2$ , while in the crumpled phase  $\mathcal{R}_g^2 \sim \log(L/a)$  ( $d_H = \infty$ ). In the critical region, the Hausdorff dimension has been computed with analytical methods ( $d_H = 2.73$  [5]) and with numerical simulations ( $d_H = 2.70(2)$  [41]).

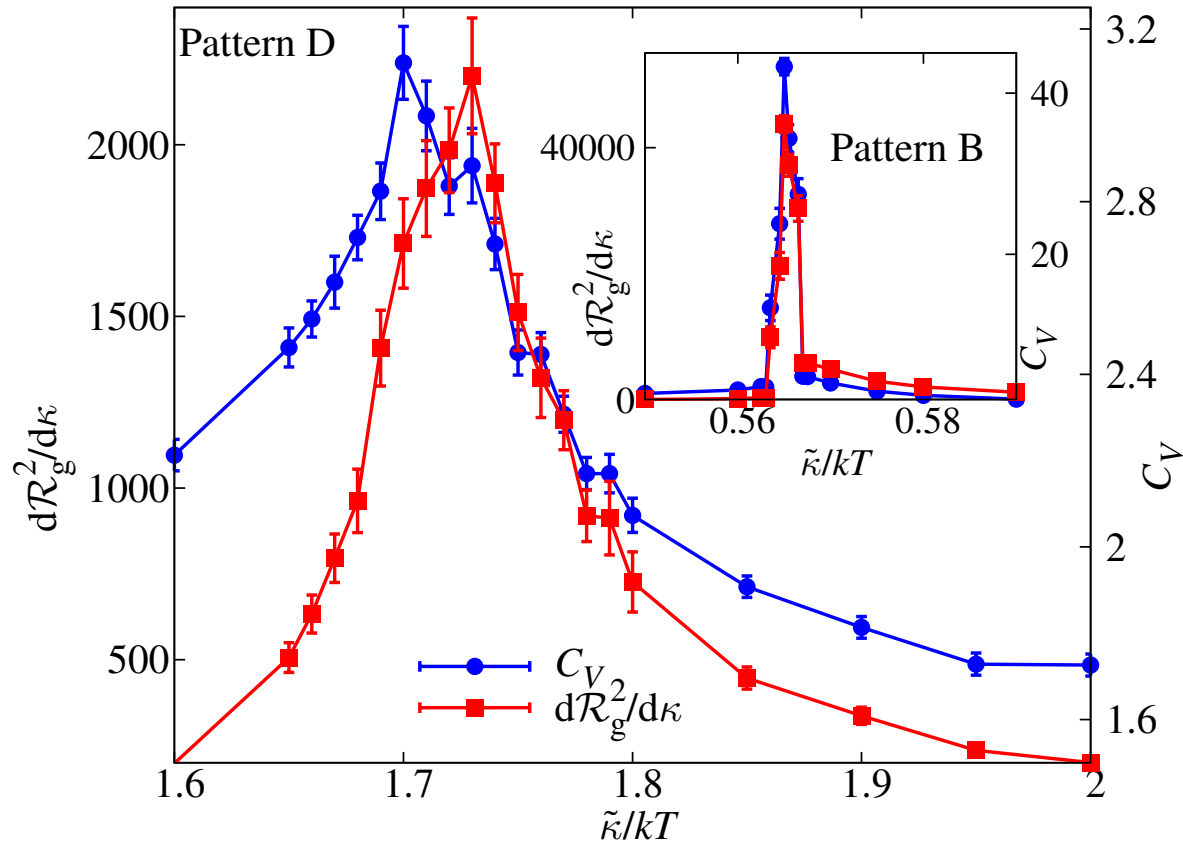
We have plotted  $\mathcal{R}_g$  as a function of  $\tilde{\kappa}$  for all our perforation patterns in Fig. 2.3. In blue (red) we represent systems with arrays of holes of radius  $R = 1$  ( $R = 2$ ) and a decreasing separation between holes. The black curve provides the baseline value of  $\mathcal{R}_g^2$  for the full membrane. We are interested in computing the critical  $kT_c/\tilde{\kappa}$  for crumpling in each of these geometries. This can be done by searching for the maximum in the specific heat of the system, which can be computed as [45]

$$C_V = \frac{1}{N} (\langle \mathcal{H}^2 \rangle - \langle \mathcal{H} \rangle^2). \quad (2.5)$$

Alternatively, we can consider the  $\tilde{\kappa}$ -derivative of  $\mathcal{R}_g^2$ , which can be evaluated as:

$$\frac{d\mathcal{R}_g^2}{d\tilde{\kappa}} = \frac{kT}{\tilde{\kappa}} (\langle \mathcal{H} \rangle \langle \mathcal{R}_g^2 \rangle - \langle \mathcal{H} \mathcal{R}_g^2 \rangle). \quad (2.6)$$

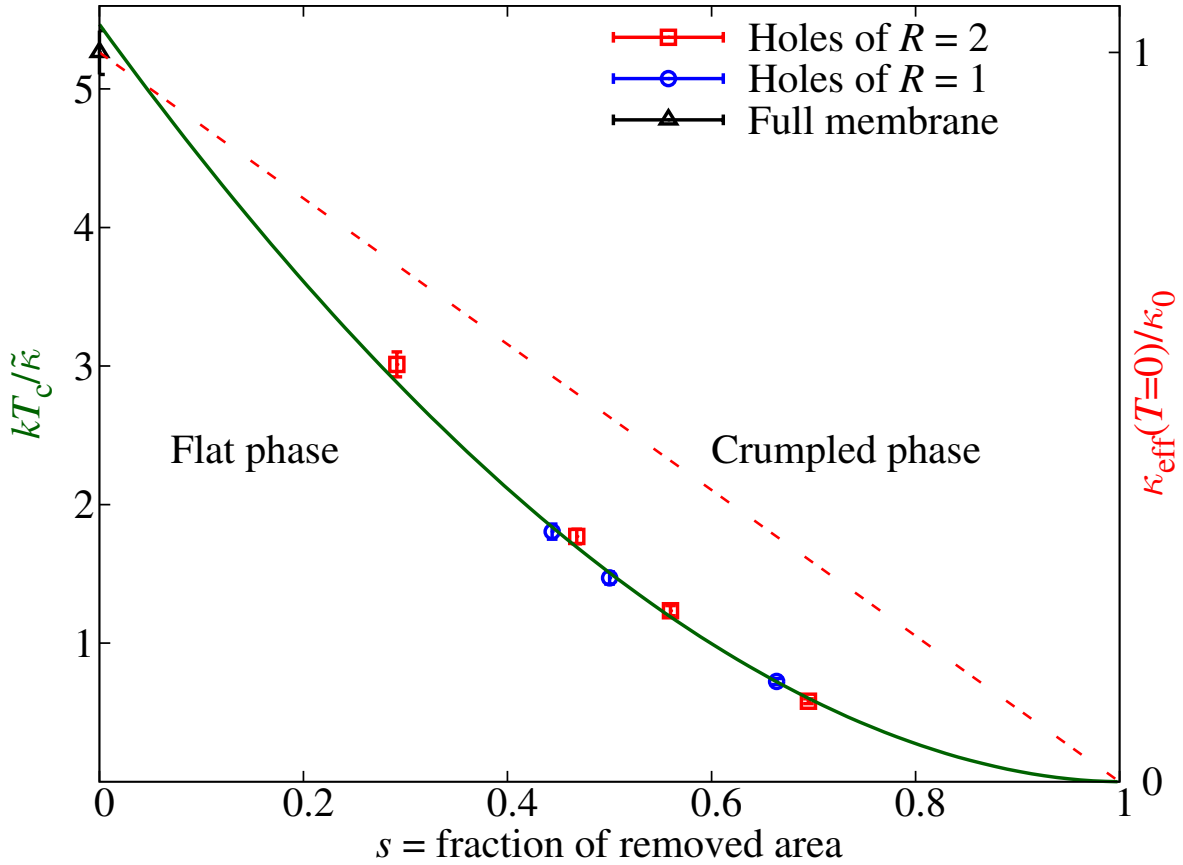
We show these two quantities for two different perforation patterns in Fig. 2.4. In the thermodynamic limit, the position of the peaks in  $C_V$  and  $d\mathcal{R}_g^2/d\tilde{\kappa}$  tend to the same  $kT_c/\tilde{\kappa}$  value. For our finite systems, we use the difference in these peak positions for our most



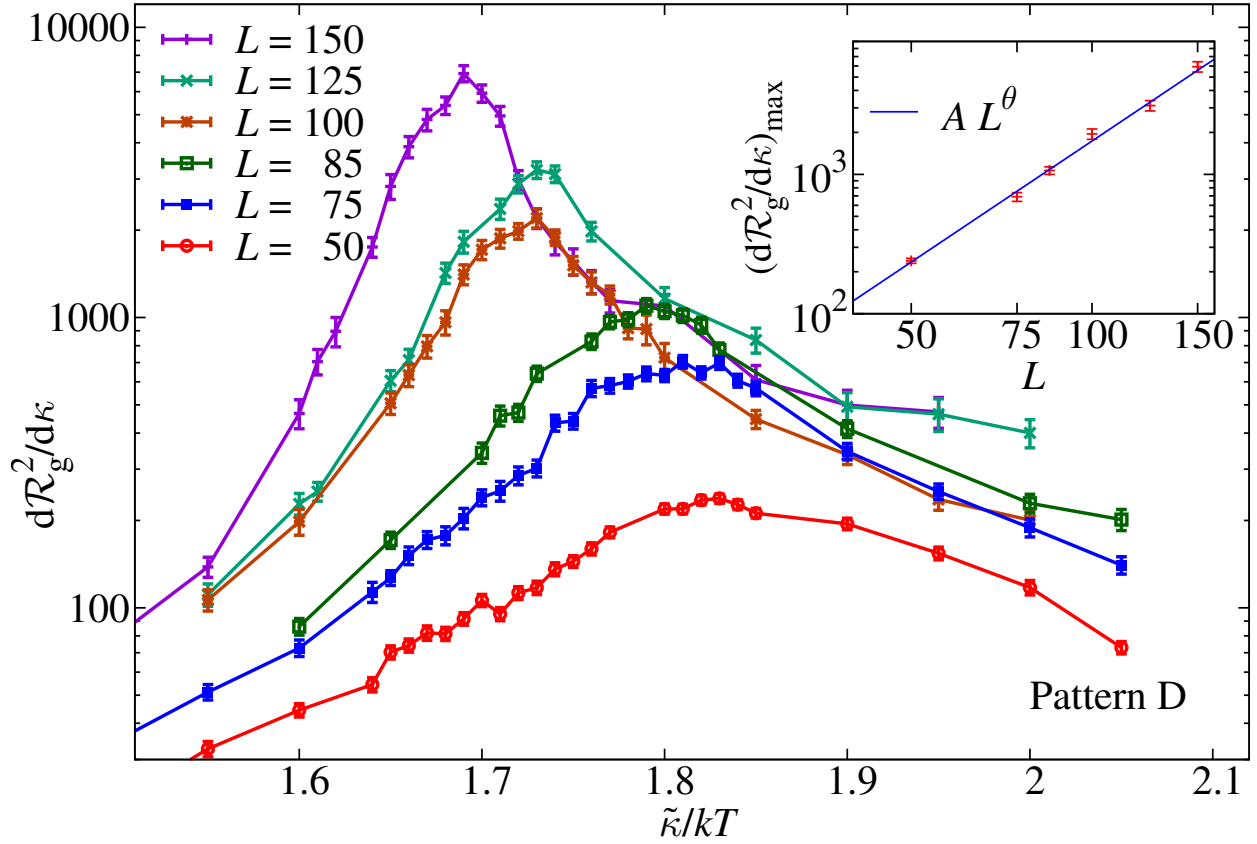
**Figure 2.4:** Location of the crumpling temperature. We plot the specific heat  $C_V$  (right axis) and  $\tilde{\kappa}$ -derivative of the radius of gyration (left axis) as a function of  $\tilde{\kappa}/kT$  for our most perforated system (corresponding to curve D in Fig. 2.3). The inset shows the analogous plot for a less perforated sheet (corresponding to curve B in Fig. 2.3), with a much sharper transition (note the different vertical scales of the axes). All error bars represent the standard error of the mean.

perforated membrane (the case where the peaks are most separated) as an estimate of our systematic error in  $\tilde{\kappa}/kT_c$ .

In principle, one could think that this  $T_c$  would depend in a complicated way on the particular spatial arrangement of the holes or on their individual sizes. Fortunately, the reality is much simpler. Indeed, in Fig. 2.5 we have plotted the  $kT_c/\tilde{\kappa}$  for each of the curves in Fig. 2.3 as a function of the fraction of removed area in the sheet. Given our discretization, this areal fraction is most easily estimated by counting the fraction of remaining dihedrals connecting adjacent triangles, after the holes have been made. As a function of this dimensionless area fraction, all our  $T_c$ , including the one for the full membrane, fall on a single smooth curve.



**Figure 2.5:** Crumpling temperature  $kT_c$  as a function of the fraction  $s$  of removed area. When plotted against this parameter, the values of  $kT_c$  for all eight curves in Fig. 2.3 collapse to a single smooth function, independent of the size of the individual holes or their precise geometrical arrangement. The curve is a fit to  $f(s) = A(1-s)^c$ , with  $c = 1.93(4)$  and a goodness-of-fit estimator  $\chi^2/\text{d.o.f.} = 8.72/6$  (d.o.f. = degrees of freedom). On the right-hand vertical axis we also plot the zero-temperature effective bending rigidity in units of  $\kappa_0$  (red dotted line), which is simply linear in  $(1-s)$ . The error bars represent an estimate of our systematic error, as explained in the text.



**Figure 2.6:** Peak of  $d\mathcal{R}_g^2/d\tilde{\kappa}$  for our most perforated sheet and six system sizes  $L$ . Inset: scaling of the height of the peak with an exponent  $\theta = 4/d_H + 1/\nu = 2.88(7)$ , from a fit with  $\chi^2/\text{d.o.f.} = 4.67/4$ . The expected value for the crumpling transition [5] is  $\theta \approx 2.82$ . All error bars represent the standard error of the mean.

In fact, if we denote by  $s$  the fraction of removed area in the perforated sheet, we have found that the following ansatz reproduces our results very accurately:

$$kT_c/\tilde{\kappa} = A(1 - s)^c. \quad (2.7)$$

With our choice of parameters, we obtain a good fit with  $c = 1.93(4)$  and  $A \approx 5.5$ . Notice, in particular, that for our most perforated sheet (where about 70% of the area has been removed), the value of  $kT_c$  is reduced by a factor of 10 compared to the full membrane. Extrapolating using Eq. (2.7), we find that removing 85% of the area in a graphene sheet would bring the crumpling temperature down to about 1600 K. Thus creating “lacey graphene” via, say, laser ablated holes that remove 85% of the carbon atoms could allow the crumpled regime to be accessed experimentally. We note that the mechanical and electrical properties of free-standing graphene springs with roughly 40% of the material removed were studied in [34].

It is important to note that the observed  $kT_c(s)$ , (2.7), cannot be explained by the effective elastic constants of the perforated sheets  $\kappa_{\text{eff}}(T = 0)$  and  $Y_{\text{eff}}(T = 0)$ . Indeed, as we explain in Supplementary Note 1, the  $T = 0$  bending modulus of the perforated sheet,  $\kappa_{\text{eff}}(T = 0)$ , linearly decays with  $(1 - s)$ . Therefore, if the onset of crumpling were simply determined by  $\kappa_{\text{eff}}(T = 0)$ , one would expect  $T_c$  to be a linear function of  $(1 - s)$ . Instead, as we obtained in Eq. (2.7),  $T_c \sim (1 - s)^{1.93}$ , a result which indicates that non-trivial thermal fluctuation effects are responsible. The effective Young’s modulus  $Y_{\text{eff}}(T = 0)$ , on the other hand, has a complicated dependence on the details of the perforation pattern (see Supplementary Note 1). However,  $Y$  only affects the crumpling temperature as a logarithmic correction, see Eq. (2.9) below. Explaining the observed value of  $c = 1.93(4)$  remains, therefore, a theoretical challenge.

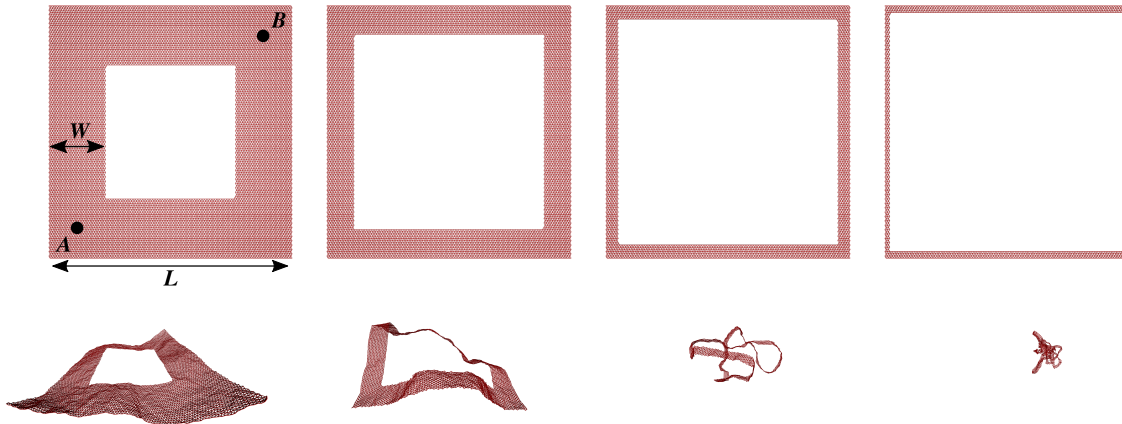
## 2.3 Finite-size scaling

We have seen that cutting holes in a membrane can induce crumpling at much lower temperatures. We have yet to show, however, that this phenomenon quantitatively corresponds to the standard crumpling transition that has been extensively studied for full sheets [5, 7, 14, 15, 29, 41, 45–51]. This can be accomplished by performing a finite-size scaling (FSS) study [52] and finding the universality class of the phase transition. This computation poses two difficulties: on the one hand our simulations cover a very wide range of temperature, rather than concentrating all the numerical effort to increase the precision at the critical region. On the other hand, the presence of the holes creates novel finite-size effects. We begin by considering the FSS of the height of the peak in  $d\mathcal{R}_g^2/d\tilde{\kappa}$ , which diverges as [41]

$$\left. \frac{d\mathcal{R}_g^2}{d\tilde{\kappa}} \right|_{\max} \sim L^{4/d_H+1/\nu}, \quad (2.8)$$

where  $\nu$  is the critical exponent describing a normal-normal correlation length that diverges at the crumpling transition. When considering this equation, it is important to notice that, while the exponent is universal, the algebraic prefactor is not and depends on all the parameters. In particular, for a given finite size, the transitions in Fig. 2.3 seem to be of varying sharpness. However, the values of the critical exponents for the sharpest looking transition (the full membrane) are known from previous work. In the following, we will perform the FSS analysis and a fit to (2.8) only for our most perforated membrane (the rightmost curve in Fig. 2.3). If its critical exponents turn out to be compatible with those of the full sheet, we can conclude that the intermediate curves will be in the same universality class too.

Fig. 2.6 shows the results of this analysis. We obtain  $\theta = 4/d_H + 1/\nu = 2.88(7)$ , to be compared with  $\theta = 2.86(1)$  from a recent dedicated FSS study for the full membrane [41]. Extracting the values of  $d_H$  and  $\nu$  separately is more difficult. In principle, one could compute  $\nu$  by studying the drift in the position of the peak  $T_c^{(L)} \simeq T_c^\infty + AL^{-1/\nu}$ , but this



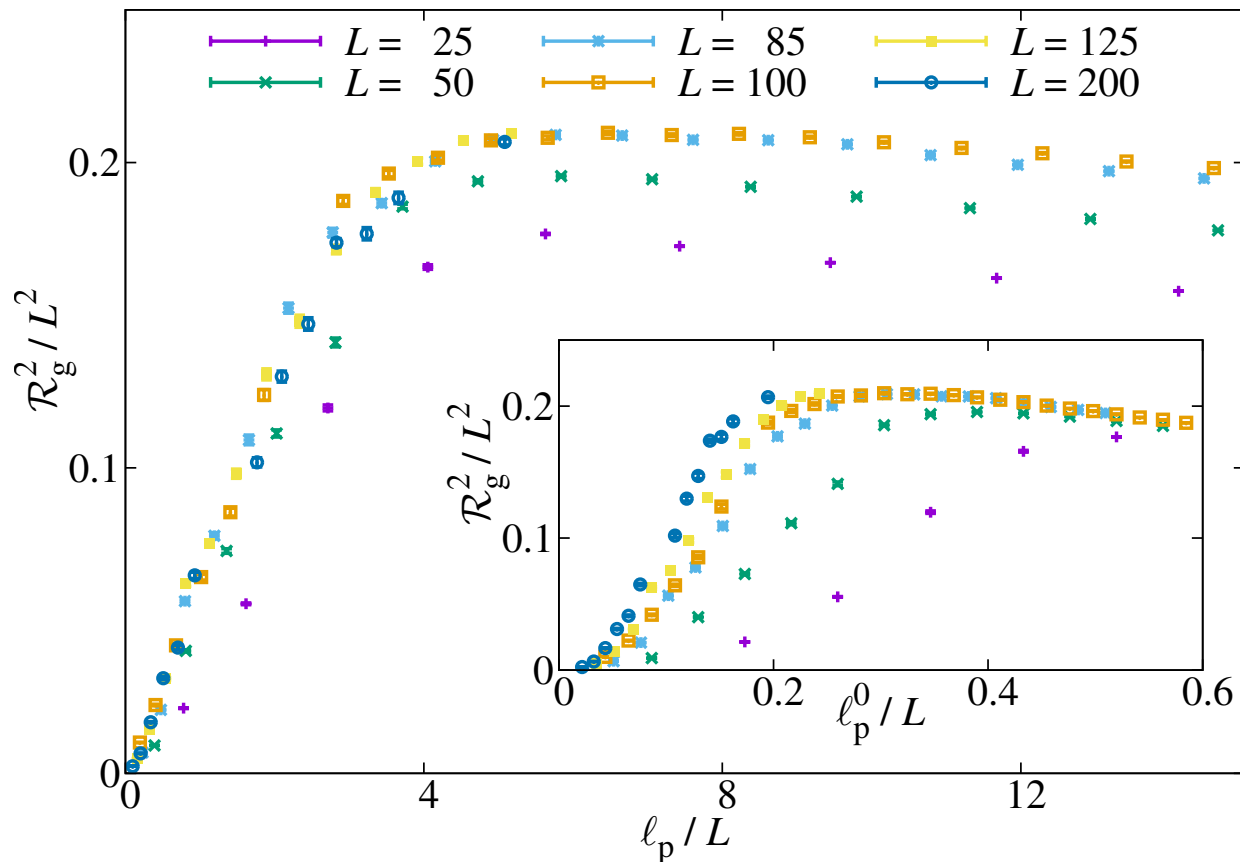
**Figure 2.7:** Crumpling of a thin frame. The top row shows the initial ( $T = 0$ ) configuration for frames of  $L = 100a$  and  $W = 24a, 12a, 6a, 3a$  (left to right). The bottom row shows thermalized configurations (for  $\tilde{\kappa} = 1.25kT$  and  $\epsilon = 1800kT/a^2$ ) for each of these geometries, showing a clear crumpling as the frame width  $W$  is reduced. Points  $A$  and  $B$  of the leftmost frame are used to define an order parameter for crumpling in the text.

has very strong corrections to the leading scaling [41]. Alternatively, one could consider the critical scaling of the specific heat (yielding  $\alpha/\nu$  and hence  $\nu$  from hyperscaling), but in this case one has to include an analytical contribution that introduces an extra fitting parameter:  $C_V = C_a + AL^{\alpha/\nu}$ . Since, unlike for the full membrane [41], we have to discard sizes  $L < 50$  due to finite-size effects, we do not have enough degrees of freedom to obtain a reliable computation of  $\nu$ . We have checked, however, that the value  $\nu = 0.74$  for the standard crumpling transition is consistent with our data (see Supplementary Note 2). Using this estimate of  $\nu$  we obtain  $d_H = 2.62(7)$ . In short, the transition in these perforated membranes is compatible with the universality class of the crumpling transition for pristine sheets, even though its location is shifted downward in temperature by an order of magnitude.

### Crumpling of thin frames

It is illuminating to consider what happens when all perforations are combined to create a thin frame of width  $W$  and overall size  $L$ , e.g., a membrane interrupted by a single large square hole. As shown in Fig. 2.7 (simulations at fixed temperature and  $L$  with varying  $W$ ), there is now a striking crumpling transition as a function of hole size. As an order parameter for this crumpling transition, imagine erecting the normal to these frames at the





**Figure 2.8:** Scaling in thin-frame crumpling. We plot the radius of gyration for frames of different  $L$  and  $W$  against  $\ell_p/L$ , where the persistence length  $\ell_p = 2W\kappa^R(W)/kT$  and the renormalized bending rigidity  $\kappa^R(W)$  are defined in (2.11). The curves for different system sizes collapse when plotted against this scaling variable. The inset shows that neglecting thermal renormalization of the bending rigidity, that is, considering  $\ell_p^0 = 2W\kappa_0/kT$ , leads to a poorer collapse. In all these simulations we have used  $\tilde{\kappa} = 1.25kT$  and  $\epsilon = 1800kT/a^2$ . All error bars (not visible at this scale in most cases) represent the standard error of the mean.

points  $A = (W/2, W/2)$  and  $B = (L - W/2, L - W/2)$ , where we use an  $xy$ -coordinate system superimposed on the frame at  $T = 0$  with origin at the lower left corner. Then, in the flat phase of the frame (left side of Fig. 2.7, when the hole is small), we expect in the limit of large frame sizes,  $\langle \hat{\mathbf{n}}_A \cdot \hat{\mathbf{n}}_B \rangle \neq 0$ . Indeed, in the limit of a vanishingly small hole ( $W \rightarrow L/2$ ), we expect [19]

$$\langle \hat{\mathbf{n}}_A \cdot \hat{\mathbf{n}}_B \rangle_L = 1 - \frac{kT}{2\pi\kappa_0} \left[ \eta^{-1} + \log \left( \frac{\ell_{\text{th}}}{a} \right) + C \frac{kT}{\kappa_0} \left( \frac{\ell_{\text{th}}}{L} \right)^\eta \right], \quad (2.9)$$

where  $C$  is a positive constant of order unity,  $\eta \approx 0.8$  and the thermal length scale is

$$\ell_{\text{th}} = \sqrt{\frac{16\pi^3\kappa_0^2}{3kTY_0}}. \quad (2.10)$$

Thus  $\lim_{L \rightarrow \infty} \langle \hat{\mathbf{n}}_A \cdot \hat{\mathbf{n}}_B \rangle_L \neq 0$ , indicating that the normals on diagonally opposite corners are correlated. In contrast, when the frame is crumpled (right side of Fig. 2.7, when the hole is large), we clearly have  $\lim_{L \rightarrow \infty} \langle \hat{\mathbf{n}}_A \cdot \hat{\mathbf{n}}_B \rangle = 0$ . In the case of square frames, we can estimate where the transition occurs by comparing the frame size  $L$  to the persistence length for thin frames of width  $W$  [19].

$$\ell_{\text{p}} = \frac{2W\kappa^{\text{R}}(W)}{kT}, \quad \kappa^{\text{R}}(W) = \kappa_0 \left( \frac{W}{\ell_{\text{th}}} \right)^\eta. \quad (2.11)$$

Here  $\kappa^{\text{R}}(W)$  is the thermally renormalized bending rigidity. Crumpling out of the flat phase should occur when  $L > \ell_{\text{p}}$ , which suggests a scaling form for the radius of gyration of Eq. (2.4), namely,

$$\mathcal{R}_{\text{g}}^2 = L^2 F(\ell_{\text{p}}/L), \quad (2.12)$$

This scaling ansatz (where crumpling is indicated by the behavior for small  $x$ ,  $\lim_{x \rightarrow 0} F(x) \sim x$ ) is checked for a wide variety of frame dimensions  $L$  as a function of  $W$  in Fig. 2.8, which shows excellent data collapse as  $L$  becomes large. Note that the collapse is not nearly so good if one simply scales with a bare persistence length (inset), indicating that thermal

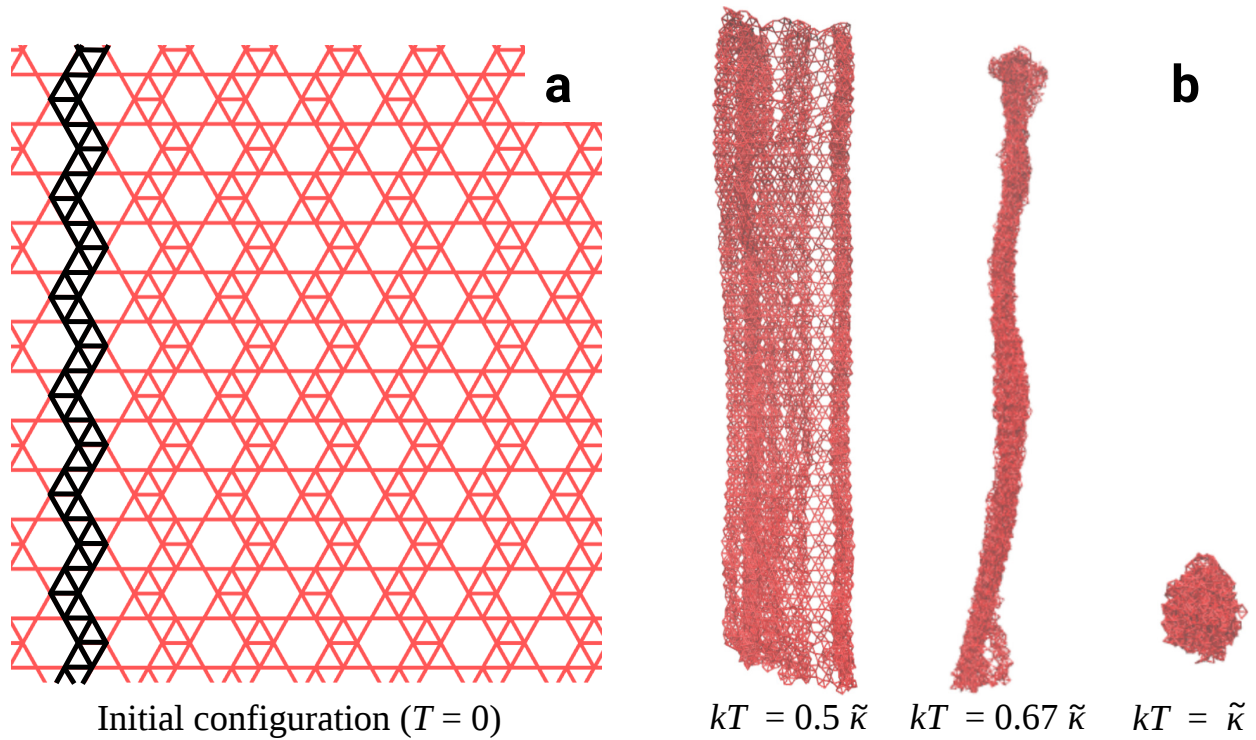
fluctuations play an important role in our simulations. For this problem, it is known that the crumpled phase is robust to distant self-avoidance [53]. Indeed, the crumpled phase only swells slightly with a scaling function in Eq. (2.12) that behaves accordingly to  $F(x) \sim x^{4/5}$  for small  $\ell_p/L$ . Of course, considerably more work would be required to demonstrate convincingly that there is a sharp phase transition in the thermodynamic limit. Here, the nontrivial width-dependent scaling of the thermally renormalized persistence length in Eq. (2.11) suggests that the appropriate limit is  $L, W \rightarrow \infty$ , with fixed  $W(W/\ell_{\text{th}})^n/L$ . In short, this analysis suggests that there could be a novel transition for single frames, where both a crumpled and flat phase would survive in a polymer-like large-size limit. Even if this transition were simply a crossover, we expect a dramatic change in mechanical properties, such as the response to bending, pulling and twisting, when the frame crumples [19].

We note finally that the crumpling temperature for unperforated membranes can be estimated (up to logarithmic corrections) from Eq. (2.9) as  $kT_c \approx 2\pi\eta\kappa_0$ , in approximate agreement with the transition temperature associated with the black curve in Fig. 2.3.

## 2.4 Methods

### Our simulations

We have simulated model (3.1) for sizes ranging from  $L = 25a$  to  $L = 150a$  with molecular dynamics in an  $NVT$  ensemble, using a standard Nosé-Hoover thermostat [54, 55]. All simulations were carried out with the help of the HOOMD-blue package [56, 57]. Smaller sizes (up to  $L = 50a$ ) were simulated on CPUs using a message-passing interface (MPI) parallelization, while for larger systems we have used GPUs. We use a simulation timestep of  $\Delta t = 0.0025$  (in natural units where  $a = m = kT = 1$ ). We start with a flat sheet in the  $xy$ -plane, and add a small random  $z$  component to all the nodes, in order to get the molecular dynamics started. We then follow the evolution for  $2 \times 10^8$  timesteps, discarding the first 10% for thermalization and using a jackknife procedure [52] to estimate statistical errors. Converted into wall-clock time,  $10^8$  steps of a simulation of size  $L = 100a$  (with 11484 nodes, 34023 bonds and 33597 dihedral angles) require about 8 hours of execution



**Figure 2.9:** Two-step crumpling transition in an anisotropic sheet. The zig-zag pattern of approximately vertical struts reinforced by edge-sharing triangles make this structure more rigid in the vertical than in the horizontal direction (see highlighted example in the figure). We show snapshots of thermalized configurations for several temperatures. As  $T$  increases, the anisotropy in the pattern of perforations makes the membrane first fold into a tight cylinder, before crumpling completely. This geometry corresponds to the system labeled Pattern 3 in Fig. 2.3 and in Supplementary Fig. 1. The  $T = 0$  snapshot (a) is a close-up to a  $30a \times 30a$  section of the lattice, while the finite-temperature snapshots (b) show the full  $100a \times 100a$  system.

time on an NVIDIA Tesla K40. Our total simulation time has been the equivalent of  $\approx 5$  months of a single Tesla K40.

## 2.5 Conclusion and Discussion

We have studied the mechanics of thermalized membranes with a dense array of holes and found that the perforations can bring the crumpling temperature into an experimentally accessible regime. From Fig. 2.5, we have  $kT_c/\tilde{\kappa} \simeq A(1-s)^{1.93}$  for the crumpling temperature as a function of the area fraction removed  $s$ , independent of the detailed arrangement and size of our periodic lattice of holes. In addition, we have found that with an anisotropic pattern of perforation one can induce a first partial crumpling at an even lower temperature. Indeed, see Fig. 2.9, a system where the perforations are asymmetric or arranged in such a way that one of the membrane's axes presents less bending resistance will first fold and roll into a very tight cylinder, before crumpling completely. See reviews by Radzihovsky and by Bowick in [32] for a discussion of two-stage crumpling. These observations provide a potential method for bridging the gap between the theoretical expectations for the crumpling transition and the experimentally accessible temperatures.

A subtle issue is our neglect of distant self-avoidance. The nearest-neighbor springs in Eq. (3.1) embody an energy penalty of order  $\epsilon a^2$  when nearest-neighbor nodes overlap, a number which greatly exceeds  $kT$ . Adding a hard sphere excluded volume interaction between second-nearest neighbors would create an entropic contribution to  $\tilde{\kappa}$  of order  $kT$ , which might produce a small shift in the crumpling temperature. The existence of a sharp crumpling transition in unperforated membranes with distant self-avoidance remains unclear at the present time [8, 58]. The presence of a lattice of large holes will certainly reduce the effect of distant self-avoiding interactions, especially when the removed area fraction becomes large. When distant excluded-volume interactions are non-zero but weak, theory predicts a sharp transition between a low-temperature flat phase and a high-temperature crumpled phase with a nontrivial fractal dimension  $d_H \approx 2.5$  [33, 59], qualitatively similar to the findings for perforated membranes presented here. In addition, we have argued for

the existence of a sharp crumpling transition when all perforations are combined to create a thin frame with a single large hole in the center of the membrane. In this case, it is well known that the crumpled ring polymer phase survives the imposition of distant self-avoidance [53]. We hope our results will stimulate allocation of resources (both experimental and computational) that will allow investigations of distant self-avoidance in the presence of a lattice of perforations. Even if distant self-avoidance smears out a sharp crumpling transition, we nevertheless expect qualitatively different mechanical behavior in the regimes identified here for thermalized kirigami sheets.

# Chapter 3

## Anharmonic effects and buckling in thermalized ribbons

### 3.1 Introduction

In this chapter, we conduct a detailed study of the thermalized ribbon clamped at both ends by running extensive molecular dynamics simulations and tuning our system to a bending dominated regime. In contrast to recent work [60] where thermal effects are neglected while designing clamped resonators we propose a simple geometric tuneability incorporating thermal fluctuations as a means to study the fundamental mode and anharmonic effects in these clamped systems. We also propose a thermalized version of Euler buckling as an easy means to achieve a two-state system whose temporal characteristics and responses are controlled using external stress and geometry.

### 3.2 Model

We simulate ribbons of size  $L \times W$ , which have been discretized using an equilateral triangular lattice with each side  $a = 1$ . We use a standard coarse-grained model [40] to compute the total energy in the ribbon. The total Hamiltonian of the system consists of a stretching

and a bending term:

$$\mathcal{H} = \mathcal{H}_{\text{stretch}} + \mathcal{H}_{\text{bend}}, \quad (3.1)$$

Stretching in the system is modeled by imagining the triangle sides as springs of elastic constant  $\epsilon$  and rest length  $a$ :

$$\mathcal{H}_{\text{stretch}} = \frac{1}{2}\epsilon \sum_{\langle i,j \rangle} (r_{ij} - a)^2, \quad (3.2)$$

where the sum is taken over all pair of vertices with a triangle edge between them. The bending energy is computed using the dihedral interaction between the normals:

$$\mathcal{H}_{\text{bend}} = \tilde{\kappa} \sum_{\langle \alpha, \beta \rangle} (1 - \hat{\mathbf{n}}_\alpha \cdot \hat{\mathbf{n}}_\beta). \quad (3.3)$$

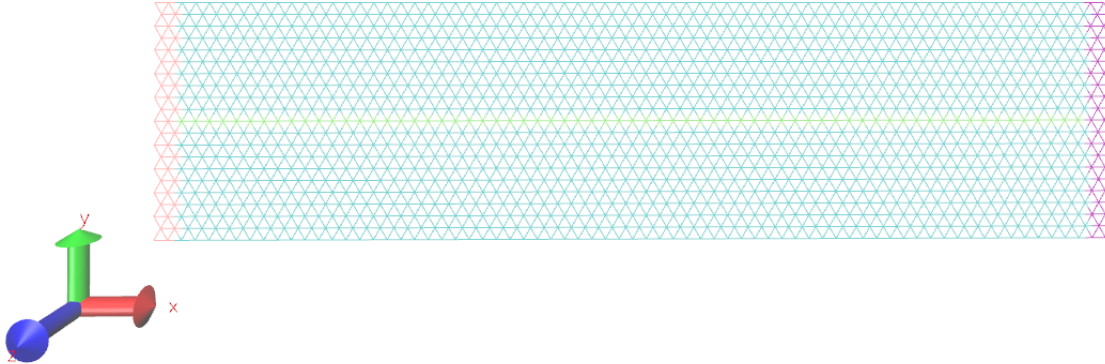
where the sum is taken over all triangle pairs which share a common triangle edge and  $\hat{\mathbf{n}}_\alpha$  is the normal to the triangle  $\alpha$ .

The elastic parameters  $\epsilon$  and  $\tilde{\kappa}$  are directly related to the continuum Young's modulus ( $Y_0 = 2\epsilon/\sqrt{3}$ ) and bare bending rigidity ( $\kappa_0 = \sqrt{3}\tilde{\kappa}/2$ ). Our goal is to study the geometric dependence of thermalized ribbons, we fix the ratio  $\epsilon/\tilde{\kappa} = 1440/a^2$  [39] and vary the system size parameters  $L$  and  $W$ . We choose  $\tilde{\kappa}/kT = 2.5$  in these simulations to be far from the crumpling regime which occurs around  $\tilde{\kappa}/kT \approx 0.5$  for these parameters [39]. For these parameters a sheet of length  $L/a = \mathcal{O}(10^2)$  and same  $W$ , results in a Föppl von Kármán number of

$$\text{vK} = Y_0 L^2 / \kappa_0 \simeq \mathcal{O}(10^7), \quad (3.4)$$



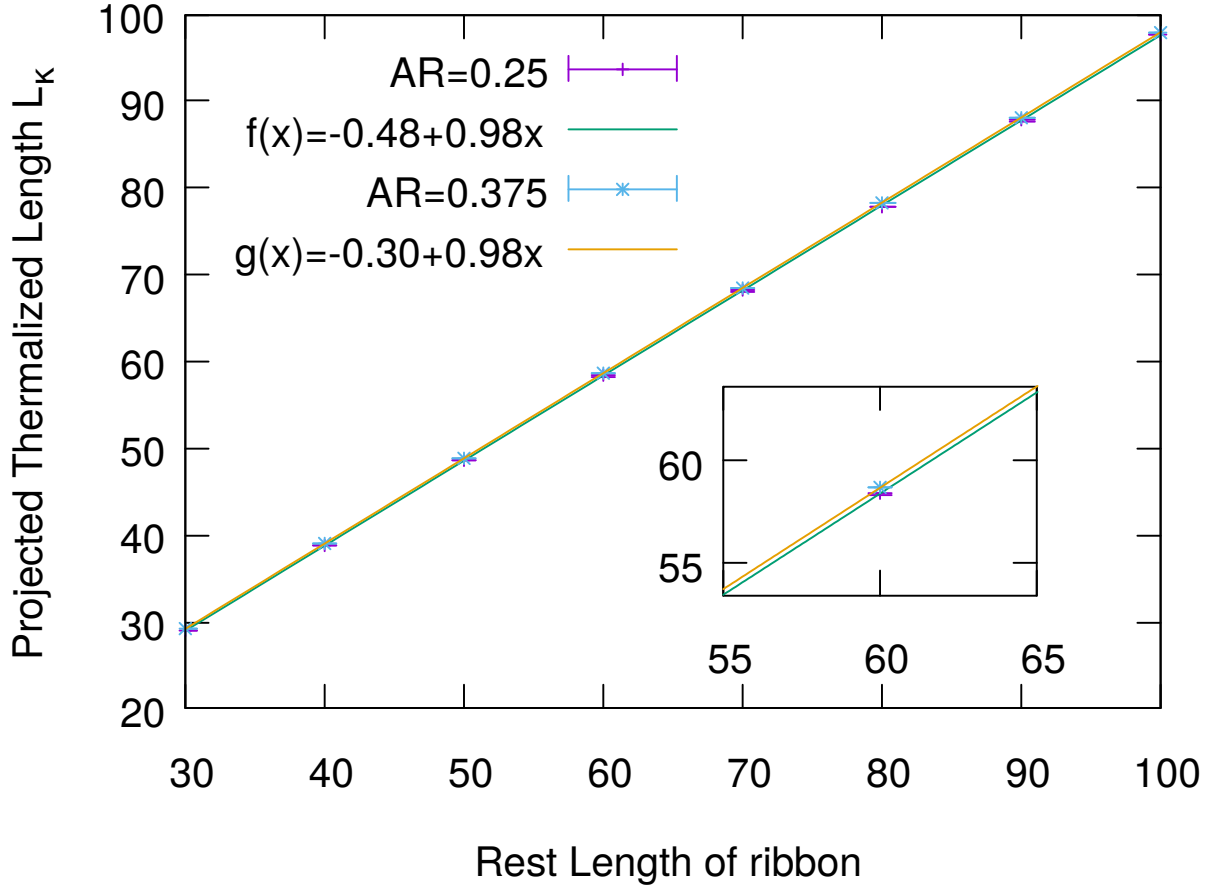
similar to that of a standard A4 sheet of paper.



**Figure 3.1:** Ribbon of dimension  $L = 70a$ ,  $W = 17.3a$  discretized using a triangular lattice with each edge length  $a = 1$ . The light green line segment depicts the backbone of the ribbon. Two columns of nodes marked Red on the left and Violet on the right, denote the nodes which are clamped. Initially, we clamp the Red nodes and allow the violet nodes to slide along the x-direction in the  $z = 0$  plane. All other nodes are free to move in the 3D Euclidean space. The x-position of violet nodes is then averaged over several runs to get the projected thermalized length  $L_{\bar{\kappa}}$  of the ribbon at this temperature. Once we have  $L_{\bar{\kappa}}$  we clamp the Violet nodes at this new position and rerun the simulations, to study long wavelength behavior of thermalized ribbons in tension-free bending dominated regime.

$$\text{AR} = W/L \tag{3.5}$$

We define the aspect ratio of the ribbon as the ratio between the width and length of the ribbon Eq.(3.5). We simulate different aspect ratios of the ribbon, by fixing  $L$  and varying  $W$  over a wide range. We repeat the same by changing  $L$  as well. The simulations are carried out using molecular dynamics in an  $NVT$  ensemble, using a standard Nosé-Hoover thermostat to keep a constant temperature of  $kT = 1$ . All the simulations were carried out with the help of the HOOMD-blue package [56, 57]. We use a simulation timestep of  $\Delta t = 0.002\tau$ , where  $\tau = a\sqrt{m/kT}$  is the reduced time unit ( $m = 1$  is the vertex mass). We follow the dynamics for  $1 \times 10^8$  timesteps for multiple runs ( 10 runs for each parameter set). We record system observables every  $1 \times 10^4$  time steps. We discard the first half of the simulation for thermalization and use a jackknife procedure [52] to estimate statistical errors.



**Figure 3.2:** Projected Thermalized Length  $L_{\tilde{\kappa}}$  for ribbons versus rest length in the range  $L = 30a$  to  $L = 100a$ . Clamping both ends of the ribbon with distance  $L_{\tilde{\kappa}}$  between them ensures that there is no tension in the system.  $L_{\tilde{\kappa}}$  is plotted for two aspect ratios  $AR=0.25$  and  $AR=0.375$ . A straight line fit for the two sets of data shows that they are parallel (see inset) and lower  $AR$  has a shorter  $L_{\tilde{\kappa}}$  for the same rest length of ribbon.

### 3.3 Tension free clamping of ribbons

We want to study dynamics of cantilevers clamped at both ends. Since clamping induces a tension if the system is stretched or compressed we need to find the stress free state to study the dynamics. We choose our simulation parameters to be close to graphene  $\epsilon/\tilde{\kappa} = 1440/a^2$  which makes bending easily compared to stretching.

We aim to extract the tension-free state of the clamped ribbon, for which the following procedure is applied. The ribbons are initially clamped at one end and the other end is free to slide in the  $x$ -direction in  $z = 0$  plane as explained in Fig. 3.1. This is done to extract

the projected thermalized length of the ribbon (we will call this  $L_{\bar{\kappa}}$  from here-on). At any given temperature ribbons will experience thermal fluctuations which will reduce its length when compared with its  $T = 0$  *rest length*. Allowing the ribbon to slide in the x-direction at one end allows the system to adjust to its natural length and we extract the projected thermalized length ( $L_{\bar{\kappa}}$ ) by averaging the slider x-position over several runs as described in Fig. 3.1. This procedure is done for various ribbon rest length's ranging from  $L = 30a$  to  $L = 100a$  and the projected thermalized length ( $L_{\bar{\kappa}}$ ) is extracted for each one as shown in Fig. 3.2. Once we have found the projected thermalized length ( $L_{\bar{\kappa}}$ ) of the ribbon, we clamp both ends of the ribbon with the distance between them equal to this new projected length ( $L_{\bar{\kappa}}$ ) and we re-run the simulations again for all. From here on we will be double clamping the ribbon.

### 3.4 Effects of double clamping

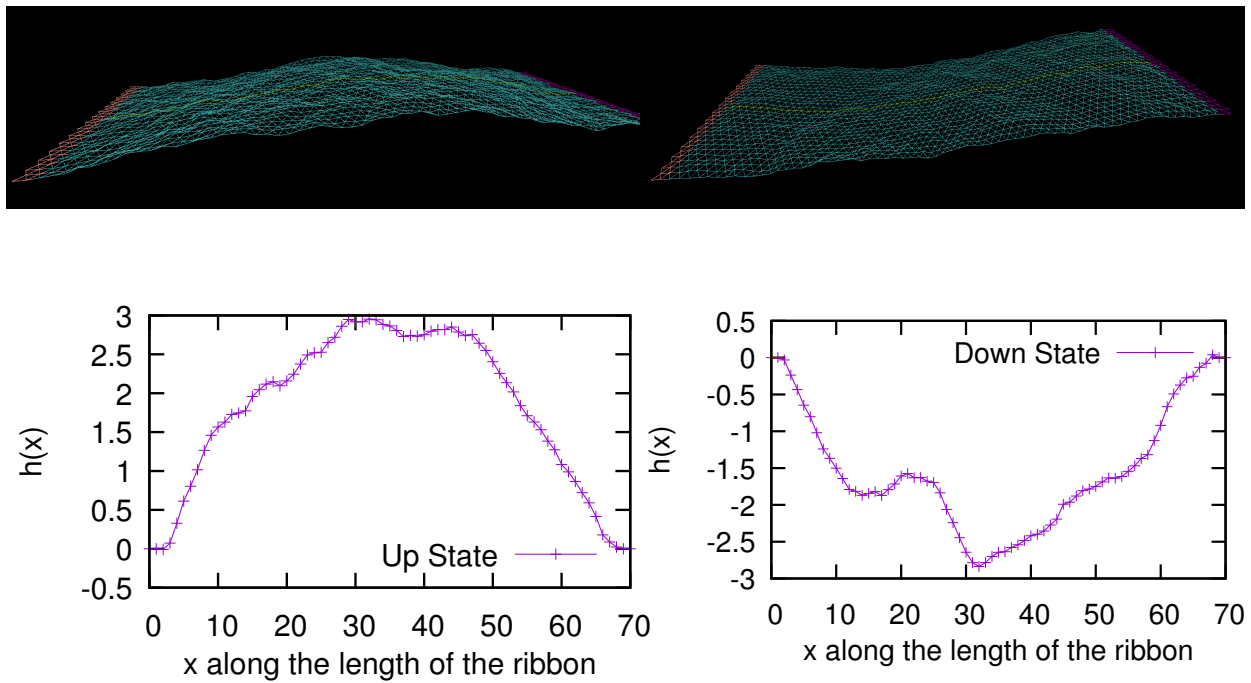
By double clamping at ( $L_{\bar{\kappa}}$ ), we intend to tune any induced tension out of our system. This is important since we want to explore system dynamics in the bending dominated regime. To verify, that we are indeed in a tension-free regime, we look at the height of the ribbon averaged along the width of the ribbon (height of nodes along a column are averaged) in the Fourier space. From theory, we know that the Fourier transform is as follows [19]:

$$\langle |\tilde{h}(q)|^2 \rangle = k_B T / (\sigma q^2 + \kappa q^4), \quad (3.6)$$

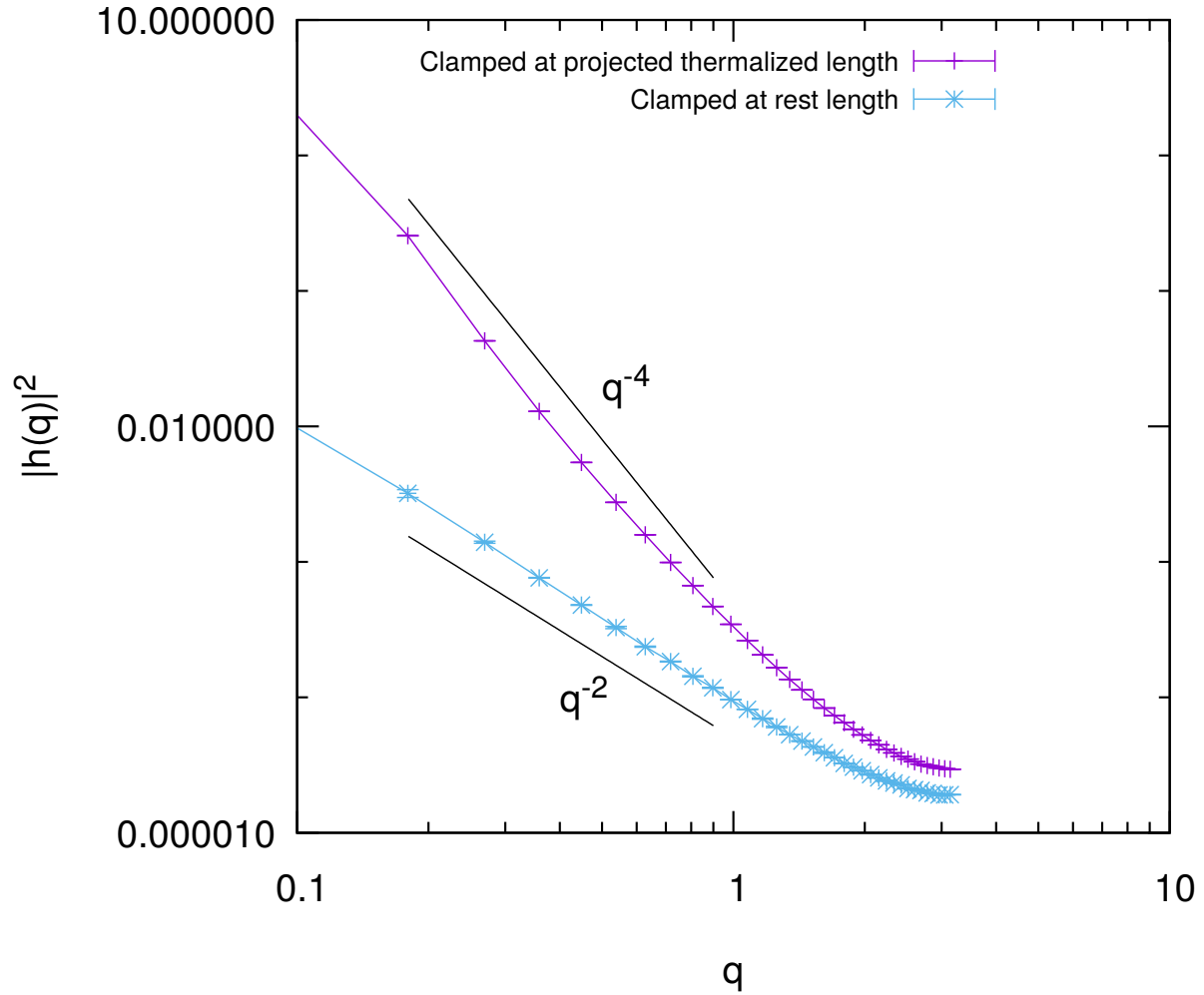
where  $\sigma$  is the stress and  $\kappa$  is the bending rigidity.

We will compare two different clamping scenario's:

- Ribbon clamped with the distance between the clamped ends equal to the  $T = 0$  length of the ribbon. We will call this the *rest length* of the ribbon. At a non-zero temperature this system is stretched.



**Figure 3.3:** Top: Snapshots of the ribbon in Up and Down state. Bottom: Height of the ribbon is averaged along the width (average height of nodes in a column in the  $y$ -direction) to get  $h(x)$ . Fourier transform of  $h(x)$  gives  $\tilde{h}(q)$  and this activity is done for all sanpshots over all runs, to evaluate  $|\tilde{h}(q)|^2$  with errors.



**Figure 3.4:** Fitting  $|\tilde{h}(q)|^2$  obtained from clamping ribbon at its projected thermalized length  $L_{\bar{\kappa}}$  (violet) and rest length (ribbon length when  $T=0$  in blue) to the function  $1/(Sq^2 + Bq^4)$ . Rest length clamping curve is dominated by  $q^{-2}$  stretching term whereas  $L_{\bar{\kappa}}$  clamping curve is dominated by  $q^{-4}$  bending term. By clamping the ribbon at its  $L_{\bar{\kappa}}$  length we have successfully tuned any induced tension out of the system. ( $S = \sigma/kT$  and  $B = \kappa/kT$ )

- Ribbon clamped with the distance between the clamped ends equal to the projected thermalized length  $L_{\tilde{\kappa}}$ .

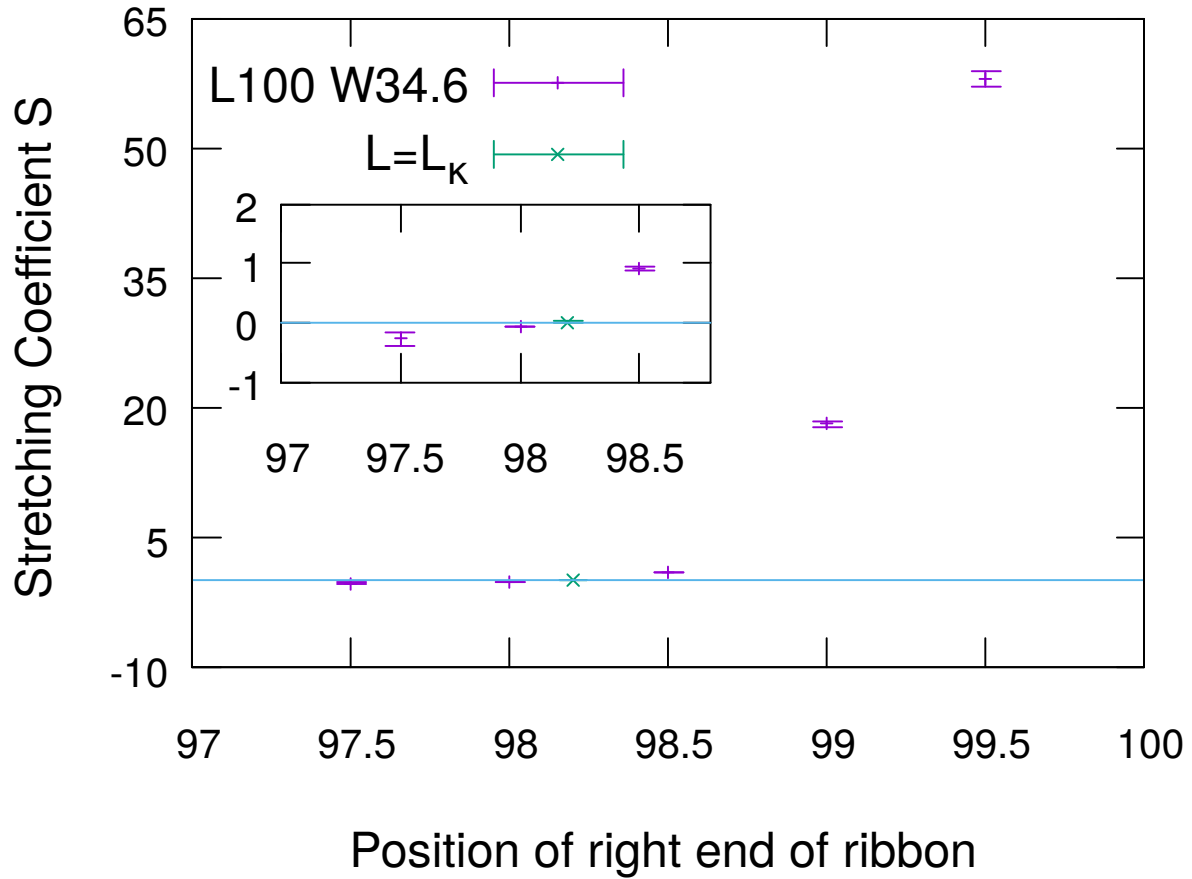
Fig. 3.3 shows the height averaged along the width( $x$ ) of the ribbon  $h(x)$  for one snapshot during the simulation and take its Fourier transform to get  $\tilde{h}(q)$ . We calculate  $\langle |\tilde{h}(q)|^2 \rangle$  from simulation data for ribbons clamped at projected thermalized length ( $L_{\tilde{\kappa}}$ ) and compare the same with thermalized ribbons clamped at its rest length. As shown in Fig. 3.4, we find that the ribbon clamped at  $L_{\tilde{\kappa}}$  fits nicely to  $q^{-4}$  for long wavelengths which is the dominant bending term in theoretical expression (Eq. (3.6)). On the other hand, the case where the ribbon is double clamped at its rest length, stretching dominates and this can be seen from the simulation data fitting to  $q^{-2}$ . Hence, by double clamping at  $L_{\tilde{\kappa}}$  we have successfully tuned out any induced tension from the system and we are nicely poised to study the long wavelength effects in this bending dominated regime.

Notice that the fitting parameter  $S$  (stretching coefficient) for the double clamped ribbon at its rest length is positive. If one were to compress the system such that the new clamping length is less than  $L_{\tilde{\kappa}}$ , this fitting parameter  $S$  should become negative. A plot of the fitting parameter  $S$  versus the various compressed length should then cross zero at  $L_{\tilde{\kappa}}$ . We make this consistency check next.

We double clamp the ribbon over a wide range of lengths (both compressed and stretched) and perform the fitting procedure for each as done in Fig. 3.4. We track the stretching coefficient  $S$  which should be positive for stretched cases and negative otherwise. The clamped length at which  $S = 0$  is the tuned length which exhibits zero induced tension. Tuned length obtained in this method should match the projected thermalized length. Fig. 3.5 shows that this indeed is true.

### 3.4.1 Bending Rigidity renormalization

It has been shown in previous work [19] that bending rigidity for thermalized ribbons renormalize depending on width given that width is much greater than the thermal length



**Figure 3.5:** Ribbon of rest length  $L = 100a$  and  $W = 34.6a$  is gradually compressed and clamped at different compressed lengths. Stretching coefficient  $S$  is evaluated for each and plotted against the compressed length (right end of the ribbon), left end is clamped at  $x = 0$ . Stretching coefficient becomes negative when compression crosses the projected thermal length of this system (marked by green point). The inset shows a zoomed section of the plot near  $S = 0$  which clearly shows the cross-over from positive  $S$  (stretched case) to negative (compressed case). The green dot is obtained from the sliding procedure explained in Fig. 3.1.

scale of the system  $l_{th}$  given by Eq. 3.7

$$l_{th} = \sqrt{\frac{16\pi^3\kappa_0^2}{3k_B T Y_0}}, \quad (3.7)$$

where  $\kappa_0$  is the bare bending rigidity and  $Y_0$  is the continuum Young's modulus.

Thermal length scale for the material parameters used by us gives  $l_{th} \sim a$  where  $a$  is the lattice spacing.

The persistence length of thermalized ribbons is dependent on width  $W$  as shown in Eq.(3.8).

$$l_p = \frac{2W\kappa_R(W)}{k_B T}, \quad (3.8)$$

The renormalized bending rigidity depends on width as per Eq.(3.9).

$$\kappa_R(W) = \kappa_0(W/l_{th})^\eta, \quad (3.9)$$

where exponent  $\eta \approx 0.80 - 0.85$  for thermalized sheets.

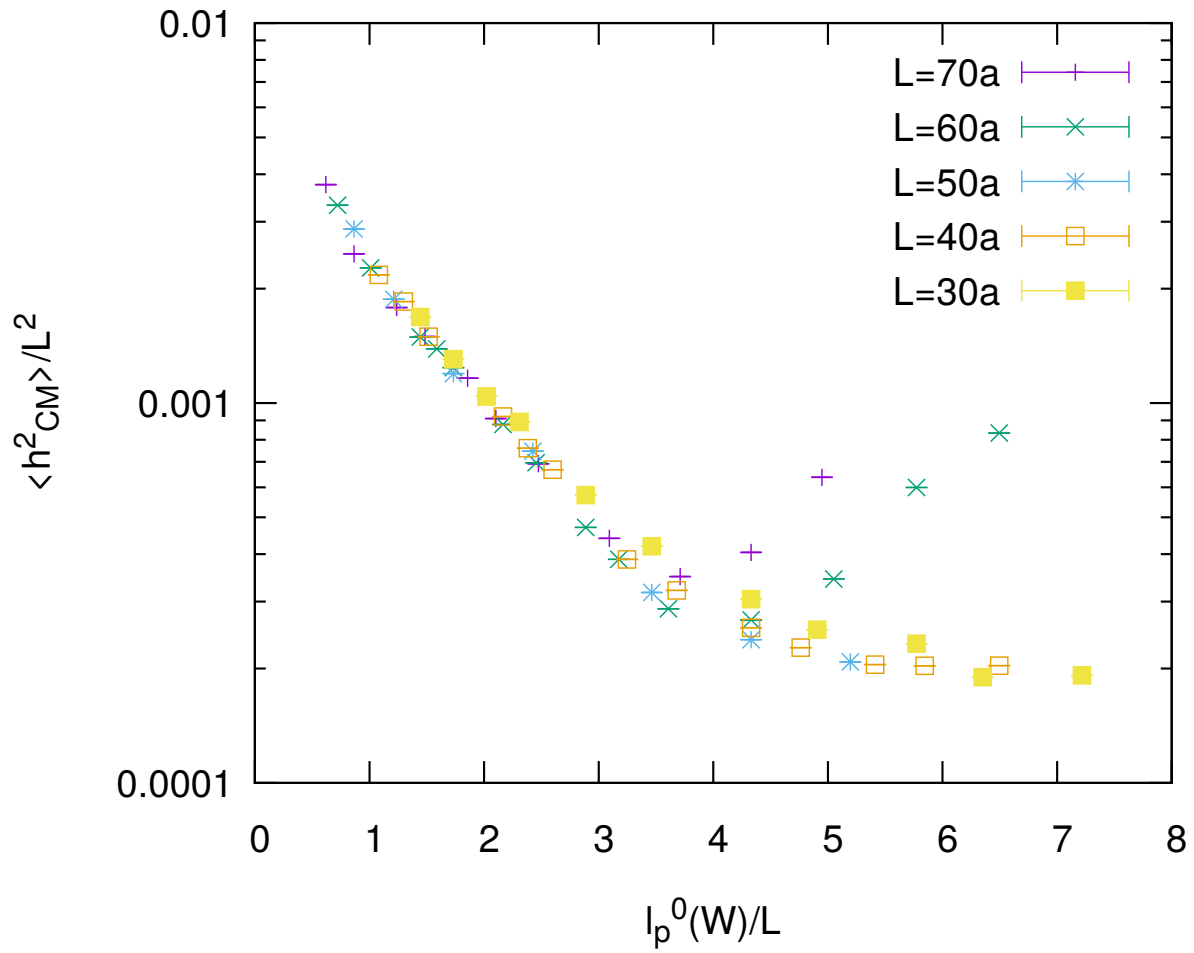
To study the effects of renormalized bending rigidity on our clamped ribbons we look at the average square height of the center of mass (CM) of the system as a function of persistence length. We first look at a naive picture, where we don't account for renormalization. The naive persistence length using bare bending rigidity comes out as Eq.(3.10).

$$l_p^0 = \frac{2W\kappa_0}{k_B T}, \quad (3.10)$$

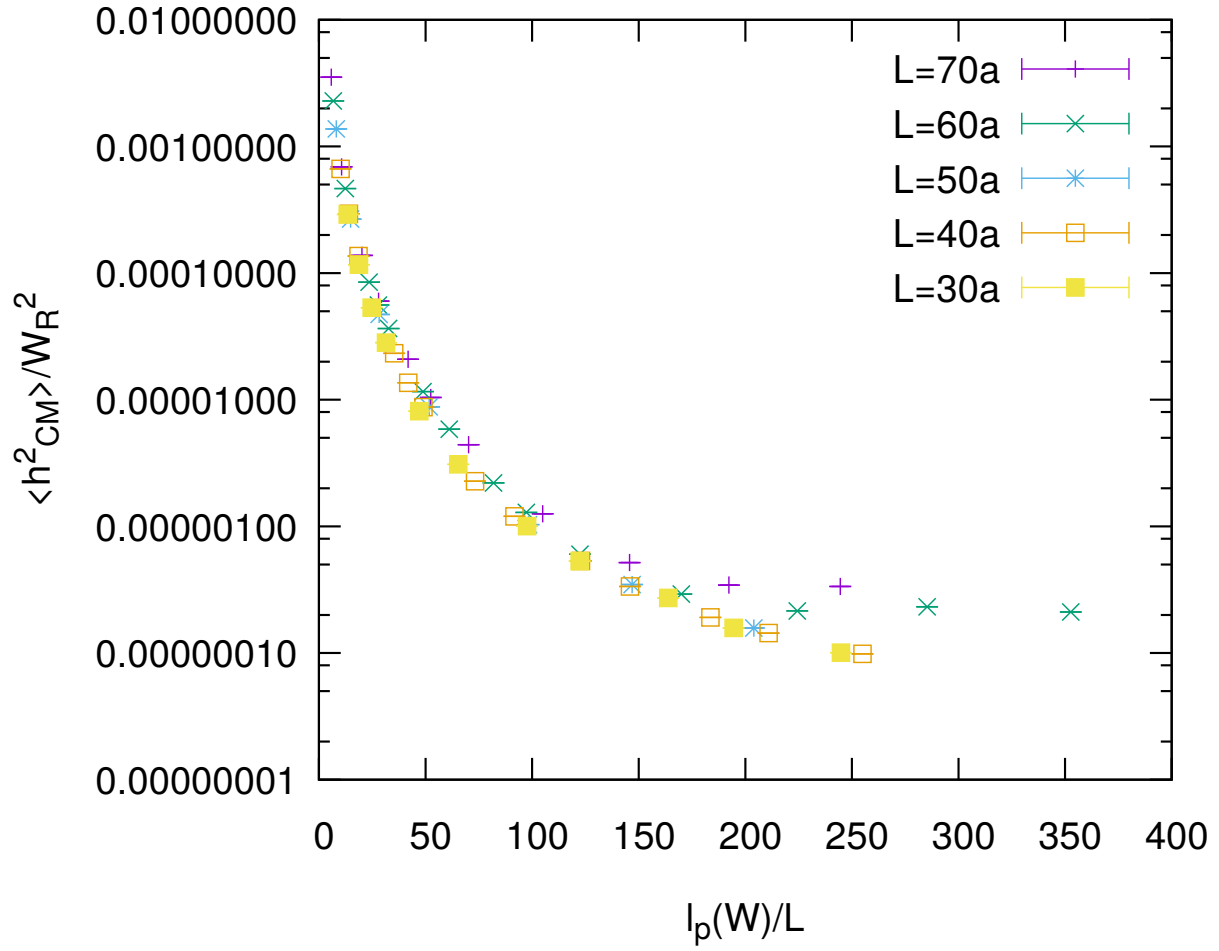
where we have used the bare bending rigidity  $\kappa_0$  instead of  $\kappa_R(W)$ .

We note here that it is difficult to extract  $\eta$  from this data. Fig. 3.6 shows a poor





**Figure 3.6:**  $\langle h_{CM}^2 \rangle$  is normalized using  $L^2$  versus  $I_p^0(W)/L$ . As one can see the plot doesn't collapse for longer ribbons.



**Figure 3.7:**  $\langle h_{CM}^2 \rangle$  is normalized using  $W^{2+2\eta}/l_{th}^{2\eta}$  versus  $l_p(W)/L$ . The plot collapses best when renormalization of the width is taken into account.

collapse when renormalization is not taken into account. But Fig. 3.7 where we have used  $\eta = 0.8$  by hand shows a better collapse. This gives some evidence that  $\eta \neq 0$  and that there is some thermal renormalization taking place.

### 3.4.2 Two State System

We will now look closely at the oscillatory motion of the system which has been double clamped at its  $L_{\bar{\kappa}}$  length. We first track the height( $z$ -coordinate) of the center of mass (CM) of the system as a function of the simulation time MD steps ( $h_{CM}(t)$ ). As can be seen from Fig. 3.8, the CM of the system exhibits two distinct states Up and Down (above and below the  $z = 0$  plane). Top panel of Fig. 3.3 shows snapshots from the simulation of the ribbon in Up and Down state.

### 3.4.3 Time Autocorrelation function

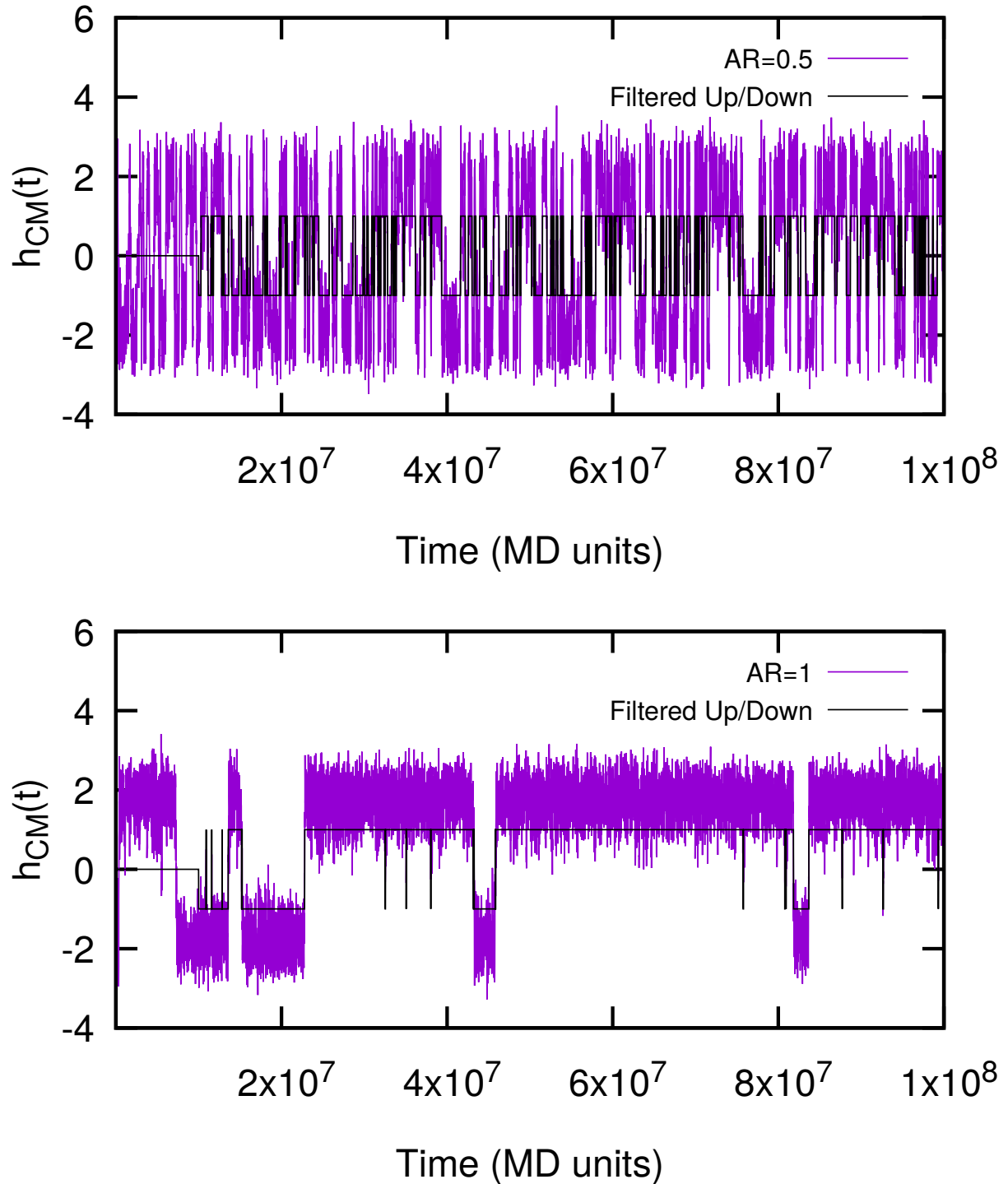
One can now extract the average time the systems CM spends in the Up/Down state. First, we look at the time height auto-correlation function of the CM ( $h_{CM}(t)$ ).

$$C(t) = \langle h_{CM}(t)h_{CM}(t + \Delta t) \rangle, \quad (3.11)$$

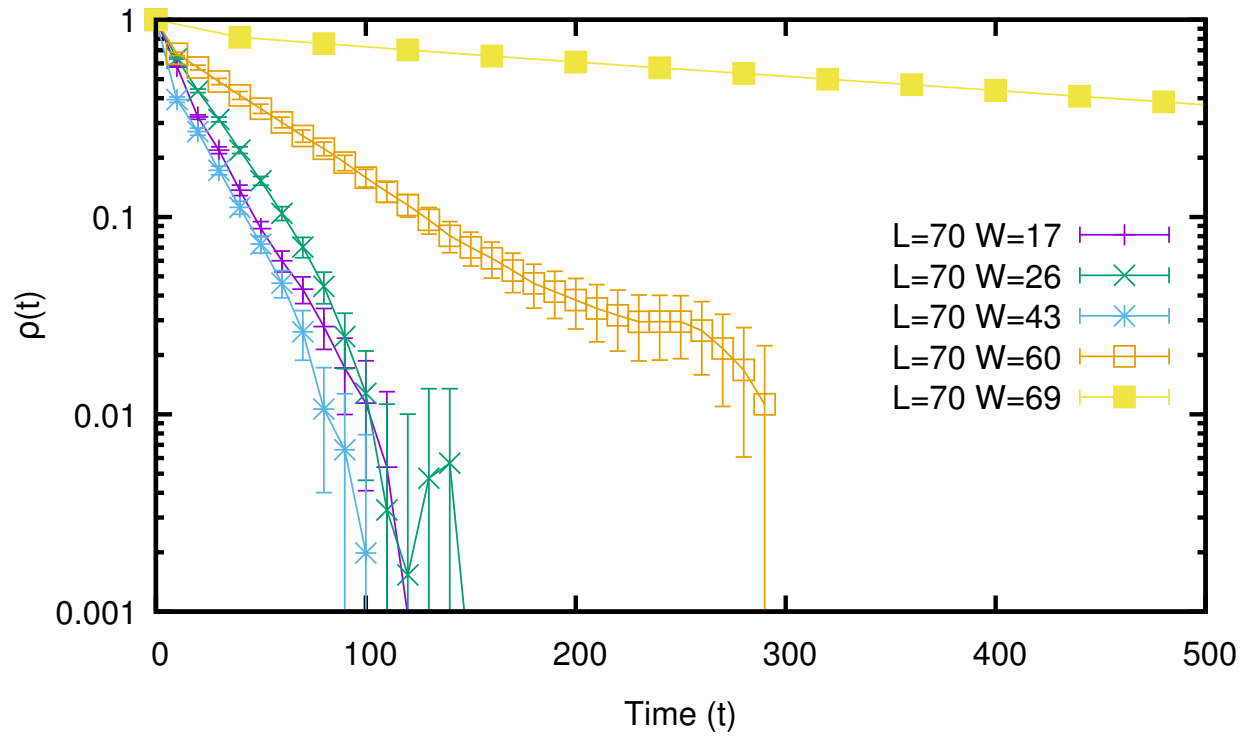
$$\rho(t) = C(t)/C(0), \quad (3.12)$$

For sufficiently long times the autocorrelation function  $\rho(t)$  given by Eq. (3.11) & (3.12) decays exponentially [52].

$$\rho(t) \sim \exp(-t/\tau_{ac}), \quad (3.13)$$



**Figure 3.8:** Top: Height time series of CM of ribbon  $h_{CM}(t)$  ( $L = 70a$ ,  $W = 34.6a$ ). First 10% of data is discarded to allow for equilibration. A filter is applied to extract the time spent by CM in the Up/Down state. Bottom: CM height time series for ( $L = 70a$ ,  $W = 69a$ ) corresponding  $AR \sim 1$ . Two states are distinctly visible in this case. Both ribbons are clamped at their projected thermalized length  $L_{\bar{r}}$ .



**Figure 3.9:** Time autocorrelation function for ribbon length  $L = 70a$  with different widths. As the width increases the survival time increases drastically.

where  $\tau_{ac}$  measures the slowest time scale in the system. In this case, it measures the time taken by the CM to overcome the energy barrier in going from the Up state to Down state or vice versa. In other words,  $\tau_{ac}$  measures the average time the systems CM spends in the Up/Down state.

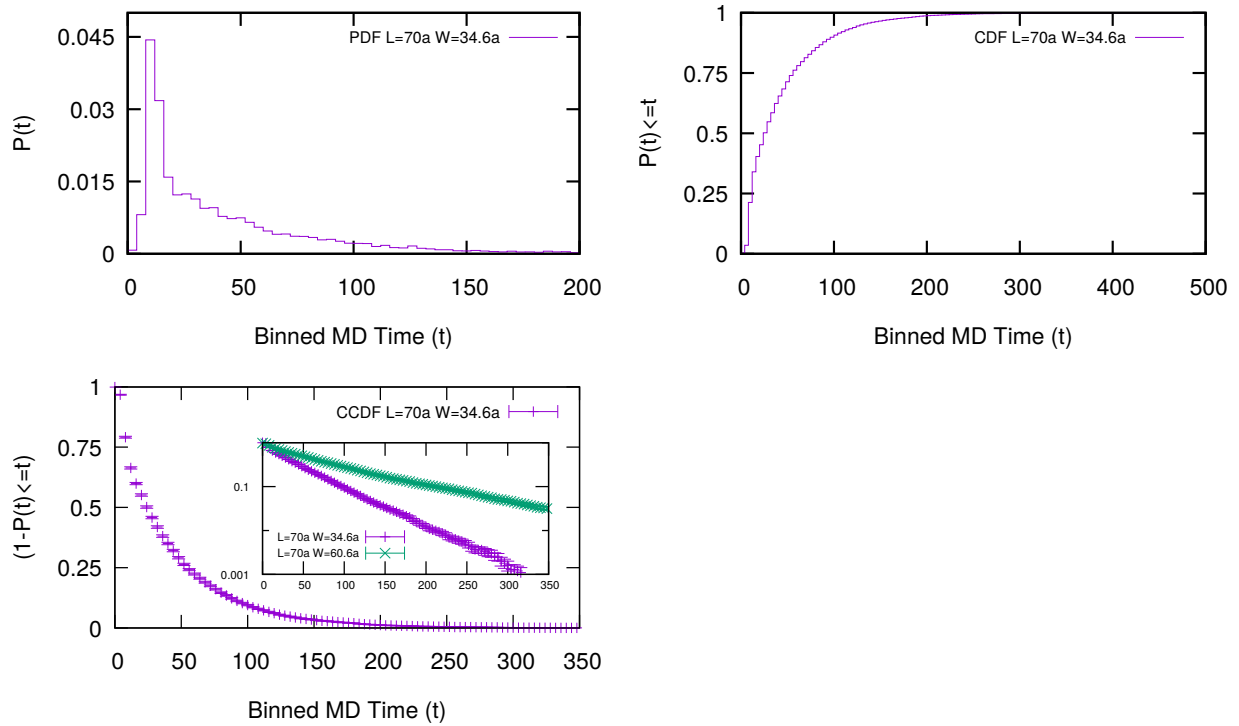
Several aspect ratio's are probed by fixing  $L = 70a$  and changing the  $W$  of the ribbon. We fit this simulation data to a decaying exponential Eq. (3.13) to extract the  $\tau_{ac}$ . Fig. 3.9 depicts the data for various aspects ratio's, showing a dramatic increase in  $\tau_{ac}$  as the aspect ratio of the system tends to one ( $W$  comparable to  $L$ ). So, ribbons with  $AR \sim 1$  tend to spend a considerable amount of time in Up/Down state, indicating that the energy barrier required to flip states in this system is very high compared to systems with  $AR < 1$ . A look at the bottom panel of Fig. 3.8 which corresponds to  $h_{CM}(t)$  for a single run for the parameter set in yellow in the top panel shows that the time spent in Up/Down state has definitely increased. Details, of the fitting procedure can be found in the Appendix 3.6.

### 3.4.4 Integrated Survival Time

The above method is dependent on a fitting range and it is not easy to get three decades of data for  $\rho(t)$  in Eq. (3.12) for all aspect ratio's (difficulty increases as AR tends to values greater than 1, as shown by the yellow plot in top panel Fig. 3.9). As  $AR \sim 1$  the system starts showing metastability by staying Up or Down for durations comparable to simulation run time. Measuring  $\tau_{ac}$  using the fitting procedure becomes difficult.

Instead, we adopt a more direct method for capturing the system's CM Up/Down time. We first filter the height time series of the CM  $h_{CM}(t)$  of the ribbon from simulation data as shown in the top Fig. 3.8. Using this filtering process we are able to extract the individual Up/Down times in each run which we will now define as the *survival time* of the system.

We will work with simulation data for the parameter set ( $L = 70a, W = 34.6a, \tilde{\kappa}/kT = 2.5$ ) to explain this procedure in detail. Fig. 3.8 shows a time series for  $L = 70a$  and  $W = 34.6a$  where we have discarded top 10% to account for equilibration time. We do this for all the runs and bin these individual survival times to generate the probability distribution.



**Figure 3.10:** Post filtering the CM height time series into Up and Down survival times from multiple runs (10 runs for each parameter set), probability distribution function (PDF) is generated. Top: Shows the PDF to be a Poisson distribution for ribbon of dimensions  $L = 70a$  and  $W = 34.6a$  for one run.

Middle: Cumulative Distribution Function (CDF) for the above PDF. Bottom: Complementary Cumulative Distribution Function (CCDF = 1 - CDF) is generated from CDF. Area under this curve is defined as the Integrated Survival Time, since  $\int_0^\infty \exp(-x/\tau) dx = \tau$ . Note that the error bars, in this case, are generated using bootstrap method [52] since the underlying distribution is not gaussian.

If there is an energy barrier associated with the Up/Down transition, survival times will be Poisson distributed. If it is a Poisson distribution,  $1-\text{CDF} = \exp(-t/\tau_{survival})$ , where CDF is the Cumulative Distribution Function and  $\tau_{survival}$  is the survival time in the Up/Down state.

$$\int_0^{\infty} \exp(-t/\tau_{survival}) dt = \tau_{survival}, \quad (3.14)$$

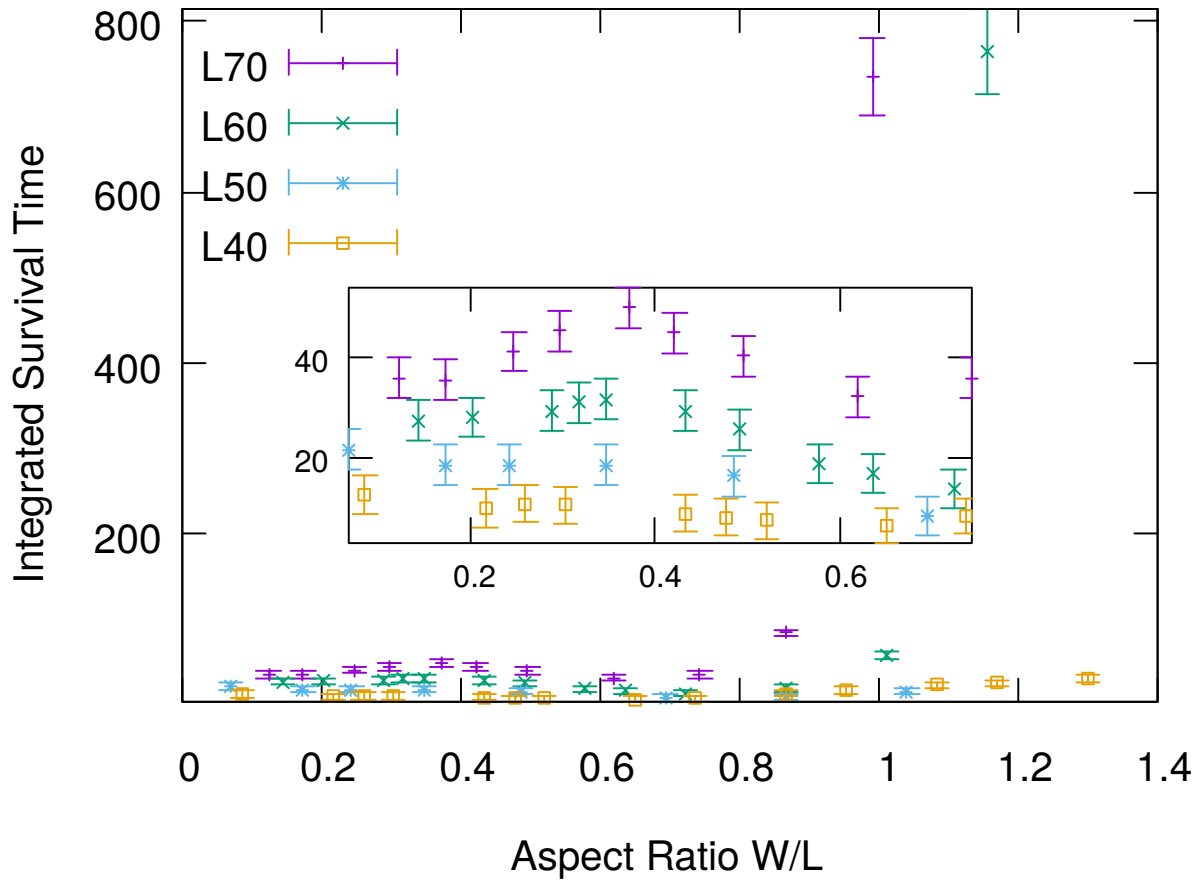
Integrating  $(1-\text{CDF})$  will give  $\tau_{survival}$  as shown in Eq. 3.14 which we define as the *Integrated Survival Time*. Top and middle panel of Fig. 3.10 show the PDF and CDF for ribbon survival times. Inset in the third panel shows CCDF ( $1-\text{CDF}$ ) plotted against binned time in semi-log plot which fits a straight line showing its decaying exponentially as per Eq. (3.13). Since the underlying distribution is not gaussian we cannot use Jack Knife for error estimation. Instead, we use bootstrap [52] method, by generating bootstrap samples with randomly selecting with repetition from the pool of observed survival times. This process gives us the integrated survival time for a given set of parameters along with the error. Using this procedure Integrated Survival Time is estimated for different geometric dimensions of ribbon to study the two state and buckling phenomenon in detail. In Appendix. 3.7 we show that Integrated Survival Time computed using this process agrees with  $\tau_{ac}$  calculated using the autocorrelation function in Sec.3.4.3

We now study the oscillatory behavior of ribbons of different ARs by changing system size  $L$ . Fig. 3.11 shows the integrated survival time  $\tau_{survival}$  jumps sharply as the AR approaches values near 1. Inset shows that  $\tau_{survival}$  at a fixed AR increases for larger system sizes. So, just by playing with the geometric dimensions of the ribbon one can derive completely different oscillatory behavior from these thermally driven clamped systems.

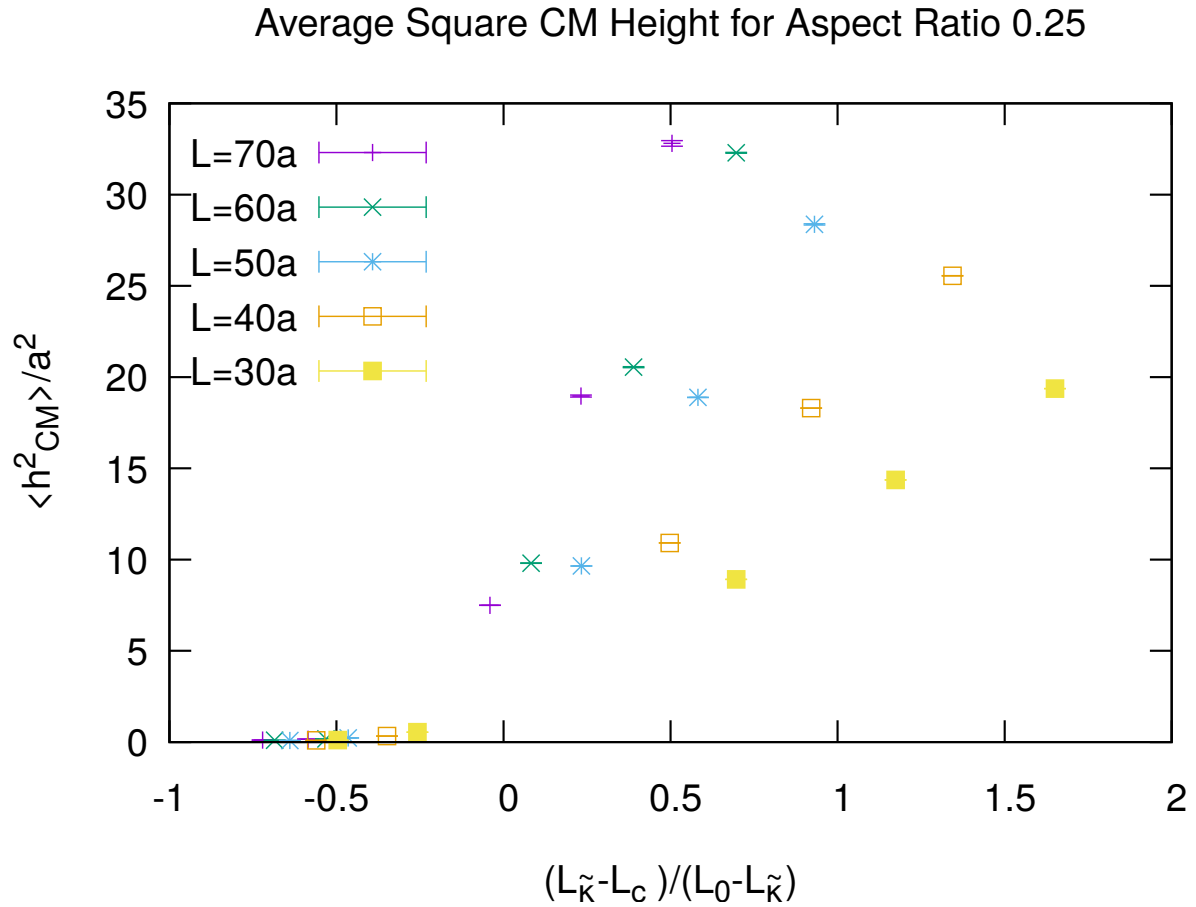
### 3.4.5 Thermalized version of Euler buckling

In classical plate theory when a plate is compressed it buckles and moves out of the plane. It remains in this state as long as the compressive stress is present and reverts back to flat

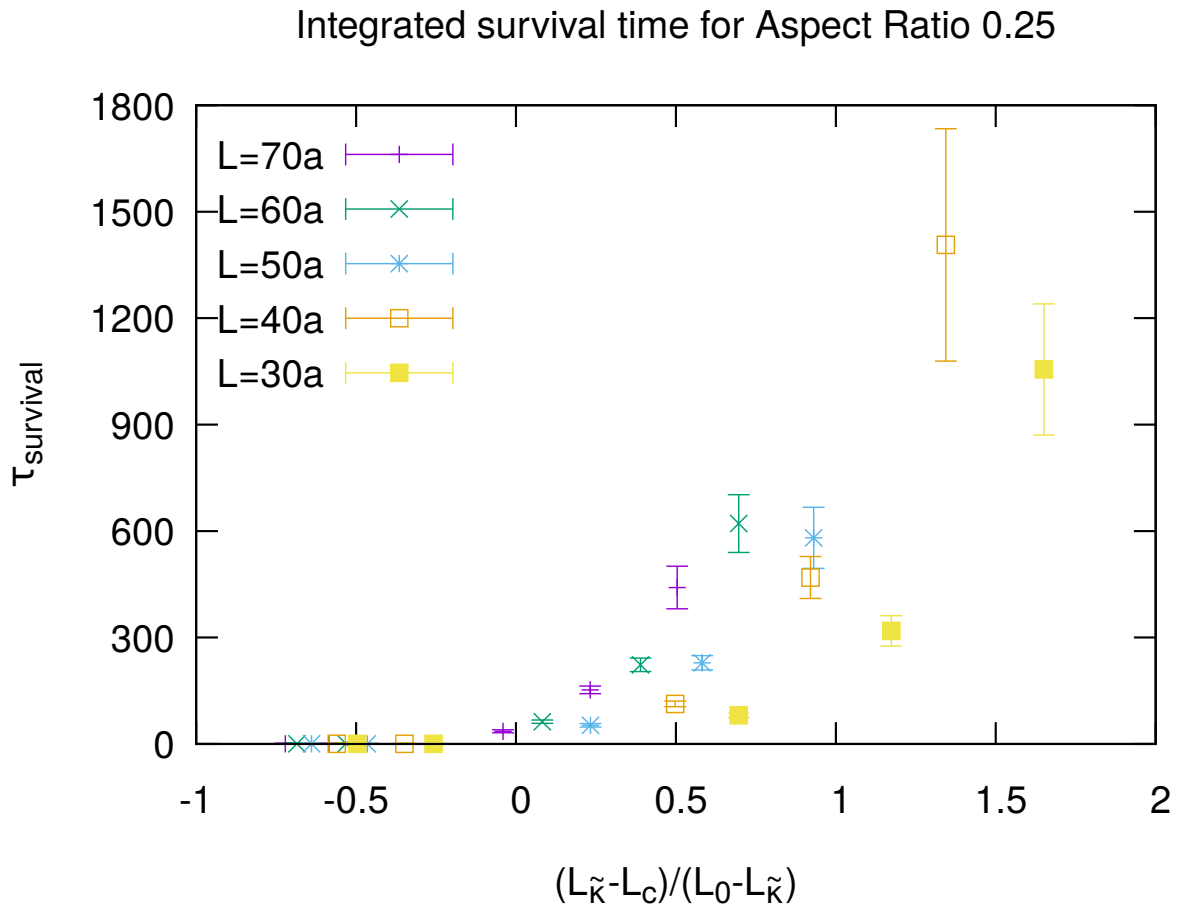




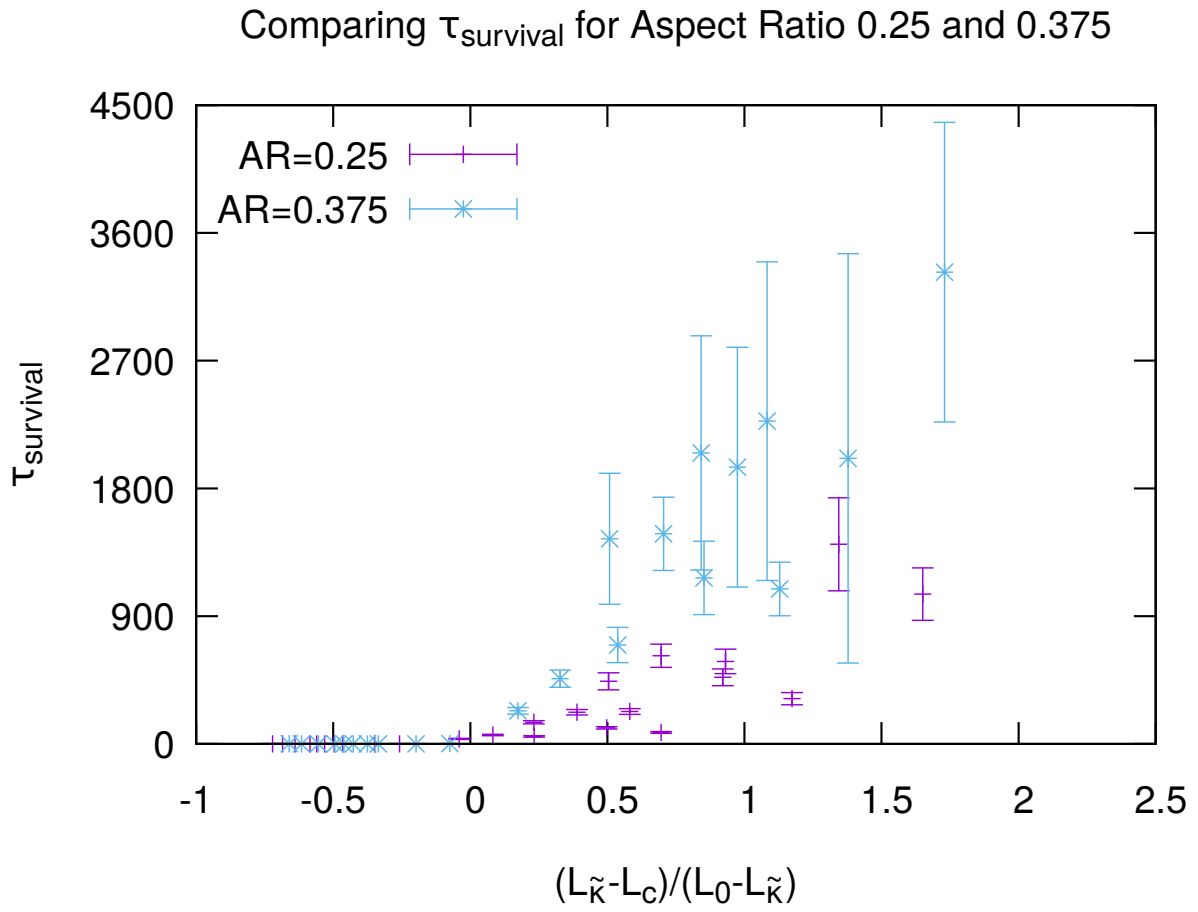
**Figure 3.11:** Integrated survival time  $\tau_{survival}$  is studied for different aspect ratio's over a range of ribbon lengths by varying width. Integrated survival time for larger system sizes ( $L = 60a$  and  $L = 70a$ ) jumps sharply around  $AR \sim 1$ . Inset depicts  $\tau_{survival}$  values for lower aspect ratio's showing clearly that at a given aspect ratio  $\tau_{survival}$  increases for larger system sizes.



**Figure 3.12:** The average squared height of the ribbon CM is plotted against relative compression for different ribbon geometries. In the stretched case (negative compressions) the CM height is almost zero. Compressing the ribbon beyond its  $L_{\tilde{\kappa}}$  causes the ribbon CM to jump immediately, which increases with compression and at a fixed relative compression is greater for the longer ribbon. The ribbons studied here all had fixed aspect ratio of 0.25.



**Figure 3.13:**  $\tau_{\text{survival}}$  as a function of relative compression but fixed AR=0.25. Notice, that  $\tau_{\text{survival}}$  takes off for cases where the ribbon is compressed beyond its  $L_{\tilde{r}}$ . Looking at relative compression values at 0.5,  $\tau_{\text{survival}}$  for longer ribbons is higher, indicating  $\tau_{\text{survival}}$  may diverge for sufficiently longer ribbons.



**Figure 3.14:** Comparing integrated survival time for ribbons with AR 0.25 and 0.375. Clearly, AR=0.375 ribbons exhibit persist in Up/Down state longer than ribbons with lower AR.

state once the stress is removed (given that the compressive forces were not so high as to induce permanent bent deformation). Looking at the result from our previous discussion in Section 3.4.4, one sees indications of Euler buckling in double clamped ribbons at  $L_{\tilde{\kappa}}$  for higher aspect ratio's. So, it should be possible to induce thermalized Euler buckling in system sizes with lower aspect ratio by applying compressive stress and clamping the system at a length lower than its projected thermalized length  $L_{\tilde{\kappa}}$ .

$$\text{RC} = (L_{\tilde{\kappa}} - L_c)/(L_0 - L_{\tilde{\kappa}}), \quad (3.15)$$

$$\text{Compression} = L_{\tilde{\kappa}} - L_c, \quad (3.16)$$

where  $L_c$  is the distance between clamped ends of the ribbon after some compression,  $L_0$  is the rest length of the ribbon and  $L_{\tilde{\kappa}}$  is the projected thermalized length.

We work with two fixed ARs and define *relative compression* ( $RC$ ) given by Eq.3.15. Several runs are made for different system sizes for the two fixed aspect ratio's ( $AR = 0.25$  and  $AR = 0.375$ ). Positive compression values represent compression and negative values represent stretching in the system.  $RC = 0$  means we are clamped at the projected thermalized length ( $L_{\tilde{\kappa}}$ ) of the ribbon.

We first look at the average square height of the system CM  $\langle h_{CM}^2 \rangle / a^2$  against  $RC$ . Fig. 3.12 shows that when the system is compressed beyond its projected thermalized length  $L_{\tilde{\kappa}}$  there is an immediate jump in the average square height of the system CM which increases further if the compression is increased. This is akin to what one sees in the classical Euler buckling where the out of plane deflection of the CM of the system increases as the compressive stress is increased. On the contrary, the average height square for the stretched case is almost zero as one would expect.

We will need to further show that once the CM of the system deflects significantly out

of the plane on compression it does so as long as the system is compressed. To do that, we compute the integrated survival time  $\tau_{survival}$  of the system as a function of relative compression. Fig. 3.13 shows that the  $\tau_{survival}$  increases for all system sizes as one increases compression. Concentrating on a fixed relative compression one sees that systems with larger  $L$  have a higher  $\tau_{survival}$ . In Fig. 3.11 comparing the two aspect ratio's shows that  $\tau_{survival}$  is greater for the larger aspect ratio.

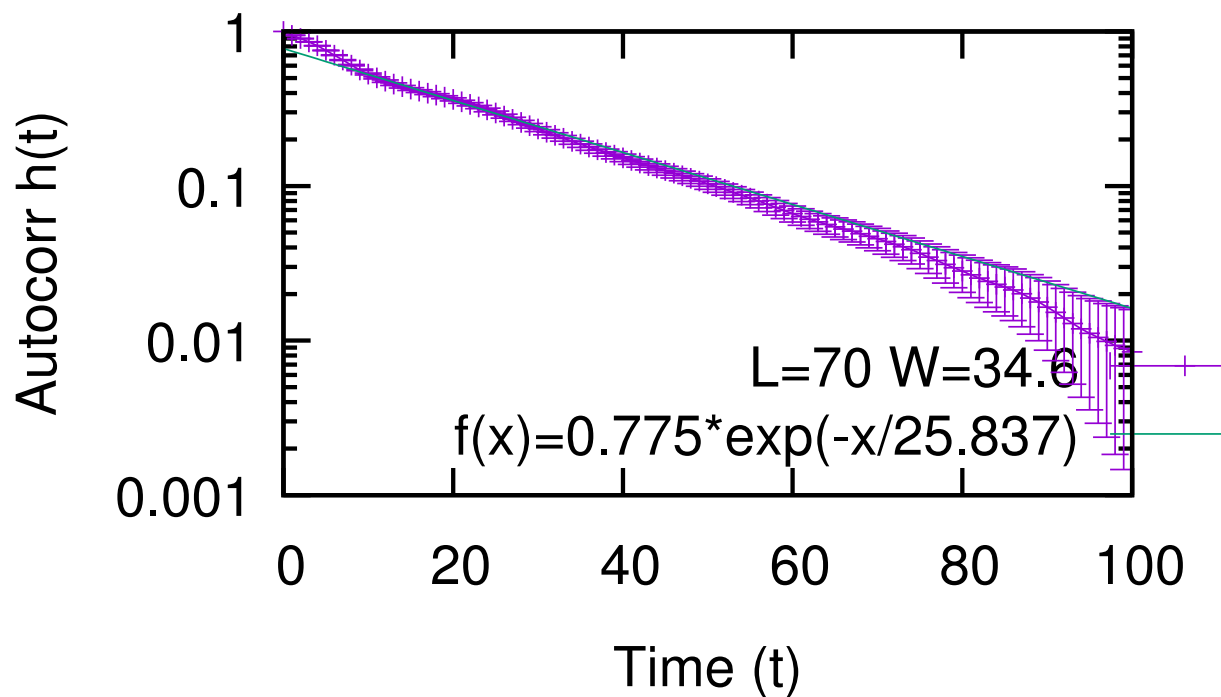
### 3.5 Conclusion and Discussion

Our results demonstrate the importance of system geometry in thermalized ribbons. Just by playing around with the system sizes one can build a thermally induced two state system with control over the duration of the states. Here we have shown that the projected thermalized length is the relaxed stress-free state for these thermalized ribbons and this provides an approach to experimentalist to tune out induced tension in systems like graphene cantilevers.

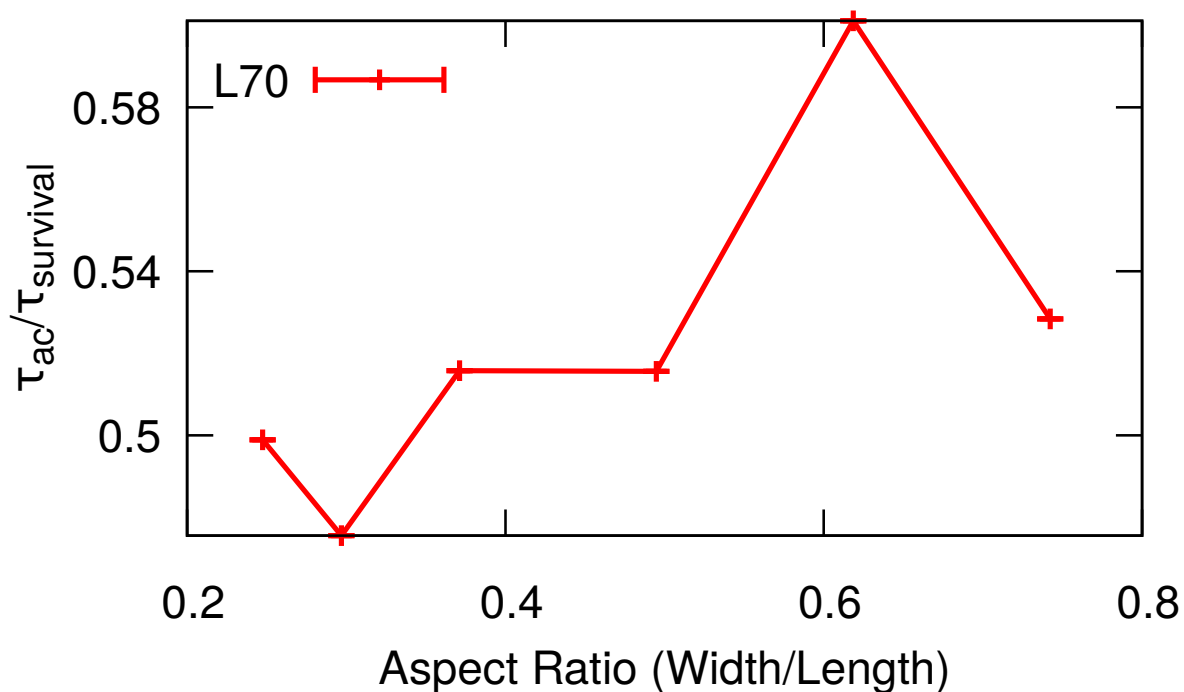
We have also shown that ribbons undergo a thermalized version of Euler buckling and external stresses in addition to system AR can be used to get different oscillatory behavior from these thermally driven systems.

### 3.6 Appendix A

As mentioned in the section on two states, we use time auto-correlation function of the CM height time series to evaluate the survival time of the system in Up/Down state. We record the CM height time series for each run for a given set of parameters and take the Fourier transform of CM height time series  $h_{CM}(t)$  minus the average height for that run. We then take the square of the modulus of the Fourier transform  $|\tilde{h}(q)|^2$  and apply inverse Fourier transform (Weiner-Khinchin theorem) to get the time autocorrelation. Error bars are estimated using Jack Knife procedure by computing autocorrelation for multiple runs and using them to build jack knife blocks. Autocorrelation is normalized using  $C(0)$ .



**Figure 3.15:** Ribbon of rest length  $L = 70a$  and  $W = 34.6$  is clamped at its projected thermalized length  $L_{\bar{r}}$  and its Up/Down survival time evaluated using the autocorrelation of height time series. In this semi-log plot, we fit a straight line and survival time in, this case is, 25.837 MD time steps.



**Figure 3.16:** Comparing the two methods for survival time estimation

For sufficiently, long times the autocorrelation function  $C(t)$  decays exponentially [52]. Hence, we next fit the time autocorrelation in a semi-log plot to a straight line in the range which has the least slope (slowest time scale of the system). The range is chosen by eye (in this case we fit between 10 and 60) and so the procedure is not systematic. The process outcome is depicted in Fig. 3.15 for a system of size  $L = 70a$  and  $W = 34.6a$ .

## 3.7 Appendix B

To show that the Integrated Survival Time estimated in Sec. 3.4.4 is equivalent to the autocorrelation method of Sec. 3.4.3. We compare  $\tau_{ac}$  and  $\tau_{survival}$  in Fig. 3.16.  $\tau_{ac}/\tau_{survival}$  turns out to be around 0.5 as per definition.



# Chapter 4

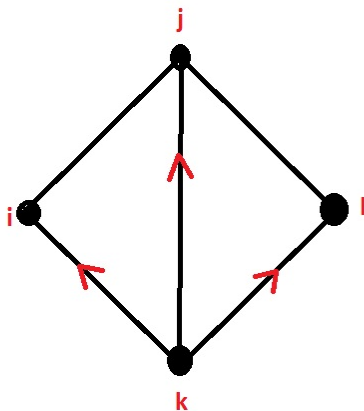
## Simulation methods

### 4.1 Introduction

In this chapter, we discuss the simulation methods used to study the thermalized sheets and ribbons. In particular, we discuss Molecular Dynamics simulation and derive the force experienced by each node in a triangulated sheet. The simulation itself is applied using both CPU and GPU code. In-house GPU code was written in CUDA C to apply custom boundary conditions to thermalized ribbons like sliding in the x-direction. Subsequently, a new release of HOOMD-Blue enabled features for applying custom boundary conditions and it was used to run the simulations in both CPU and GPUs. Computing facility for the research group namely the SMATTER cluster was used to run most of the CPU simulation runs. GPU runs were made with the help of David in his NVIDIA Tesla K80 graphics card and Cierzo supercomputer in Spain. Simulation output was visualized using VMD (Visual Molecular Dynamics) [61]. Finally, we also take a Monte Carlo approach to generate the dynamics of the thermalized sheet using the Metropolis algorithm.

### 4.2 Molecular Dynamics

As mentioned in previous chapters the thermalized sheets are modeled using coarse-graining approach using a triangular lattice of equilateral triangles of side length  $a = 1$ . In this sec-



**Figure 4.1:** Neighboring triangular plaquettes with indexed lattices

tion, we derive the force at each lattice site. Thermalized elastic sheets experience stretching and bending forces. Stretching force will be modeled using a ball spring model and bending force will be evaluated using the angle between two neighboring triangular plaquettes.

### 4.2.1 Derivation of Bending Force

Bending force at each lattice point is derived by taking the negative derivative of the bending energy at each lattice point. Looking at Fig.4.1 one will notice that for each pair of neighboring triangles, the nodes fall into two categories, they are either lying on the common edge or they don't. So, while deriving the bending force contribution at a particular node we will need to consider both this scenarios.

$$E_{\text{Bend}} = \tilde{k}(1 - \cos \Theta_{IJ}) \quad (4.1)$$

where  $\Theta_{IJ}$  is the angle between the normal's on the two triangular faces and  $\tilde{k}$  is the bending rigidity.

Fig.4.1 illustrates the two triangular faces and their respective normals. Bending force has two contributions depending on whether the lattice site where force is being calculated

lies on the common edge or not. We will derive each separately and bring them together in the end.

### Bending Force Term I

First, force is calculated for lattice site  $k$  by wiggling it and holding other three lattice sites  $i, j$  and  $l$  stationary.

$$\vec{F}_k = -\nabla_k \sum_{IJ} \tilde{k} (1 - \cos \Theta_{IJ}) \quad (4.2)$$

$$\vec{F}_k = -\tilde{k} \nabla_k \sum_{IJ} \left( 1 + \frac{r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}}{|r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}|} \cdot \frac{r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}}{|r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}|} \right) \quad (4.3)$$

where  $r_{jk}^{\vec{}} = r_j^{\vec{}} - r_k^{\vec{}}$

$$\vec{F}_k = -\tilde{k} \nabla_k \sum_{IJ} \frac{r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}}{|r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}|} \cdot \frac{r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}}{|r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}|} \quad (4.4)$$

Using identity,

$$\nabla(\vec{a} \cdot \vec{b}) = (\nabla \vec{a}) \cdot \vec{b} + \vec{a} \cdot (\nabla \vec{b}) \quad (4.5)$$

Note that  $\nabla \vec{a}$  is a rank 2 tensor.

#### Notation:

- Indices  $i, j, k$  and  $l$  label the lattice sites.
- Greek letters  $\alpha, \beta$  and  $\gamma$  will be used to label the tensor indices i.e. the respective  $x, y$  and  $z$  components of a vector.
- $\nabla_k \vec{a}$  amounts to taking the derivative  $(\frac{\partial}{\partial X_\alpha}) a_\beta$  where  $X_\alpha \equiv r_{k\alpha}$ .

Comparing Eqn.4.4 and Eqn.4.5,  $a = \frac{r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}}{|r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}|}$  and  $b = \frac{r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}}{|r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}|}$

Looking at rank 2 tensor from Eqn.4.4:

$$\frac{\partial a_{\beta}}{\partial r_{k\alpha}} = \frac{|r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}| \frac{\partial (r_{jk}^{\vec{}} \times r_{ik}^{\vec{}})_{\beta}}{\partial r_{k\alpha}} - (r_{jk}^{\vec{}} \times r_{ik}^{\vec{}})_{\beta} \frac{\partial |r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}|}{\partial r_{k\alpha}}}{|r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}|^2} \quad (4.6)$$

Next we will take the derivatives of terms in numerator of the Eqn.4.6 starting with  $\frac{\partial (r_{jk}^{\vec{}} \times r_{ik}^{\vec{}})_{\beta}}{\partial r_{k\alpha}}$ .

$$\frac{\partial (r_{jk}^{\vec{}} \times r_{ik}^{\vec{}})_{\beta}}{\partial r_{k\alpha}} = \frac{\partial \epsilon_{\beta\mu\nu} (r_{jk}^{\vec{}})^{\mu} (r_{ik}^{\vec{}})^{\nu}}{\partial r_{k\alpha}} = \epsilon_{\beta\mu\nu} \frac{\partial (r_{jk}^{\vec{}})^{\mu}}{\partial r_{k\alpha}} (r_{ik}^{\vec{}})^{\nu} + \epsilon_{\beta\mu\nu} \frac{\partial (r_{ik}^{\vec{}})^{\nu}}{\partial r_{k\alpha}} (r_{jk}^{\vec{}})^{\mu} \quad (4.7)$$

Eqn.4.7 can be further simplified by observing that

$$\frac{\partial (r_{jk}^{\vec{}})^{\mu}}{\partial r_{k\alpha}} = -\delta_{\alpha}^{\mu} \quad (4.8)$$

$$\frac{\partial (r_{ik}^{\vec{}})^{\nu}}{\partial r_{k\alpha}} = -\delta_{\alpha}^{\nu} \quad (4.9)$$

$$\frac{\partial (r_{jk}^{\vec{}} \times r_{ik}^{\vec{}})_{\beta}}{\partial r_{k\alpha}} = -\epsilon_{\beta\mu\nu} \delta_{\alpha}^{\mu} r_{ik}^{\vec{}}{}^{\nu} - \epsilon_{\beta\mu\nu} \delta_{\alpha}^{\nu} r_{jk}^{\vec{}}{}^{\mu} = -\epsilon_{\beta\alpha\nu} r_{ik}^{\vec{}}{}^{\nu} - \epsilon_{\beta\mu\alpha} r_{jk}^{\vec{}}{}^{\mu} \quad (4.10)$$

Simplifying  $\frac{\partial |r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}|}{\partial r_{k\alpha}}$  appearing in Eqn.4.6 using Levi-Civita formulation of vector cross product by introducing new indices  $\theta$  and  $\phi$ .

$$(r_{jk}^{\vec{}} \times r_{ik}^{\vec{}})_{\eta} = \epsilon_{\eta\theta\phi} r_{jk}^{\vec{}}{}^{\theta} r_{ik}^{\vec{}}{}^{\phi} \quad (4.11)$$

$$\frac{\partial |\vec{r}_{jk} \times \vec{r}_{ik}|}{\partial r_{k\alpha}} = \frac{\partial \sqrt{(\vec{r}_{jk} \times \vec{r}_{ik})_\eta (\vec{r}_{jk} \times \vec{r}_{ik})^\eta}}{\partial r_{k\alpha}} = \frac{\frac{\partial (\vec{r}_{jk} \times \vec{r}_{ik})_\eta}{\partial r_{k\alpha}} (\vec{r}_{jk} \times \vec{r}_{ik})^\eta + (\vec{r}_{jk} \times \vec{r}_{ik})_\eta \frac{\partial (\vec{r}_{jk} \times \vec{r}_{ik})^\eta}{\partial r_{k\alpha}}}{2 |\vec{r}_{jk} \times \vec{r}_{ik}|} \quad (4.12)$$

Plug Eqn.4.10 and Eqn.4.11 into Eqn.4.12 to further simplify.

$$\frac{\partial |\vec{r}_{jk} \times \vec{r}_{ik}|}{\partial r_{k\alpha}} = - \left( \frac{(\epsilon_{\eta\alpha\phi} r_{ik}^\phi + \epsilon_{\eta\theta\alpha} r_{jk}^\theta) (\vec{r}_{jk} \times \vec{r}_{ik})^\eta + (\vec{r}_{jk} \times \vec{r}_{ik})_\eta (\epsilon^{\eta\alpha\phi} (r_{ik})_\phi + \epsilon^{\eta\theta\alpha} (r_{jk})_\theta)}{2 |\vec{r}_{jk} \times \vec{r}_{ik}|} \right) \quad (4.13)$$

$$= \frac{-2(\epsilon_{\eta\alpha\phi} r_{ik}^\phi + \epsilon_{\eta\theta\alpha} r_{jk}^\theta) (\vec{r}_{jk} \times \vec{r}_{ik})^\eta}{2 |\vec{r}_{jk} \times \vec{r}_{ik}|} = \frac{-(\epsilon_{\eta\alpha\phi} r_{ik}^\phi + \epsilon_{\eta\theta\alpha} r_{jk}^\theta) (\vec{r}_{jk} \times \vec{r}_{ik})^\eta}{|\vec{r}_{jk} \times \vec{r}_{ik}|} \quad (4.14)$$

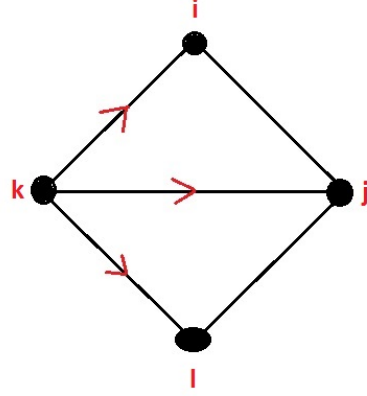
Putting Eqn.4.10 & Eqn.4.14 back into Eqn.4.6.

$$\frac{\partial a_\beta}{\partial r_{k\alpha}} = - \frac{(\epsilon_{\beta\alpha\nu} r_{ik}^\nu + \epsilon_{\beta\mu\alpha} r_{jk}^\mu)}{|\vec{r}_{jk} \times \vec{r}_{ik}|} + \frac{(\epsilon_{\eta\alpha\phi} r_{ik}^\phi + \epsilon_{\eta\theta\alpha} r_{jk}^\theta) (\vec{r}_{jk} \times \vec{r}_{ik})^\eta (\vec{r}_{jk} \times \vec{r}_{ik})_\beta}{|\vec{r}_{jk} \times \vec{r}_{ik}|^3} \quad (4.15)$$

First term in Eqn.4.5 can now be written down as follows:

$$(\nabla \vec{a}) \cdot \vec{b} = - \frac{(\epsilon_{\beta\alpha\nu} r_{ik}^\nu + \epsilon_{\beta\mu\alpha} r_{jk}^\mu) (\vec{r}_{jk} \times \vec{r}_{lk})^\beta}{|\vec{r}_{jk} \times \vec{r}_{ik}| |\vec{r}_{jk} \times \vec{r}_{lk}|} + \frac{(\epsilon_{\eta\alpha\phi} r_{ik}^\phi + \epsilon_{\eta\theta\alpha} r_{jk}^\theta) (\vec{r}_{jk} \times \vec{r}_{ik})^\eta (\vec{r}_{jk} \times \vec{r}_{ik})_\beta (\vec{r}_{jk} \times \vec{r}_{lk})^\beta}{|\vec{r}_{jk} \times \vec{r}_{ik}|^3 |\vec{r}_{jk} \times \vec{r}_{lk}|} \quad (4.16)$$

Similarly, one obtains the equation for  $\vec{a} \cdot (\nabla \vec{b})$  by interchanging **node index from  $\mathbf{i} \leftrightarrow \mathbf{l}$** . Using Eqn.4.16, we can now write down the complete expression for Eqn.4.3 which we do next.



**Figure 4.2:** Neighboring triangular plaquettes with indexed lattices

Completing the equation one gets acceleration at lattice site  $k$  as

$$\begin{aligned}
 (\vec{F}_k)_\alpha = \tilde{k} & \left( \frac{(\epsilon_{\beta\alpha\nu} r_{ik}^\nu + \epsilon_{\beta\mu\alpha} r_{jk}^\mu)(r_{jk} \times r_{lk})^\beta}{|r_{jk} \times r_{ik}| |r_{jk} \times r_{lk}|} + \frac{(\epsilon_{\beta\alpha\nu} r_{lk}^\nu + \epsilon_{\beta\mu\alpha} r_{jk}^\mu)(r_{jk} \times r_{ik})^\beta}{|r_{jk} \times r_{ik}| |r_{jk} \times r_{lk}|} \right. \\
 & - \frac{(r_{jk} \times r_{ik})_\beta (r_{jk} \times r_{lk})^\beta (\epsilon_{\eta\alpha\phi} r_{ik}^\phi + \epsilon_{\eta\theta\alpha} r_{jk}^\theta)(r_{jk} \times r_{ik})^\eta}{|r_{jk} \times r_{ik}|^3 |r_{jk} \times r_{lk}|} \\
 & \left. - \frac{(r_{jk} \times r_{lk})_\beta (r_{jk} \times r_{ik})^\beta (\epsilon_{\eta\alpha\phi} r_{lk}^\phi + \epsilon_{\eta\theta\alpha} r_{jk}^\theta)(r_{jk} \times r_{lk})^\eta}{|r_{jk} \times r_{lk}|^3 |r_{jk} \times r_{ik}|} \right) \quad (4.17)
 \end{aligned}$$

Eqn.4.17 is expression for the first bending contribution to force when node  $k$  in Fig.4.1 is wiggled keeping other nodes fixed.

## Bending Force II

The second contribution comes from wiggling lattice site  $l$  and holding other lattice sites  $i, j$  and  $k$  stationary. **Note that this time the wiggled lattice site  $l$  doesn't lie on the common edge of the two triangles.**

$$\vec{F}_l = -\nabla_l \sum_{IJ} \tilde{k}(1 - \cos \Theta_{IJ}) = -\tilde{k} \nabla_l \sum_{IJ} \frac{(\vec{r}_{jk} \times \vec{r}_{ik}) \cdot (\vec{r}_{jk} \times \vec{r}_{lk})}{|\vec{r}_{jk} \times \vec{r}_{ik}| |\vec{r}_{jk} \times \vec{r}_{lk}|} \quad (4.18)$$

Using the identity Eqn.4.5 we get

$$\vec{F}_l = -\tilde{k}((\nabla_l \vec{a}) \cdot \vec{b} + \vec{a} \cdot (\nabla_l \vec{b})) \quad (4.19)$$

**Define:**

- We identify  $a_\beta \equiv \frac{(\vec{r}_{jk} \times \vec{r}_{ik})_\beta}{|\vec{r}_{jk} \times \vec{r}_{ik}|}$  and  $b_\beta \equiv \frac{(\vec{r}_{jk} \times \vec{r}_{lk})_\beta}{|\vec{r}_{jk} \times \vec{r}_{lk}|}$ .
- $\nabla_l \vec{a} \rightarrow \frac{\partial a_\beta}{\partial X_\alpha}$  where  $X_\alpha \equiv r_{l\alpha}$
- we are wiggling lattice point  $l$  keeping other lattice sites fixed.

Expanding  $\frac{\partial a_\beta}{\partial r_{l\alpha}}$  we get Eqn.4.20

$$\frac{\partial a_\beta}{\partial r_{l\alpha}} = \frac{|\vec{r}_{jk} \times \vec{r}_{ik}| \frac{\partial (\vec{r}_{jk} \times \vec{r}_{ik})_\beta}{\partial r_{l\alpha}} - (\vec{r}_{jk} \times \vec{r}_{ik})_\beta \frac{\partial |\vec{r}_{jk} \times \vec{r}_{ik}|}{\partial r_{l\alpha}}}{|\vec{r}_{jk} \times \vec{r}_{ik}|^2} \quad (4.20)$$

Again we will look at the terms in the numerator of  $(\nabla_l \vec{a}) \cdot \vec{b}$ .

$$\frac{\partial (\vec{r}_{jk} \times \vec{r}_{ik})_\beta}{\partial r_{l\alpha}} = \frac{\partial \epsilon_{\beta\mu\nu} (r_{jk})^\mu (r_{ik})^\nu}{\partial r_{l\alpha}} = \epsilon_{\beta\mu\nu} \frac{\partial (r_{jk})^\mu}{\partial r_{l\alpha}} (r_{ik})^\nu + \epsilon_{\beta\mu\nu} (r_{jk})^\mu \frac{\partial (r_{ik})^\nu}{\partial r_{l\alpha}} = 0 \quad (4.21)$$

$$\frac{\partial |\vec{r}_{jk} \times \vec{r}_{ik}|}{\partial r_{l\alpha}} = \frac{\partial \sqrt{(\vec{r}_{jk} \times \vec{r}_{ik})_\eta \cdot (\vec{r}_{jk} \times \vec{r}_{ik})^\eta}}{\partial r_{l\alpha}} = 0 \quad (4.22)$$

Eqn.4.22 is simplified by using Eqn.4.21. So,  $(\nabla_l \vec{a}) \cdot \vec{b} = 0$ . Next, we will look at the term  $\vec{a} \cdot (\nabla_l \vec{b})$  where  $b_\beta = \frac{(\vec{r}_{jk} \times \vec{r}_{lk})_\beta}{|\vec{r}_{jk} \times \vec{r}_{lk}|}$ .

$$\begin{aligned} \frac{\partial(r_{kj}^{\vec{}} \times r_{lk}^{\vec{}})_\beta}{\partial r_{l\alpha}} &= \frac{\partial \epsilon_{\beta\mu\nu} (r_{jk}^{\vec{}})^\mu (r_{lk}^{\vec{}})^\nu}{\partial r_{l\alpha}} = \epsilon_{\beta\mu\nu} \frac{\partial (r_{kj}^{\vec{}})^\mu}{\partial r_{l\alpha}} (r_{lk}^{\vec{}})^\nu + \epsilon_{\beta\mu\nu} (r_{jk}^{\vec{}})^\mu \frac{\partial (r_{lk}^{\vec{}})^\nu}{\partial r_{l\alpha}} \\ &= \epsilon_{\beta\mu\nu} (r_{jk}^{\vec{}})^\mu \delta_\alpha^\nu = \epsilon_{\beta\mu\alpha} (r_{jk}^{\vec{}})^\mu \end{aligned} \quad (4.23)$$

$$\frac{\partial |r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}|}{\partial r_{l\alpha}} = \frac{\partial (r_{jk}^{\vec{}} \times r_{lk}^{\vec{}})_\eta}{\partial r_{l\alpha}} \frac{(r_{jk}^{\vec{}} \times r_{lk}^{\vec{}})_\eta}{|r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}|} = \frac{\epsilon_{\eta\mu\alpha} (r_{jk}^{\vec{}})^\mu (r_{jk}^{\vec{}} \times r_{lk}^{\vec{}})_\eta}{|r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}|} \quad (4.24)$$

Solving for  $\frac{\partial b_\beta}{\partial r_{l\alpha}}$  using Eqn.4.23 and Eqn.4.24.

$$\frac{\partial b_\beta}{\partial r_{l\alpha}} = \frac{\epsilon_{\beta\mu\alpha} (r_{jk}^{\vec{}})^\mu}{|r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}|} - \frac{(r_{jk}^{\vec{}} \times r_{lk}^{\vec{}})_\beta \epsilon_{\eta\mu\alpha} (r_{jk}^{\vec{}})^\mu (r_{jk}^{\vec{}} \times r_{lk}^{\vec{}})_\eta}{|r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}|^3} \quad (4.25)$$

Solve for  $\vec{a} \cdot (\nabla_l \vec{b})$ . This leads to writing the acceleration  $(\vec{F}_l)_\alpha$  from this contribution.

$$(\vec{F}_l)_\alpha = -\tilde{k} \vec{a} \cdot (\nabla_l \vec{b}) = -\frac{\tilde{k} \epsilon_{\alpha\beta\gamma} (r_{jk}^{\vec{}} \times r_{ik}^{\vec{}})^\beta (r_{jk}^{\vec{}})^\gamma}{|r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}| |r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}|} + \frac{\tilde{k} (r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}) \cdot (r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}) \epsilon_{\alpha\beta\gamma} (r_{jk}^{\vec{}} \times r_{lk}^{\vec{}})^\beta (r_{jk}^{\vec{}})^\gamma}{|r_{jk}^{\vec{}} \times r_{lk}^{\vec{}}|^3 |r_{jk}^{\vec{}} \times r_{ik}^{\vec{}}|} \quad (4.26)$$

### Total bending force

The calculations in this Secs.4.2.1 gives the total force each node experiences because of bending. Sum of Eqn.4.17 and Eqn.4.26 represents this net force on the lattice node. The equations are complicated and considerable care has to be taken when implementing them in a computer simulation.



### 4.2.2 Derivation of bond harmonic force

The bond-harmonic energy between two nodes is modeled using a ball spring model. The bond harmonic energy itself is given by the

$$E_{\text{stretch}} = \frac{1}{2}\epsilon(r_{ij} - a)^2 \quad (4.27)$$

where  $i, j$  runs overall bonds and  $a$  is the rest bond length (no stretching).

$$\begin{aligned} E_{\text{stretch}} &= \frac{1}{2}\epsilon(\sqrt{(X_{i,\alpha} - X_{j,\alpha})^2} - a)^2 \\ \vec{a}_{\text{stretch}} &= -\frac{1}{m}\nabla E_{\text{stretch}} = \frac{-\epsilon}{2m} \frac{\partial(\sqrt{(X_{i,\alpha} - X_{j,\alpha})^2} - a)^2}{\partial X_{i,\alpha}} \\ a_{\text{stretch},\alpha} &= \frac{-\epsilon(r_{ij} - a)(X_{i,\alpha} - X_{j,\alpha})}{mr_{ij}} \end{aligned} \quad (4.28)$$

The bond harmonic force at each node is given by the Eqn.4.28.

### 4.2.3 Noose Hoover Algorithms

Noose Hoover [62] heat bath provides an efficient method for constant-temperature molecular dynamics simulations. This algorithm entails solving the Velocity Verlet equations using Newton Raphson method [63]. This approach of evolving the NVT system is *irreversible*.

$$\begin{aligned} v_i(t + \Delta t) &= v_i(t + \frac{\Delta t}{2}) + (\frac{f_i(t + \Delta t)}{m_i} - \zeta(t + \Delta t)v_i(t + \Delta t))\frac{\Delta t}{2} \\ \zeta(t + \Delta t) &= \zeta(t + \frac{\Delta t}{2}) + (\sum_i m_i v_i^2(t + \Delta t) - gT)\frac{\Delta t}{2Q} \end{aligned} \quad (4.29)$$

The index  $i$  runs over all the lattice sites.  $\zeta$  is the **thermodynamic friction coefficient**, which characterizes the Noose Hoover heat bath. In both, the equations above  $v_i(t + \Delta t)$  and  $\zeta(t + \Delta t)$  appear on the right and left-hand sides, therefore these equations

cannot be integrated exactly. The Newton-Raphson scheme is used to solve these equations numerically.

## 4.3 CUDA C GPU implementation

In the previous Sec.4.2 we derived the forces that act at each lattice point. As one can see from the expressions for computation of forces Eqn.4.17 and Eqn.4.26 are complicated and as such the computation cost for each term in CPU time will be considerable. This means computation time will increase with the number of nodes and by extension the system size. The equations of motion in the velocity verlet algorithm 4.2.3 and the force equations (Eqn.4.17 and Eqn.4.26) depend on the node positions at time  $t_i$  for evaluating positions at time  $t_{i+1}$ . In a conventional sequential CPU code, one would need to visit each node, evolve using the EOM and in the end, when all nodes have been visited the system evolves to its next configuration. Computation cost will be considered in this method for large systems with many nodes.

Modern NVIDIA GPUs provide an alternative to increasing speed of code execution. GPUs are suited for computations which support parallel computations. In our case, the time evolution of nodes is independent of each other in the sense that EOMs for the nodes are independent of each other. Hence, our system of equations 4.2.3 supports parallelization.

CUDA C [64] is a programming language which enables GPU coding and exploits GPU computing power for increasing computation speed. The general framework of the coding language treats variables for the CPU and GPU separately. GPU variables cannot be accessed by CPU and vice versa. The framework allows for copying of data between GPU and CPU variables and this allows for data transfer between the two as and when required. Programming functions in CUDA C are called **kernels** [64].

### 4.3.1 Kernels

A typical C function allows performing a set of instructions only once when they are called. By definition they allow a **single thread** execution. On the other hand, kernels are executed

$N$  times in parallel by  $N$  **different CUDA threads**. We exploit this feature while coding MD for thermalized elastic sheets with  $N$  nodes.

A CUDA thread is associated with each node which executes its time evolution. All the  $N$  threads work in parallel and update the position of  $N$  nodes in the end. The GPU variable holding the position data for nodes has to be chosen carefully. GPU provides two types of memory shared and global. Shared memory is limited in size but access is fast as opposed to the global memory which can accommodate a very large array but data fetching is slow. The tradeoff is similar to the one experienced in RAM swaps where stack variables can be accessed quickly as opposed to heap variables. Heap allows to store very large arrays but call stack size is limited to 8Mb in a Linux system.

The time evolution proceeds much faster as the evolution of all nodes occurs concurrently. Each node maintains its own position, velocity, and acceleration variables. One needs to take care to wait for an adequate amount of time before moving to the next step, as all threads should complete processing before moving on. Also, GPU simulation does not increase speed for all system sizes, in fact for smaller system sizes, it is slower compared to CPU. A major factor in speed throughput is the system size and how many CUDA threads are launched. If the number of threads fills all Algorithm Units in a GPU chip, the simulation will run most efficiently. Therefore, while launching a GPU code significant effort should be given to work with a system size which provides adequate number of parallel thread launches [65].

## 4.4 HOOMD Blue package

HOOMD-blue [66] is a general purpose Molecular Dynamics toolkit. It can be used to run MD simulations in both CPU and GPUs. The initial starting conditions and boundary conditions can be supplied using a high-level python script. The intricate details of force at each node, heat bath and evolution itself is handled completely by the HOOMD software. Major advantages of using HOOMD-Blue package are:

- Fast prototyping, to quickly check a new system or existing system with new boundary condition.

- Scalable can be used in CPU as well as GPU clusters.
- Easy to initialize and supply conditions using a python script.
- Installation is straightforward using conda [67].
- Documentation has illustrated examples to apply boundary conditions.
- A well maintained Hoomd User Google group where questions are answered by developers of HOOMD-blue.

## 4.5 Visual Molecular Dynamics package

VMD [61] is a visualization program for displaying, animating, and analyzing raw data file generated from various other simulation software. VMD can be used in Linux, Mac OS and Windows systems. Installation is straightforward for Linux systems but may require some modifications for Mac OS which is case specific. In our case, we use VMD to load and visualize the trajectory files (with .gsd extension) which we get from our HOOMD-blue simulation. VMD does not support GSD file systems out of the box since, GSD files are a fairly new addition to HOOMD-blue. But thankfully, a GSD plugin for VMD has been written which one has to download and install separately [68].

## 4.6 Monte Carlo Implementation

The Monte Carlo method is used widely not only in Physics but also in Chemistry, Biology and Finance. It is a sampling technique which is used to calculate thermodynamic variables of the form

$$\langle O \rangle = \frac{\sum_{\nu} O e^{-\beta T}}{\sum_{\nu} e^{-\beta T}}, \quad (4.30)$$

where  $\nu$  runs over all possible configurations that the system can take at a given temperature  $T$ . For example, an Ising model consisting of  $N$  lattice sites will have  $2^N$  possible configurations. In our case of membranes, each vertex can move arbitrarily in the embedded 3D euclidean space and hence the number of available configurations is infinite. Even using a very powerful computer it will not be possible to run through all the configurations that are available to the system. So to solve this problem one uses Monte Carlo method to approximate  $\langle O \rangle$  by using random samples of the configuration weighted by the Boltzmann factor.

### 4.6.1 Metropolis method

In this method successive configurations available to the system are treated as states in a Markov process. So, state at time  $t+1$  is constructed from the previous state at time  $t$  via a transition probability  $W(x_{(t)} \rightarrow x_{(t+1)})$ . If we look at two states at time  $t$  and  $t'$  then the principle of detail balance has to be fulfilled [69].

$$P_{eq}(x_t)W(x_t \rightarrow x_{t'}) = P_{eq}(x_{t'})W(x_{t'} \rightarrow x_t), \quad (4.31)$$

where  $P_{eq}(x_t)$  is the probability equilibrium distribution.

In our membrane simulations we perform Monte Carlo simulations as described below:

- Choose any random vertex and move it by a small amount  $\delta$ .
- Compute the change in energy of the membrane before and after the movement. Evaluate the energy difference  $\Delta H_\delta$ .
- Accept this movement with a probability

$$p = \min(1, \exp(-\beta \Delta H_\delta)) \quad (4.32)$$

Eqn.4.32 satisfies detailed balance Eqn.4.31.

# Chapter 5

## Simulation Data Analysis

### 5.1 Introduction

In this chapter, we discuss the various analysis techniques used to analyze the data. First and foremost thing in any simulation is to arrive at the equilibration point beyond which any meaningful recording of observables can be made. We will look at how to check for equilibration and how to estimate the number of steps required to get enough data for analysis. Error estimation is an integral part of data analysis and we will go in depth to understand Jack knife and Bootstrap method of obtaining the errors.

### 5.2 System Equilibration time

When performing monte carlo or molecular dynamics simulations one has to check for thermalization of the system. One can ask questions like, how long should one run a simulation? At what interval should one record simulation data so that they are uncorrelated? How much data be discarded when evaluating average values? At which point has the system reached thermalization? To answer these questions we will look at the time autocorrelation for a given system observable [52]. The equilibrium normalized autocorrelation function is defined as below:

$$C_{OO}(t) \equiv \langle O_s O_{s+t} \rangle_\beta - \langle O \rangle_\beta^2, \quad (\text{unnormalized}), \quad (5.1)$$

$$\rho_{OO}(t) \equiv \frac{C_O(t)}{C_O(0)}, \quad (\text{normalized}), \quad (5.2)$$

For long enough simulation run times,  $\rho_{OO}(t)$  will decay exponentially. From the autocorrelation function, one can extract two characteristic times which have different significance:

### 5.2.1 Integrated autocorrelation time

The integrated autocorrelation time is given by the area under the autocorrelation function.

$$\tau_{\text{int},O} = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_{OO}(t), \quad (5.3)$$

The integrated autocorrelation time measures the effective number of independent measurements for an observable  $O$  for which we have recorded  $N$  data points. Which means we will have  $N/\tau_{\text{int},O}$  independent measurements. Put differently,  $\tau_{\text{int},O}$  is the time between two independent measurements.

### 5.2.2 Exponential time

The exponential time is obtained by fitting the autocorrelation function to an exponential decay function. The exponential autocorrelation time is the time one has to wait before the system thermalizes.

$$\tau_{\text{exp},O} = \limsup_{t \rightarrow \infty} \frac{1}{-\log|\rho_{OO}(t)|} \quad (5.4)$$



The exponential time gives the point at which the system thermalizes. If  $n_{dis}$  represents the number of data points we discard to account for thermalization,  $n_{dis} = 20\tau_{exp,O}$  will be more than enough to make a correct discard. In our simulation, we have discarded the first half of the data recorded which is well beyond the thermalization point of the system.

## 5.3 Error Estimation

We start by explaining the two methods of obtaining errors in observable data in our simulations. We record data at fixed intervals in our simulation and we end up with a time series of system observable. Even though one can ascertain the interval at which the data is independent for a given set parameter by evaluating the time autocorrelation function as explained in Sec. 5.2.1, its cumbersome to do so for each and every parameter set. So, we fix the time interval of data recording (10000 steps) in these simulations, and use error estimation techniques namely Jack Knife and Bootstrap to get rid of any correlation or bias in the data.

### 5.3.1 Jack Knife

Jack Knife error estimation works by blocking the observable data [52]. Say, we have  $R$  observable data ( $\langle O^{(1)} \rangle_\beta, \langle O^{(2)} \rangle_\beta, \dots, \langle O^{(R)} \rangle_\beta$ ) and we want to report the average and error in  $f(\langle O^{(1)} \rangle_\beta, \langle O^{(2)} \rangle_\beta, \dots, \langle O^{(R)} \rangle_\beta)$ .

- Estimate  $f(\langle O^{(1)} \rangle_\beta, \langle O^{(2)} \rangle_\beta, \dots, \langle O^{(R)} \rangle_\beta)$  using the standard formula to evaluate the average, which is sum total of all recordings of  $f$  divided by the number of total recordings.
- Next, for each of the observable data above ( $\langle O^{(1)} \rangle_\beta, \langle O^{(2)} \rangle_\beta, \dots, \langle O^{(R)} \rangle_\beta$ ) form block data, using large enough block size  $b$ . Go from set of  $N$  data, to a set of  $N/b$  data blocks each of size  $b$ :

$$O_{b,i} = \frac{1}{b} \sum_{t=(i-1)b+1}^{bi} O_t, \quad (5.5)$$

- Then use the blocked data to form Jack-Knife blocks as follows:

$$O_{JK,b,i}^{(r)} = \frac{1}{\frac{N}{b} - 1} \sum_{j \neq i} O_{b,j}^{(r)}, \quad r = 1, 2, \dots, R. \quad (5.6)$$

Use the Jack Knife blocks for individual observable data to generate Jack Knife blocks for  $f$ :

$$f_{JK,b,i} = f(O_{JK,b,i}^{(1)}, O_{JK,b,i}^{(2)}, \dots, O_{JK,b,i}^{(R)}) \quad (5.7)$$

- The (square) error in  $f$  goes as below:

$$\overline{\Delta_f^2} = \left(\frac{N}{b} - 1\right) \left[ \frac{b}{N} \sum_{i=1}^{N/b} f_{JK,b,i}^2 - \left(\frac{b}{N} \sum_{i=1}^{N/b} f_{JK,b,i}\right)^2 \right], \quad r = 1, 2, \dots, R. \quad (5.8)$$

The Jack knife blocking takes into account any correlations within the data, since all jack knife blocks are allowed to fluctuate against each other. In our simulations, we have multiple runs for the same parameter set (10 runs) in most cases, so we do not need to make separate blocks, each run data acts as a block data set and can be used to form the Jack knife blocks.

### 5.3.2 Bootstrap

In bootstrap, we resample the  $N$  data points. We go from one data set of  $N$  points to  $N_{boot}$  data sets each containing  $N$  points. We achieve this by performing random (Monte Carlo)

sampling of the original set of  $N$  points. Repetition of data points within a new sampled data set is allowed [70].

- Imagine we start with  $N$  data points for an observable  $x$ . We perform random sampling and obtain  $N_{boot}$  datasets as mentioned above. For each bootstrap data set the average will be given by

$$x_{\alpha}^B = \frac{1}{N} \sum_i^N n_{i,\alpha} x_i, \quad (5.9)$$

where  $n_{i,\alpha}$  is the number of times  $x_i$  is repeated in a bootstrap sample.

- We next compute the bootstrap average of the mean of  $x$  and the bootstrap variance in the mean by averaging over all the bootstrap data sets.

$$\overline{x^B} = \frac{1}{N_{boot}} \sum_{\alpha=1}^{N_{boot}} x_{\alpha}^B, \quad (5.10)$$

$$\overline{(x^B)^2} = \frac{1}{N_{boot}} \sum_{\alpha=1}^{N_{boot}} [(x_{\alpha}^B)^2], \quad (5.11)$$

- The bootstrap error estimate is then given by

$$\sigma_{\overline{x}} = \sqrt{\frac{N}{N-1} (\overline{(x^B)^2} - (\overline{x^B})^2)}, \quad (5.12)$$

We have used bootstrap method while estimating errors in the Cumulative Distribution Function in the analysis of thermalized Euler buckling.

# Bibliography

- [1] L. D. Landau and E. M. Lifshitz, *Theory of Elasticity* (Butterworth-Heinemann,, Singapore, 1999), 3rd ed.
- [2] D. Nelson and L. Peliti, *J. Phys. France* **48**, 1085 (1987).
- [3] J. A. Aronovitz and T. C. Lubensky, *Phys. Rev. Lett.* **60**, 2634 (1988).
- [4] E. Guitter, F. David, S. Leibler, and L. Peliti, *Journal de Physique* **50(14)**, 1787 (1989).
- [5] P. Le Doussal and L. Radzihovsky, *Phys. Rev. Lett.* **69**, 1209 (1992), URL <http://link.aps.org/doi/10.1103/PhysRevLett.69.1209>.
- [6] Z. Zhang, H. T. Davis, and D. M. Kroll, *Phys. Rev. E* **48**, R651 (1993), URL <https://link.aps.org/doi/10.1103/PhysRevE.48.R651>.
- [7] M. J. Bowick, S. M. Catterall, M. Falcioni, G. Thorleifsson, and K. N. Anagnostopoulos, *J. Phys. I France* **6**, 1321 (1996).
- [8] M. J. Bowick and A. Travesset, *Phys. Rep.* **344**, 255 (2001), URL <https://arxiv.org/abs/cond-mat/0002038>.
- [9] A. Fasolino, J. H. Los, and M. I. Katsnelson, *Nat. Mat.* **6**, 858 (2007).
- [10] M. J. Bowick and L. Giomi, *Adv. Phys.* **58**, 449 (2009).
- [11] D. Gazit, *Phys. Rev. E* **80**, 041117 (2009).

- [12] J. H. Los, M. I. Katsnelson, O. V. Yazyev, K. V. Zakharchenko, and A. Fasolino, Phys. Rev. B **80**, 121405 (2009).
- [13] K. V. Zakharchenko, R. Roldán, A. Fasolino, and M. I. Katsnelson, Phys. Rev. B **82**, 125435 (2010).
- [14] F. L. Braghin and N. Hasselmann, Phys. Rev. B **82**, 035407 (2010).
- [15] N. Hasselmann and F. L. Braghin, Phys. Rev. E **83**, 031137 (2011).
- [16] A. Tröster, Phys. Rev. B **87**, 104112 (2013), URL <https://link.aps.org/doi/10.1103/PhysRevB.87.104112>.
- [17] A. Tröster, Phys. Rev. E **91**, 022132 (2015), URL <https://link.aps.org/doi/10.1103/PhysRevE.91.022132>.
- [18] J. H. Los, A. Fasolino, and M. I. Katsnelson, Phys. Rev. Lett. **116**, 015901 (2016), URL <https://link.aps.org/doi/10.1103/PhysRevLett.116.015901>.
- [19] A. Košmrlj and D. R. Nelson, Phys. Rev. B **93**, 125431 (2016).
- [20] P. D. de Gennes, *Scaling concepts in Polymer Physics* (Cornell University Press, Cornell, Ithaca, 1979).
- [21] Y. Oono, Adv. Chem. Phys. **61**, 301 (1985).
- [22] P. D. de Gennes, *The Physics of Liquid Crystals* (Clarendon, Oxford, 1974).
- [23] Y. Kantor and D. R. Nelson, Phys. Rev. Lett. **58**, 2774 (1987), URL <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.58.2774>.
- [24] T. J. Byers and D. Branton, PNAS **82(18)**, 6153 (1985), URL <https://doi.org/10.1073/pnas.82.18.6153>.
- [25] C. F. Schmidt, K. Svoboda, N. Lei, I. Petsche, L. Berman, C. Safinya, and G. Grest, Science **259**, 952 (1993), URL <https://d2ufo471rtsv5s.cloudfront.net/content/259/5097/952>.

- [26] P. G. De Gennes, *The Journal of Physical Chemistry* **86** (13), 2294 (1982).
- [27] D. R. Nelson, *Defects and Geometry in Condensed Matter Physics* (Cambridge University Press, Cambridge, UK, 2002).
- [28] L. Peliti and S. Leibler, *Phys. Rev. Lett.* **54**, 1690 (1985), URL <https://doi.org/10.1103/PhysRevLett.54.1690>.
- [29] Y. Kantor and D. R. Nelson, *Phys. Rev. A* **36**, 4020 (1987), URL <https://link.aps.org/doi/10.1103/PhysRevA.36.4020>.
- [30] M. J. Aziz, E. Nygren, J. F. Hays, and D. Turnbull, *J. Appl. Phys* **57**, 2233 (1985), URL <https://aip.scitation.org/doi/abs/10.1063/1.334368>.
- [31] M. I. Katsnelson, *Graphene: Carbon in Two Dimensions* (Cambridge University Press, New York, 2012).
- [32] D. Nelson, T. Piran, and S. Weinberg, *Statistical Mechanics of Membranes and Surfaces* (World Scientific, Singapore, 2004), 2nd ed.
- [33] Y. Kantor, M. Kardar, and D. R. Nelson, *Phys. Rev. A* **35**, 3056 (1987), URL <https://link.aps.org/doi/10.1103/PhysRevA.35.3056>.
- [34] M. K. Blees, A. W. Barnard, P. A. Rose, S. P. Roberts, K. L. McGill, P. Y. Huang, A. R. Ruyack, J. W. Kevek, B. Kobrin, D. A. Muller, et al., *Nature* **524**, 204 (2015).
- [35] E. Russell, R. Sknepnek, and M. J. Bowick (2015), [arXiv:1512.04670](https://arxiv.org/abs/1512.04670).
- [36] C. Chena, D. H. Zanette, D. A. C. S. Shaw, and D. Lopez, *Nature Communications* **8** (2017).
- [37] V. B. Braginsky, V. Mitrofanov, V. I. Panov, and C. Eller, University of Chicago Press (1985).
- [38] M. Dykman, OUP Oxford (2012).

- [39] D. Yllanes, S. S. Bhabesh, D. R. Nelson, and M. J. Bowick, Nature Communications **8** (2018), URL <https://www.nature.com/articles/s41467-017-01551-y>.
- [40] H. S. Seung and D. R. Nelson, Phys. Rev. A **38**, 1005 (1988).
- [41] R. Cuerno, R. Gallardo Caballero, A. Gordillo-Guerrero, P. Monroy, and J. J. Ruiz-Lorenzo, Phys. Rev. E **93**, 022111 (2016).
- [42] M. J. Bowick, A. Kosmrlj, D. R. Nelson, and R. Sknepnek, Phys. Rev. B **95**, 104109 (2017), [arXiv:1608.04197](https://arxiv.org/abs/1608.04197).
- [43] C. Lee, X. Wei, J. W. Kysar, and J. Hone, Science **321**, 385 (2008), ISSN 0036-8075.
- [44] H. Zhao, K. Min, and N. R. Aluru, Nano Letters **9**, 3012 (2009), PMID: 19719113.
- [45] R. G. Harnish and J. F. Wheeler, Nuclear Physics B **350**, 861 (1991), URL <http://www.sciencedirect.com/science/article/pii/055032139190166U>.
- [46] E. Guitter, F. David, S. Leibler, and L. Peliti, Phys. Rev. Lett. **61**, 2949 (1988), URL <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.61.2949>.
- [47] R. L. Renken and J. B. Kogut, Nuclear Physics B **342**, 753 (1990), ISSN 0550-3213, URL <http://www.sciencedirect.com/science/article/pii/055032139090336C>.
- [48] J. Wheeler, Nuclear Physics B **458**, 671 (1996), ISSN 0550-3213, URL <http://www.sciencedirect.com/science/article/pii/0550321395005447>.
- [49] D. Espriu and A. Travasset, Nuclear Physics B **468**, 514 (1996), ISSN 0550-3213.
- [50] H. Koibuchi, N. Kusano, A. Nidaira, K. Suzuki, and M. Yamada, Phys. Rev. E **69**, 066139 (2004), URL <https://link.aps.org/doi/10.1103/PhysRevE.69.066139>.
- [51] J.-P. Kownacki and D. Mouhanna, Phys. Rev. E **79**, 040101 (2009).
- [52] D. J. Amit and V. Martin-Mayor, *Field Theory, the Renormalization Group, and Critical Phenomena* (World Scientific, Singapore, 2005), 3rd ed.

- [53] P.-G. de Gennes, *Scaling Concepts in Polymer Physics* (Cornell University Press, Ithaca, 1979).
- [54] S. Nosé, *The Journal of Chemical Physics* **81**, 511 (1984).
- [55] W. G. Hoover, *Phys. Rev. A* **31**, 1695 (1985), URL <https://link.aps.org/doi/10.1103/PhysRevA.31.1695>.
- [56] J. Glaser, T. D. Nguyen, J. A. Anderson, P. Lui, F. Spiga, J. A. Millan, D. C. Morse, and S. C. Glotzer, *Comp. Phys. Comm.* **192**, 97 (2015).
- [57] J. A. Anderson, C. D. Lorenz, and A. Travesset, *J. Comp. Phys.* **227**, 5342 (2008).
- [58] K. J. Wiese, in *Phase Transitions and Critical Phenomena*, edited by C. Domb and J. Lebowitz (Academic Press, London, 2001), vol. 19, pp. 253–480.
- [59] M. Kardar and D. R. Nelson, *Phys. Rev. A* **38**, 966 (1988), URL <https://link.aps.org/doi/10.1103/PhysRevA.38.966>.
- [60] S. Dou, B. S. Strachana, S. W. Shaw, and J. S. Jensen, *Philosophical Transactions of the Royal Society A* (2015).
- [61] *VMD*, <http://www.ks.uiuc.edu/Research/vmd/>.
- [62] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Academic Press, 2001), 2nd ed.
- [63] H. William, A. Saul, T. William, and P. Brian, *Numerical Recipes The Art of Scientific Computing* (Cambridge University Press, Hong Kong, 2007), 3rd ed.
- [64] *CUDA C*, <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>.
- [65] C. John, G. Max, and M. Ty, *CUDA C Programming* (John Wiley, 2014), 1st ed.
- [66] *HOOMD*, <https://glotzerlab.engin.umich.edu/hoomd-blue/>.



- [67] *HOOMD Installation*, <http://glotzerlab.engin.umich.edu/hoomd-blue/download.html>.
- [68] *HOOMD GSD plugin*, <https://bitbucket.org/azpanag/gsd-vmd>.
- [69] K. Binder and D. W. Heermann, *Monte Carlo Simulations in Statistical Physics* (Springer Series in Solid-State Science, 1992).
- [70] P. Young, arxiv.org (2012), URL <https://arxiv.org/abs/1210.3781>.

# VITA

**AUTHOR:** Sourav Bhabesh

**NATIONALITY:** Indian

**DATE OF BIRTH:** May 14, 1987

**DEGREES AWARDED:**

- MS, Physics, Syracuse University, New York, 2015
- B.Tech, Engineering Physics, Indian Institute of Technology, Madras, 2010

**PROFESSIONAL EMPLOYMENT:**

- Lead Business Analyst, J. P. Morgan, Mumbai, India (2010-2012).
- Associate Software Developer, Flextrade, Mumbai, India (2012-2013).
- Graduate Teaching/Research Assistant, Department of Physics, Syracuse University (2013-2018).

**PUBLICATIONS:**

- *Thermal crumpling of perforated two-dimensional sheets*, David Yllanes, Sourav. S. Bhabesh, David. R. Nelson, Mark. J. Bowick, Nature Communications, 8, 1381 (2017).

**AWARDS:**

- Travel grant from Soft Matter Program for APS March meeting (2017).
- Nominated for Syracuse University Outstanding Teaching Assistant Award (2016).
- Phi Beta Delta International Honor Society Award for academic excellence (2015).
- Graduate Assistantship, Department of Physics, Syracuse University (2016-2018).

**TEACHING:**

- PHY 663 : PhD Qualification Exam preparatory course (2017).
- PHY 424 : Electromagnetics(I) (2016).
- PHY 212 : General Physics II (2014, 2015).

**PRESENTATIONS:**

- Thermalized Euler Buckling in clamped ribbons with variable aspect ratio, APS March Meeting, Los Angeles, 2018.
- Statistical Mechanics of Square Frame, APS March Meeting, New Orleans, Louisiana, 2017.