Department of Physics and Astronomy

Ruprecht-Karls-University of Heidelberg

Master Thesis

in Physics

submitted by

Johannes Dreckhoff

born in Duesseldorf

2025

Curvature Generation in Dynamic Clathrin Coats: An Agent-Based Kinetic Monte Carlo Model

This master thesis by Johannes Dreckhoff

carried out at the

Institute for theoretical physics, BioQuant

under the supervision of

Prof. Ulrich Schwarz

and

Prof. Tristan Bereau

Abstract

Clathrin-mediated endocytosis is a fundamental transport pathway across cell membranes, yet the physical mechanisms by which the clathrin coat assembles to drive curvature generation are not understood well. This thesis investigates the structural assembly dynamics of clathrin by introducing an agent-based model employing Kinetic Monte Carlo simulations in spherical geometry. By formulating a microscopic Hamiltonian governing individual clathrin legs, we elucidate the interplay between lattice stiffening, membrane resistance, and curvature generation.

We demonstrate that complex global mechanical properties emerge from a local Hamiltonian describing single triskelia. Our analysis reveals that the effective bending rigidity of the coat increases by approximately two orders of magnitude during assembly, driving the curvature generation, together with the growth curvature imprinting itself onto the lattice, resulting in a "curvature memory". Crucially, our simulations capture the distinct dynamical regimes observed in experiments: we successfully reproduce the flat-to-curved transition, while also predicting stalled, flat growth events, potentially driven by premature lattice stiffening. This unifying description demonstrates that clathrin-mediated endocytosis is best described by neither the Constant Curvature nor the Constant Area Model exclusively. Instead, our findings provide support for a hybrid framework, such as the Cooperative Curvature Model, where the interplay of coat maturation and membrane mechanics dictates the curvature pathway.

Zusammenfassung

Clathrin-mediated Endocytosis ist ein fundamentaler zellulärer Transportweg, doch der physikalische Mechanismus, durch den sich die Clathrin-Hülle assembliert und Membrankrümmung erzeugt, bleibt Gegenstand von Diskussionen. Diese Arbeit untersucht die strukturelle Assemblierdynamik von Clathrin mittels eines neuen agentenbasierten Modells auf der Grundlage kinetischer Monte-Carlo-Simulationen in sphärischer Geometrie. Durch die Formulierung eines mikroskopischen Hamiltonians, der die einzelnen Clathrin-Beine beschreibt, beleuchten wir das emergente Zusammenspiel zwischen Gitterversteifung, topologischen Defekten und Membranwiderstand.

Von besonderer Bedeutung ist, dass wir zeigen, dass komplexe globale mechanische Eigenschaften ausschließlich aus einem lokalen Hamiltonian hervorgehen, der einzelnes Clathrin beschreibt. Unsere Analyse offenbart, dass die effektive Biegesteifigkeit des Mantels während der Assemblierung um etwa zwei Größenordnungen ansteigt. Außerdem zeigt sich ein ‚Krümmungsgedächtnis' (curvature memory), indem sich die Wachstumsgeometrie in das Gitter mechanisch einprägt. Entscheidend ist dabei, dass unsere Simulationen die experimentell beobachteten unterschiedlichen dynamischen Regime erfassen: Wir reproduzieren erfolgreich den Übergang von der flachen zur gekrümmten Membrangeometrie (flat-to-curved) und finden gleichzeitig stagnierende, flache Wachstumsprozesse, die potenziell durch eine vorzeitige Gitterversteifung verursacht werden. Diese vereinheitlichende Beschreibung zeigt, dass clathrin-mediated endocytosis weder durch das Constant Curvature Model noch durch das Constant Area Model allein hinreichend beschrieben wird. Stattdessen stützen unsere Ergebnisse einen hybriden Ansatz, wie das Cooperative Curvature Model, bei dem das Zusammenspiel von Mantelwachstum und Membranmechanik den Krümmungsverlauf bestimmt.

Overview

Life at the cellular level is defined by the selective transport of matter across the plasma membrane. While small ions and molecules traverse via channels and pumps, and large pathogens are engulfed by phagocytosis, there exists a critical functional gap for the uptake of intermediate-sized particles.

The biological solution to this transport problem is Clathrin-Mediated Endocytosis (CME). In this process, the protein clathrin assembles into a hexagonal lattice on the cytosolic face of the plasma membrane. This lattice grows into a spherical coat, driving the localized membrane bending necessary to form a transport vesicle, which is eventually separated from the parent membrane via scission.

Despite its fundamental importance in eukaryotic life, the physical mechanism by which the clathrin coat forms and simultaneously bends the membrane remains a subject of intense biophysical debate. Historically, two opposing models have dominated the discourse:

- The Constant Curvature Model (CCM), which posits that the coat assembles directly with its final curvature, implying that the membrane must be bent from the very onset of polymerization.
- The Constant Area Model (CAM), which suggests that the coat grows as a flat hexagonal plaque before undergoing a global conformational change to initiate bending.

This master thesis addresses the structural assembly dynamics of clathrin by introducing a novel agentic model. By simulating the statistical mechanics of coat growth, we aim to elucidate the interplay between the geometric stiffening of the protein lattice, the energy stored in topological defects, and the resistance of the underlying membrane.

The Agentic Model and Simulation Methodology

To deconstruct the mechanics of CME, we developed a comprehensive agentic model of clathrin assembly. In this framework, clathrin units are modelled as hubs (nodes) confined to a spherical surface with a variable radius R. These hubs dynamically form and rupture bonds—up to three direct bonds and three next-nearest-neighbour (NNN) bonds—mimicking the tripod structure of the clathrin triskelion.

The simulation framework was built from the ground up, resulting in a robust and flexible codebase comprising several thousand lines of Python code. It leverages the `JAX` library for high-performance numerical computing. A key feature of this implementation is the use of automatic differentiation (AutoGrad), which allows us to rigorously calculate forces and probe the minimized energy landscape of the growing coat without relying on approximations of the gradients.

The dynamics of the system are governed by a Hamiltonian formulated microscopically at the level of individual clathrin legs. The core energy contributions stem from harmonic potentials acting on:

- **Leg stretching stiffness (k_L):** Maintaining the equilibrium leg length L_0.
- **Projection angle stiffness (k_ϕ):** Favouring the ideal hexagonal lattice geometry (typically $\phi_0 = 120°$), stabilizing the honeycomb structure of the coat.
- **Pucker angle stiffness (k_χ):** Defining the energy cost associated with the leg dipping angle relative to the tangential plane, which sets the local preferred curvature H_0.

To capture the multi-scale nature of the assembly, the state evolution is divided into two distinct regimes:

1. **Microstate Changes:** Node movements and curvature reconfigurations are modelled using Metropolis-Hastings Monte Carlo (MH-MC). This assumes that the clathrin coat adiabatically relaxes around an energetic minimum on a fast timescale compared to growth.
2. **Macrostate Changes:** Processes that drive the system's time evolution, such as bond formation, bond removal, and polymerization, are simulated using a Kinetic Monte Carlo (KMC) scheme.

To ensure computational tractability for systems with low polymerization rates, where physically relevant events are rare, a custom oversampling technique was introduced to accelerate the KMC acceptance of these critical transitions.

Key Results I: Emergent Rigidity and Curvature Memory

Analysis of coats grown at fixed curvature demonstrated that the assembly process is highly energy-driven. Under ideal conditions, the probabilistic assembly converges towards highly regular structures, specifically configurations resembling the truncated icosahedron (composed of 12 pentagons and 20 hexagons), which minimizes energy at a corresponding curvature of $H \approx 0.4\,\mathrm{L}_0^{-1}$.

A central finding of this work concerns the effective coat bending rigidity (κ_C), calculated from the differentiated zero-temperature energy. Our analysis reveals that the mechanical properties of the clathrin lattice are not constant; rather, the coat undergoes a drastic stiffening process as it assembles.

Initially, the nascent coat is soft and flexible, with an estimated rigidity of $\kappa_C^{\mathrm{init}} \sim 20\,\mathrm{k_B T}$. In this phase, the mechanics are dominated by the relatively soft angular constraints of the pucker angle (k_χ). However, as the lattice matures and the network connectivity increases, we observed a major increase in rigidity spanning approximately two orders of magnitude. The final bending rigidity reaches $\kappa_C^{\mathrm{fin}} \sim 1 \times 10^3\,\mathrm{k_B T}$. In this mature state, the coat is mechanically locked, and further deformation is resisted by the much higher stretching stiffness of the clathrin legs ($k_L \sim 800\,\mathrm{k_B T}$).

This stiffening process leads to a phenomenon we describe as curvature memory. The process can be likened to a growing vine that hardens into wood: initially, the structure

is pliable and can adapt to its environment, but as it matures, it "burns in" its shape. We quantified this memory effect by observing the energy-minimizing curvature ($H_{\min}$) of the coat. While initially close to the microscopic preference H_0, $H_{\min}$ shifts over time to match the curvature at which the coat was grown. This implies that the coat eventually loses memory of its original microscopic preference. This mechanism provides a compelling explanation for why large, flat clathrin patches observed in experiments often become stalled: they have essentially "hardened" in the wrong shape, making subsequent curvature generation energetically prohibitive.

Key Results II: Dynamic Assembly and Trajectory Classes

When simulating the full dynamic system, incorporating the competition between the coat's preferred curvature and the membrane's bending resistance (κ_C), we observed two distinct classes of assembly trajectories. In the majority of cases, we successfully reproduced a flat-to-curved transition. The coat initiated growth at a near-flat curvature ($H \approx 0$), followed by a sharp increase to a final value around $H_{\text{fin}} \approx 0.22\,\mathrm{L}_0^{-1}$, consistent with the biologically observed range for clathrin-coated vesicles. This curvature generation coincided with the inclusion of pentagonal defects, confirming that the model captures the essential topological requirements for vesicle formation.

In contrast to these successful transitions, a subset of simulations resulted in stalled, flat-growing lattices. In these instances, the coat continued to grow without initiating curvature, consistent with our findings on curvature memory and premature lattice stiffening. Regarding these stalled states, we noted a preliminary and qualitative observation: in the limited instances where node deletion occurred within a locked, flat lattice, the resulting structural disruption was followed by an increase in curvature. While the precise causality remains to be determined, this behaviour hints that the stalled lattice may store geometric frustration energy. It points toward a possible direction for future research, suggesting that stochastic bond rupture might play a role in destabilizing kinetic traps and allowing the system to access curved configurations.

Contents

List of Abbreviations iii

1 Introduction 1
- 1.1 Biological Background . . . 1
- 1.2 Physical Modelling of CME . . . 6

2 Theoretical Background 14
- 2.1 Membrane Bending Energy . . . 14
- 2.2 Connecting Flexural Rigidity to Bending Rigidity . . . 17
- 2.3 The Cooperative Curvature Model . . . 19
- 2.4 Graph Theory of a Clathrin Lattice . . . 20
- 2.5 Monte Carlo Algorithms . . . 22
- 2.6 Computational Techniques . . . 28

3 An Agentic Model of Clathrin Assembly in Spherical Geometry 33
- 3.1 General Idea . . . 33
- 3.2 Geometry & Notation . . . 35
- 3.3 Grid States, State Changes and Time Evolution . . . 37
- 3.4 Implementing Curvature Changes . . . 38
- 3.5 Energy Function . . . 40
- 3.6 Effective Membrane Rigidity . . . 45
- 3.7 Microstate Changes . . . 46
- 3.8 Macrostate Changes . . . 47
- 3.9 Constraining the Free Parameters . . . 58

4 Results 65
- 4.1 Metrics and Visualisations . . . 65
- 4.2 Sanity Checks . . . 69
- 4.3 Defining a Differentiable Minimal Energy . . . 73
- 4.4 Grid Growth at Fixed Curvature . . . 76
- 4.5 Grids at Variable Curvature . . . 88

5 Conclusion and Outlook 93
- 5.1 Summary of Findings . . . 93
- 5.2 Critical Assessment . . . 96
- 5.3 Outlook . . . 96

A Supplementary Calculations and Additional Theory **98**
A.1 Sampling Random Numbers . . . 98
A.2 Stereographic Projection . . . 102
A.3 Ideal Node Number . . . 102
A.4 Incorporating a Middle Layer . . . 103
A.5 Relaxation Times . . . 103
A.6 Preferred Hub-Angle . . . 105
A.7 Radius sampling . . . 106
A.8 Simpler Form of the Helfrich Hamiltonian . . . 106
A.9 Calculating the Base Rate . . . 107
A.10 Equivalence of BKL-KMC and rKMC . . . 108
A.11 Rescaling the Radius is Equivalent to Rescaling the Equilibrium Leg-Length 110

B Supplementary Figures **112**

C Additional Data **117**

List of Figures **119**

List of Tables **121**

Bibliography **122**

List of Abbreviations

Abbreviation	Meaning
AP2	Adaptor Protein 2
BKL	Bortz-Kalos-Lebowitz (synonym for rejection-free KMC)
CAM	Constant Area Model
CCM	Constant Curvature Model
CCV	Clathrin Coated Vesicle
CDF	Cumulative Density Function
CHC	Clathrin Heavy Chain
CLC	Clathrin Light Chain
CME	Clathrin Mediated Endocytosis
CoopCM	Cooperative Curvature Model
KMC	Kinetic Monte-Carlo
MC	Monte-Carlo
MCMC	Markov chain Monte Carlo
MH-MC	Metropolis-Hastings Monte-Carlo
ML	Machine Learning
MSD	Mean Squared Displacement
NNN	Next Nearest Neighbor
OOP	Object Oriented Programming
PDF	Probability Density Function
rfKMC	rejection-free KMC
rKMC	rejection-KMC
RKT	Receptor Tyrosine Kinases
TI	Truncated Icosahedron
VSSKMC	Variable Step Size KMC (synonym for rejection-free KMC)

Chapter 1

Introduction

1.1 Biological Background

The cell is one of the fundamental building blocks of almost all life, constituting a small self-replicating unit that is shielded from the outside. This protection is achieved through the cellular membrane, usually in the form of a lipid bilayer, spiked with membrane proteins. Such a bilayer has a thickness of around 5 nm and otherwise behaves like a fluctuating membrane, forming the cell. A thus-formed eukaryotic cell (i.e. with a membrane-protected *nucleus*) is of order $\approx 10\,\mu\text{m}$, see Figure 1.1. The membrane not only acts as a protective measure, but also as a filter for exchange of e.g., nutrients or information with the environment, also an essential ability.

Some membrane proteins, so-called *receptors*, act as sensors of external signals, allowing the transport of *information* across the cellular membrane. However, for other forms of *cargo* (e.g., nutrients, proteins or lipids), the membrane initially acts as an obstacle.

There are various mechanisms of achieving transport of such cargo. Those passing the cellular membrane are called (depending on the direction) *endocytic* or *exocytic pathways* (from Greek *endo*: within, *exo*: outside, *kytos*: cell and *-osis*: process). A certain type of transport pathway (also for intracellular transport) is that of *vesicle transport*. As explained in Figure 1.2, a part of one membrane *buds off* from the donor compartment and is transported to the target compartment where it fuses with the target membrane. The transported membrane shell is called the *transport vesicle*.

One of the challenges is to bend the membrane to the correct shape such that scission (the separation of the budded membrane from the main membrane) can take place. Every transport vesicle starts off as a budded region of membrane. The budding process is often initiated through specific proteins, which start to assemble as a flat coat on the membrane and shape it in the form of a *pit*. In this initial pit, the cargo can be collected. In the following, the pit matures to a budded region of coated membrane which encloses the cargo. The so created bud is disconnected in a process called *scission* from the membrane, creating a *coated vesicle* that encapsulates the cargo. Finally, the coat needs to dissolve such that the vesicle can fuse with its target membrane. During that process, the vesicle with its cargo inside is transported to its destination.

The four well-characterised vesicle-transport types are named by their major coat protein. They are *clathrin*, *COPI*, *COPII* and *retromer*, and their specific transport routes are shown in Figure 1.3.

In a more general sense, endocytic pathways are more than just trafficking routes. They seem to play a role in signal transduction, especially in signal attenuation (Sorkin et

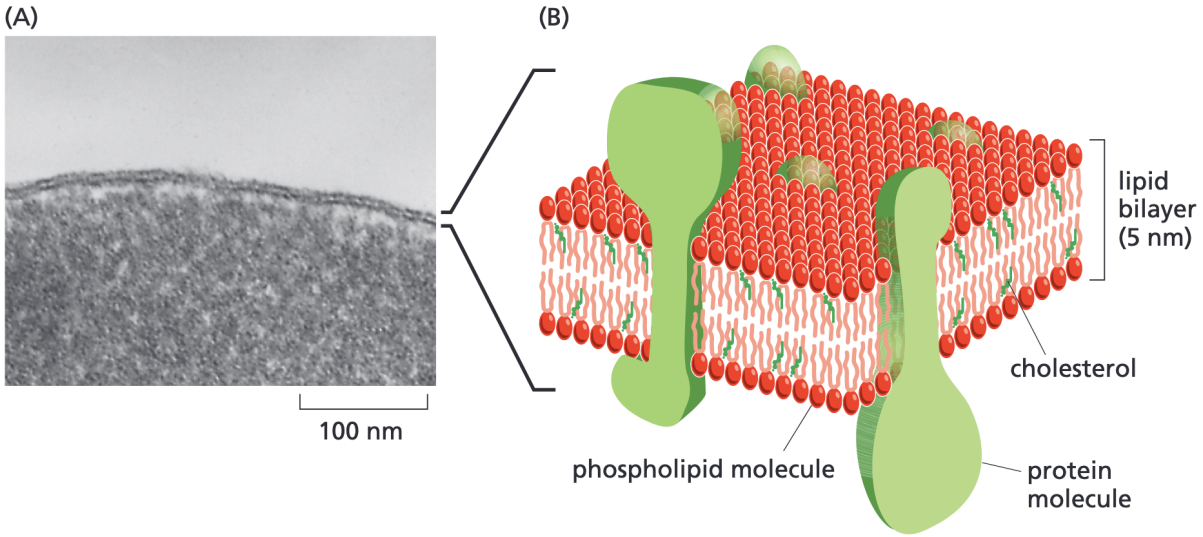

Figure 1.1: The structure of the lipid bilayer. (A) Electron micrograph of the lipid bilayer of a human red blood cell cross section. (B) Schematic depiction of the lipid bilayer. Phospholipid molecules orient their hydrophobic tails towards one another, forming the 5 nm bilayer with outward-protruding hydrophilic heads. Membrane proteins pierce this membrane. Taken from Alberts et al. (2022).

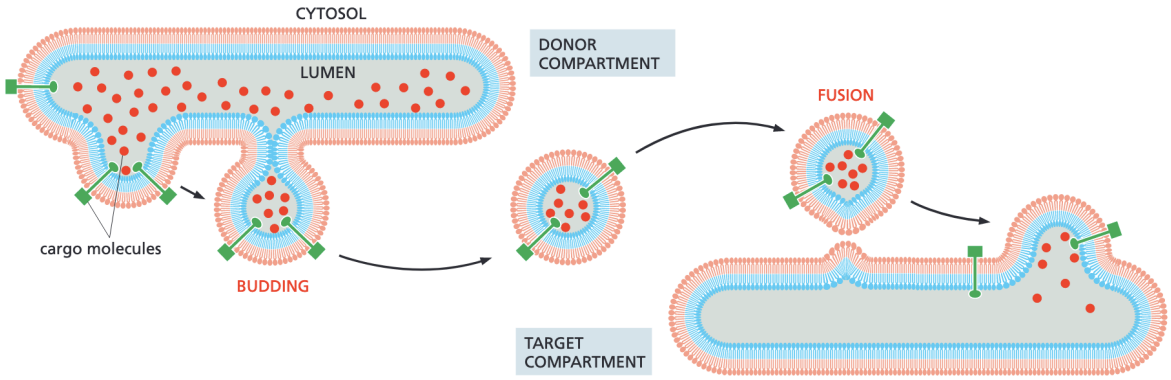

Figure 1.2: Vesicle transport between different compartments. To transport cargo (red dots) from one compartment to another, membrane can bend around the cargo, bud off and be transported from the donor to the target compartment. There, the vesicle fuses with the membrane, releasing the cargo to the target lumen. Taken from Alberts et al. (2022).

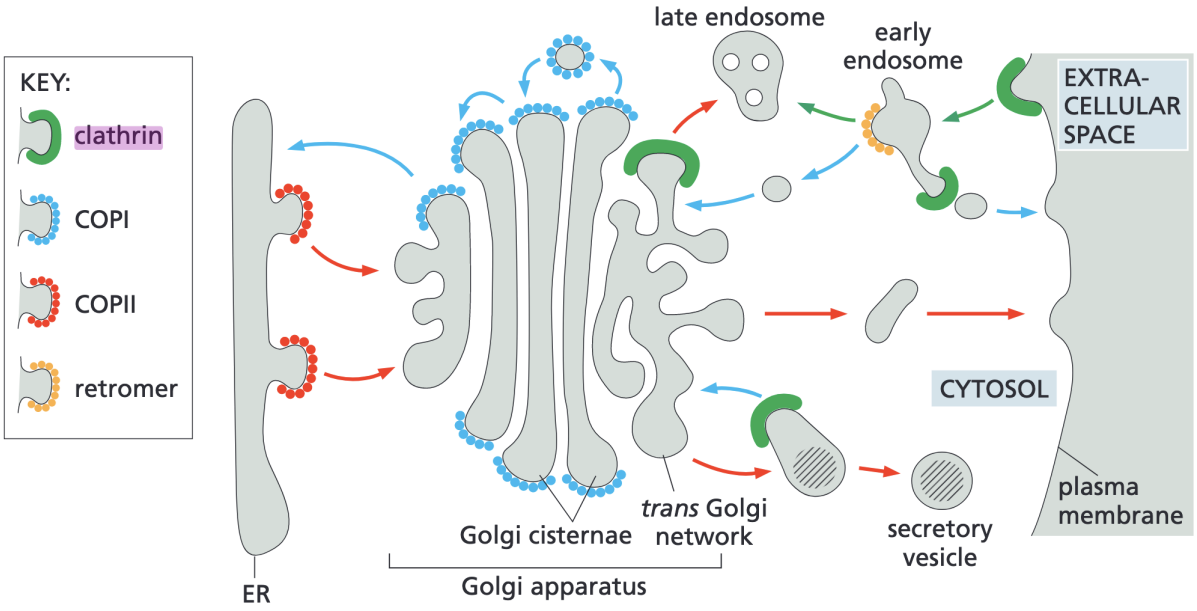

Figure 1.3: Where do different vesicular transport pathways take place. Different vesicle transport pathways are responsible for communication between different cell compartments. Through combinations of different pathways, the whole cell, from endoplasmic reticulum to cellular membrane can be reached. Figure taken from Alberts et al. (2022).

al. 2002). In addition, defective endocytosis can lead to impaired deactivation of *receptor tyrosine kinases*, potentially leading to a similar effect as overactivation of RTKs, which is strongly associated with carcinogenesis (Bache et al. 2004; Polo et al. 2004). Endocytosis is also a part of competing internalization pathways for receptors of the ErbB family, playing a role in the cardiac development and pathological signalling in neural disorders, e.g., Schizophrenia (Linggi et al. 2006; Ramanan et al. 2011; Arnold et al. 2005). And bacterial as well as viral pathogens (e.g., HIV, HSV and E. coli) hijack the endocytic machinery to gain access and infect cells (Eto et al. 2008; von Kleist et al. 2011).[1]

1.1.1 Clathrin Mediated Endocytic Vesicle Transport

In this work we focus on the endocytic pathway that is mediated through the major protein named *clathrin*, therefore called ***clathrin mediated endocytosis*** (CME). However, more than 50 cytosolic proteins are involved in CME (Haucke et al. 2018). The first observation of this pathway was in 1964 by Roth and Porter (Roth et al. 1964). Clathrin itself was first purified in 1975 by Pearse (Pearse 1975), where already the long-reaching network-like structure was noted[2]. She was also the one to propose the name *clathrin*, derived from the Latin word *clathri* for *cage*, in reference to the lattice-structure of clathrin coats.

[1] For a good and comprehensive overview on endocytosis, I recommend the following books and review articles. With a focus on molecular biology: Alberts et al. (2022), Brett et al. (2006), and Kaksonen et al. (2018), with focus on the different stages of CME: Edeling et al. (2006) and McMahon and Boucrot (2011), with a focus on the molecular structure: T. Kirchhausen et al. (2014), for an overview of the history of CME research: Maib et al. (2017), for the underlying physical principles: Zhang et al. (2015).

[2] Pearse also already gave an estimate on the number of clathrins involved in the formation of a clathrin coated vesicle of around 150.

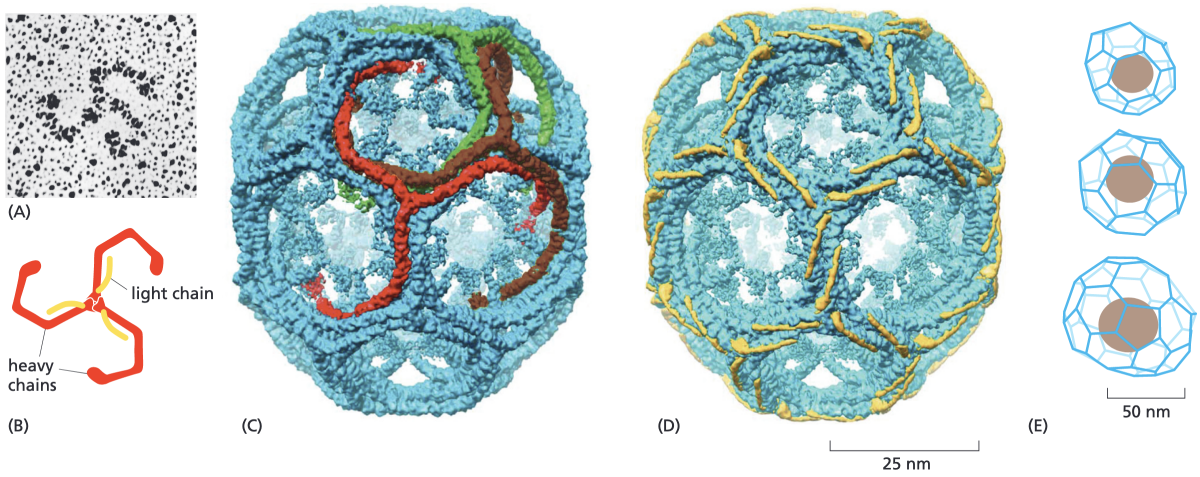

Figure 1.4: The structure of clathrin and clathrin coated vesicles. (A) Electron micrograph of a clathrin triskelion. (B) A clathrin triskelion is composed of three heavy chains (red) and three light chains (yellow). (C, D) Cryo-electron micrograph of a clathrin cage (i.e. no membrane) with heavy (C) and light (D) chains marked. (E) Diagrammatic structure of CCVs found in bovine brain, encapsulating the cargo. Figure taken from Alberts et al. (2022). Original sources are (A): Ungewickell et al. (1981), (C, D): Fotin et al. (2004), (E): Cheng et al. (2007)

In 1977, Anderson, Brown and Goldstein proposed in a three paper series (R. G. Anderson et al. 1977; R. G. W. Anderson et al. 1977; Goldstein et al. 1979) the first conceptual idea for the endocytic pathway we know today. It is responsible for communication between the cellular membrane, endosomes and the Golgi apparatus, specifically the trans-Golgi network (see Figure 1.3).

Clathrin itself is a cytosolic hetero-hexamer[3], consisting of three subunits called the *clathrin-chains*, which are themselves subdivided into the *clathrin-heavy-chain* (CHC) and the *clathrin-light-chain* (CLC). Three such clathrin-chains form a *triskelion*[4] (Figure 1.4 B), first observed by Ungewickell and Branton in 1981 (Ungewickell et al. 1981). Several triskelia can combine to form a clathrin-network. The precise nature of this network will be a major focus point of this thesis. As clathrin legs ($\sim 54\,\mathrm{nm}$) are much longer than the clathrin-clathrin distance in a clathrin lattice ($\sim 18\,\mathrm{nm}$), clathrin forms also bonds to next-nearest neighbours. A single clathrin has its three *legs* protrude separated by roughly 120° angles, such that all three legs combine to 360°, with the distant part of the leg bent again. Due to this geometry, such a flat network will generate a mostly hexagonal grid. The precise network geometry, especially when bent, is however more complicated. In general pentagons need to be included, see Figure 1.4. In experiments, clathrin isolated from cellular membrane was observed to form closed structures called *clathrin-cages* (see Figure 1.4 C, D and E) with cage radius $32-50\,\mathrm{nm}$ (Tagiltsev et al. 2021), which indicates that clathrin possesses a *spontaneous curvature*.

The light chains link to the actin cytoskeleton, which helps generate force for membrane budding and vesicle movement, and their phosphorylation regulates clathrin coat assembly.

On the membrane, clathrin-assembly induces so-called *clathrin-pits*, which are the

[3] i.e., comprised of six monomers, which are not all the same.
[4] The name again deriving from Greek *tri*: three, *skelos*: leg.

breeding ground for the final *clathrin-coated vesicle*. An essential part in the recruitment of clathrin is the formation of a layer of *adaptor proteins* which form a layer between the membrane and the clathrin-coat. One of the best understood adaptor proteins (which is involved in CME) is called *AP2* (adaptor protein 2). They are involved in a multitude of tasks. Besides clathrin recruitment, they bind to transmembrane protein cargos and receptors and imprint an initial curvature to the membrane. The precise interactions of the adaptor proteins are rather complex. They are often cooperative in nature. As an example, when AP2 binds to the lipid of the membrane, its conformation changes, increasing the binding rate to cargo receptors. When such a receptor binds to AP2, it increases the binding of said AP2 to the membrane. During all this, AP2 induces some curvature to the membrane, increasing the binding of further AP2.

In a simplified view, cargo together with the adaptor proteins lead to clathrin recruitment, which then assembles on the membrane. The AP2 binding process is cooperative, as it itself induces some form of membrane curvature, thereby enhancing the binding of further AP2 to the membrane. Once the clathrin accumulation starts, this effect is further increased.

Different adaptor proteins are responsible for budding at different membranes, as different membranes pose different challenges for the budding process. We are looking at a process where the cellular membrane is involved. In comparison to other membranes, it is quite stiff because of its high cholesterol-lipid composition. Endocytosis therefore requires additional membrane-bending proteins to generate enough force. One such class of proteins is called *BAR-domain proteins*[5]. Their most striking property is a crescent-shaped domain (the BAR-domain) that can impose its curvature on the membrane, see Figure 1.5, through electrostatic interactions with the lipids. These BAR-proteins are also involved in curvature sensing and the recruitment of *actin* and *dynamin* (Dawson et al. 2006), which play a crucial role in the scission process.

The region where the bent membrane goes over into the larger, flatter part of the membrane is called a *neck*. It is a saddle and as an almost free surface minimises its bending energy and is quite stable. Compared to the whole involved membrane, it carries only around 20% of the total energy (Frey, Ziebert, et al. 2019). Because of the stability of the neck, dynamin needs to be recruited to form a so-called *dynamin-collar* that wraps around the neck like a scarf, subsequently tightening and thereby rupturing the membrane for scission. At the same time actin is recruited, which builds a scaffold that pushes against or pulls on the bud. Together, the actin-dynamin machinery finalises the budding process (Cocucci, Gaudin, et al. 2014; Merrifield et al. 2002; Macia et al. 2006; Taylor, Perrais, et al. 2011; Taylor, Lampe, et al. 2012; Aguet et al. 2013).

After the coated vesicle has departed its origin membrane, the vesicle needs to lose its clathrin coat. This happens through a phosphatase depleting a phosphoinositide that is responsible for adaptor-protein binding together with auxilin peeling off the clathrin coat. While fusion is also an integral part of the endocytic pathway, it is independent of the clathrin that is used to initiate it. We will therefore not go into detail there.

As a precursor for CME, flat hexagonal lattices have been observed in the cell (Heuser 1980). They can vary in size, and the largest ones, called *clathrin plaques*, are up to several μm (Lampe et al. 2016). It is believed that these patches are used by the cell to interrogate its mechanical environment, i.e., the extracellular matrix and are formed in response to increasing substrate rigidity, or following frustrated endocytosis(Baschieri et al. 2018).

[5]Named after three proteins where BAR-domains are found in: *Bin*, *Amphiphysin* and *Rvs*.

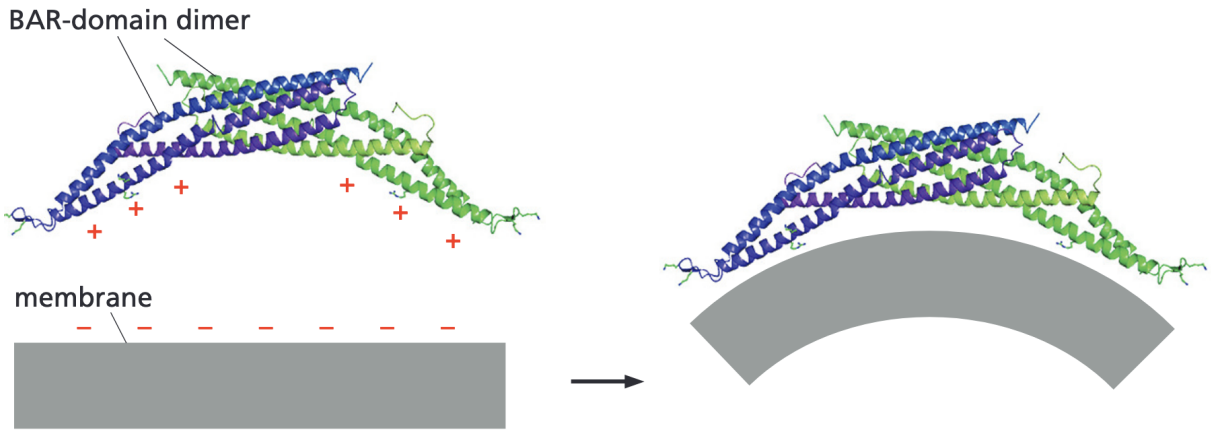

Figure 1.5: BAR-domain proteins. A BAR-domain protein has a crescent shape, which it can impose on a membrane after electrostatic interaction. Figure taken from Alberts et al. (2022).

Using fluorescence microscopy, it is possible to observe clathrin coat growth and clathrin density in pseudo-time. This has also revealed the existence of *aborted CCPs*, which are short-lived domains of clathrin that form, but do not mature and generate curvature, and finally dissolve after a lifetime of $\tau_{\text{aborted}} \lesssim 20\,\text{s}$ (Tom Kirchhausen 2009). It is believed that they do not recruit enough clathrin to progress maturation, which might relate to the cargo they are about to carry (Banerjee et al. 2012; Tom Kirchhausen 2009). There are different hypotheses on why these aborted pits form, e.g., that thermal fluctuations increase the AP2 concentration temporarily above a certain critical threshold such that coat formation is initiated, but cannot be sustained for long (Loerke et al. 2009).

1.2 Physical Modelling of CME

The biological perspective on membrane bending to achieve vesicular transport is already highly complex. However, looking at the problem from an abstract physical point of view reveals many new insights and questions. A comprehensive review that discusses the physical aspects of CME is (Haucke et al. 2018).

To get a sense on what such aspects are, let us look at the simplest uptake process: Adhesion energy mediated membrane wrapping. This can happen if the binding of membrane to a particle's surface releases the binding energy ω per bound area. Then the combined energy of the bent membrane (given by the Helfrich energy see 2.1, (Helfrich 1973)) bound to the particle is given by

$$E_{\text{tot}} = \int_A \mathrm{d}A \left(2\kappa H^2 - \omega\right) = -A\omega + 8\pi\kappa \tag{1.1}$$

since for a spherical particle the squared curvature $H = 1/R$ counteracts the integral over the area. Therefore, for given membrane stiffness κ, only particles above a critical size $A \geq 8\pi\kappa/\omega$ can be wrapped by a membrane using only adhesion energy as the driver of the process. This shows the necessity of assisted membrane wrapping for cargo under a certain size.

Coming back to clathrin, several aspects are of interest from a physics point of view. First let us start with the grid-structure of the clathrin lattice. As previously mentioned,

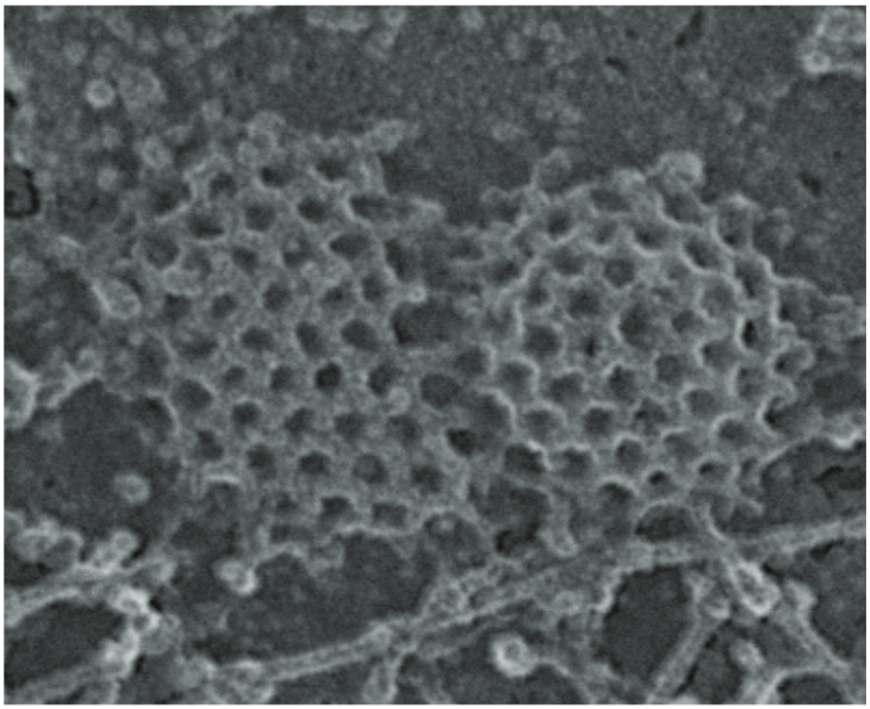

Figure 1.6: Domed clathrin patch adjacent to flat clathrin structure. A domed clathrin patch (right) can be seen adjacent to a flat clathrin patch. Both are only connected through 3 bonds. Taken from Sochacki et al. (2021).

a single clathrin triskelion possesses a trifold symmetry allowing it to assemble into hexagonal lattices on a flat substrate (Fotin et al. 2004). However, as is known from Euler's polyhedron formula, if such a lattice wants to form a closed cage, it needs to include (for a complete closure 12) *pentagons*, which constitute a suboptimal arrangement for clathrin. One of the main questions for biophysics to answer in regards to CME is how and when the pentagons necessary for a bent membrane are included. The ideas range from an inclusion from the side of the growing coat to geometric reshuffling during maturation (Albert J Jin et al. 1993). The inclusion of the pentagons might play a crucial role in answering the question what energy drives curvature generation. Sochacki et al. observed in 2021 (Sochacki et al. 2021) that curvature of clathrin coats happens even if the region of the membrane is separated from the cell, indicating that it is not additional energy or protein-aggregation that drives curvature generation. However, it was observed that already flat clathrin lattices are built with many defects (pentagons). These might store potential energy which gets released to propel curvature generation, like a locked Brownian ratchet (Sochacki et al. 2021). Regarding this question, they observed that curved clathrin lattices are often found adjacent to larger flat patches with visible ruptures, as can be seen in Figure 1.6.

The idea of such *preloaded* grids is strengthened by the observation that flat clathrin lattices are regularly built with fissures, which are only possible if adjacent lattice points are missing triskelia. These would not be visible using traditional imaging techniques. It was proposed that growth of such lattices is best described by an *Eden* growth model (Frey, Bucher, et al. 2020).

Also, it has been observed that under certain conditions (depending on ionic strength and acidity), clathrin can arrange in more extreme shapes, e.g., in cubes or tetrahedra

(Sorger et al. 1986; Heuser 1980). However, what all these shapes have in common is that they fulfil Euler's polyhedron formula.

Together with the questions regarding the grid structure are questions about the onset of curvature generation during pit-maturation. Over the years, different models have been proposed for the initiation of curvature acquisition during CCP formation. To be precise, the differences in the models try to answer questions about:

- The time evolution of curvature generation,
- The different impact of endocytic proteins on generating curvature,
- The spatial organisation of endocytic proteins.

From an abstract point of view, we have two competing processes during CME: the accumulation of proteins (1) and the generation of curvature (2). The precise temporal relation of these two processes is a topic of debate. In fact, it seems that for different processes, these relations might differ, as CCPs have been observed coexisting with clathrin plaques (Heuser 1980; Heuser 1989; J. Heuser et al. 1985; Larkin et al. 1986).

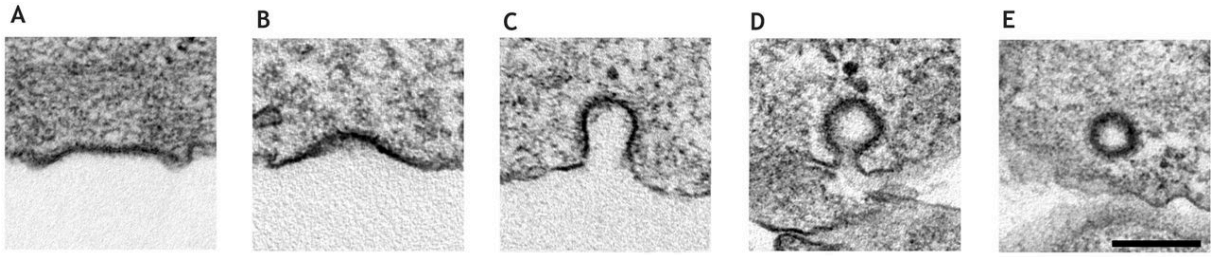

Figure 1.7: Different stages of the cellular membrane during CME. Shown is a gallery of different stages (from A to D) of electron micrographs of the cellular membrane during endocytosis. We see a growing flat clathrin coat (A), which curves slightly (B), then takes on a dome-like shape (C) to then form a tight neck (D). Finally, scission takes place (D) and the coated vesicle is transported into the cell. Taken from Haucke et al. (2018).

Over the years, two major schools of thought have developed, summarised in Figure 1.8. The first one is called the *constant curvature model* (CCM), which describes the maturation of CCPs by assembly of protein and thereby growth of the clathrin-coat at constant curvature (Cocucci, Aguet, et al. 2012; Tom Kirchhausen 2009; Saffarian et al. 2009). A main argument in favour of this model is that a hexagonal coat that would curl up to form a spherical coat would need major coat rearrangements to include the necessary pentagons, which are believed to be to too costly in energy (Albert J Jin et al. 1993). In this view, the coat already grows at the preferred curvature of the single clathrin molecule. This is also backed up by the high stiffness of the matured clathrin coat (Albert J. Jin, Prasad, et al. 2006). However, the curvatures of clathrin cages and CCVs do not match, see Table 1.1.

The other popular school of thought is called the *constant area model* (CAM). Here it is believed that the coat first grows as a flat patch to a fixed area size. Then, while the top-down projected area stays constant, the coat increases its curvature gradually to mature (Avinoam et al. 2015). In this picture, small amounts of clathrin do not need to bend its underlying membrane already at recruitment to the final curvature, but can do so gradually.

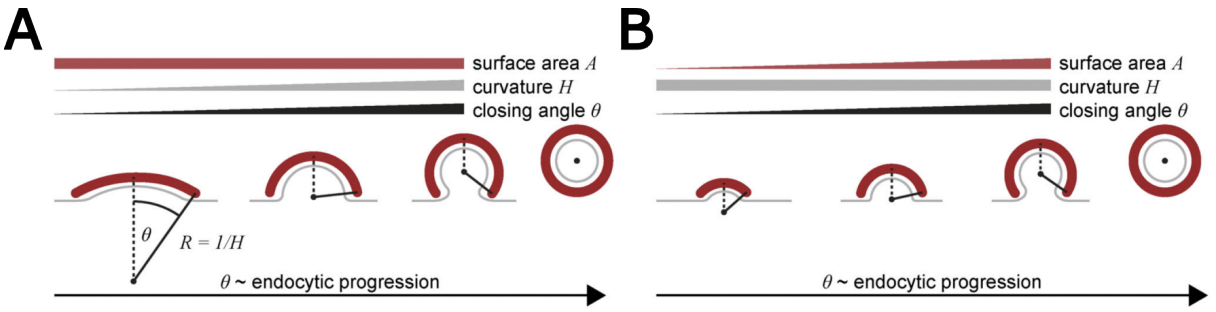

Figure 1.8: The constant area vs constant curvature model. A: In the constant area model, the clathrin coat first growths to its final area size. Then it starts to bend, increasing curvature simultaneously with the closing angle θ. B: In the constant curvature model, the coat initially already grows at its final curvature. The area growth is responsible for the increase in the closing angle θ. Taken and altered from Mund et al. (2023).

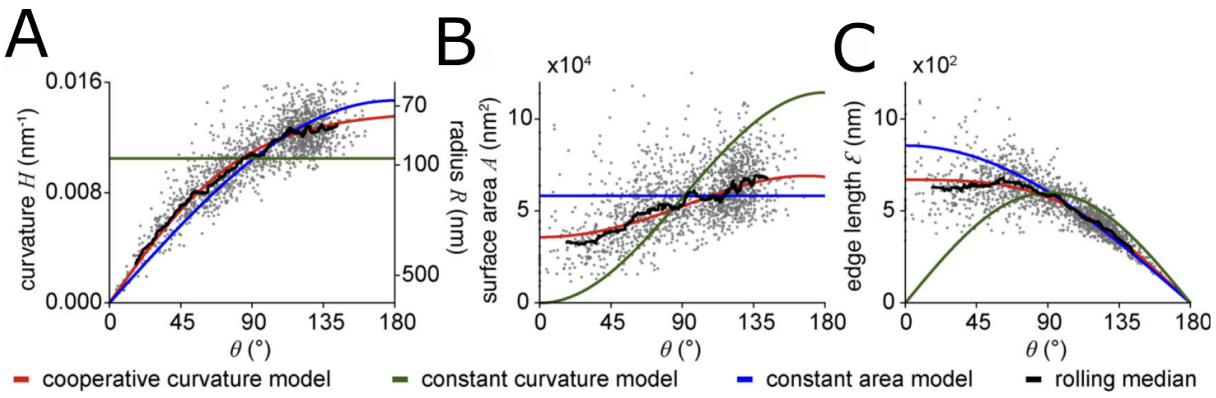

Figure 1.9: Evolution of geometric parameters against closing angle. Shown are curvature and bending radius (A), total surface area (B) and projected area (C) against the closing angle θ. The datapoints are taken from superresolution microscopy of many snapshots of endocytic sites. With a window width of 82 sites, the rolling mean (black) is shown. Overlayed are the predictions of the CAM, the CCM and the CoopCM. Figure taken and altered from Mund et al. (2023).

The difference in these two models can be summarised by the question of how the curvature H and the area A are coupled together during maturation. Recent studies suggest that neither the CCM nor the CAM are describing the CCP maturation completely correctly. While some have observed that the clathrin coat first grows to either about 70% of its final size (Bucher et al. 2018) or for approximately 30 s (Yoshida et al. 2018), at an almost flat curvature and only then does the major curvature generation start, others suggest that multiple membrane-bending trajectories might reflect the energetics of coat-assembly (Scott et al. 2018).

However, a recent study (Mund et al. 2023) combined superresolution microscopy with a new model fitting framework to extract geometrical parameters of endocytic sites. By sampling many snapshots of endocytic processes in fixed cells and sorting them by their progression along the endocytic timeline (i.e. along their closing angles θ, see Figure 1.8), they were able to measure the parameter evolution during CME, which are shown in Figure 1.9. While the predictions made by the CCM and CAM agree to a degree with the data, both show drastic deviations in at least one measured quantity. It was found that the curvature does not stay constant during closure and that closure starts at a finite surface area.

To remedy these problems, the *cooperative curvature model* (CoopCM) was proposed which, as the name suggests, emphasises the cooperative nature of the protein aggregation and curvature generation (Bucher et al. 2018; Frey and Schwarz 2020; Mund et al. 2023; Frey and Schwarz 2024). It is based on two equations. First, the area A changes in time in proportion to its circumference ε through a triskelion inclusion rate k_{on}:

$$\frac{\mathrm{d}A}{\mathrm{d}t} = k_{\text{on}}\varepsilon \tag{1.2}$$

Second, the curvature H can effectively be described as a function of the closing angle θ by:

$$\frac{\mathrm{d}H}{\mathrm{d}\theta} = \gamma\left(1 - \frac{H^2}{H_0^2}\right) \;\Rightarrow\; H(\theta) = H_0 \tanh\left(\frac{\gamma\theta}{H_0}\right) \tag{1.3}$$

using a differential equation that describes the slowing down of curvature generation towards the preferred curvature. This differential equation is readily solved to yield the dependence of a hyperbolic tangent. From the evolution of the curvature and the closing angle, all other geometric parameters can be derived. The resulting predictions are shown in Figure 1.9 as the red line. We see that the CoopCM yields an overall better agreement for every measured quantity than the CCM or the CAM.

By sorting all measured data points by their closing angle as a strictly monotonically increasing quantity during coat maturation, binning an equal number of events together and assigning each bin the same pseudotime, it was possible to generate a pseudotime resolved evolution for the measured quantities, see Figure 1.10. Now employing Equation (1.2) to link the data to time, the CoopCM again correctly predicted the pseudotime evolution.

The unclear temporal link between the growth of clathrin coated structures and their curvature generation has led some to the conclusion that an intermediate process must link both. An example would be the secondary recruitment of curvature-inducing proteins such as *epsin* and *CALM* (Chen et al. 1998; Ford et al. 2002; Maritzen et al. 2012). The mechanism by which these proteins can induce curvature could be through hydrophobic wedging (essentially a 'wedge-shaped' protein pressing into the bilayer) (Zimmerberg et al. 2006; Ford et al. 2002; Campelo et al. 2014) or through crescent-shaped proteins, such as the previously mentioned *BAR-domain proteins* (McMahon and Gallop 2005).

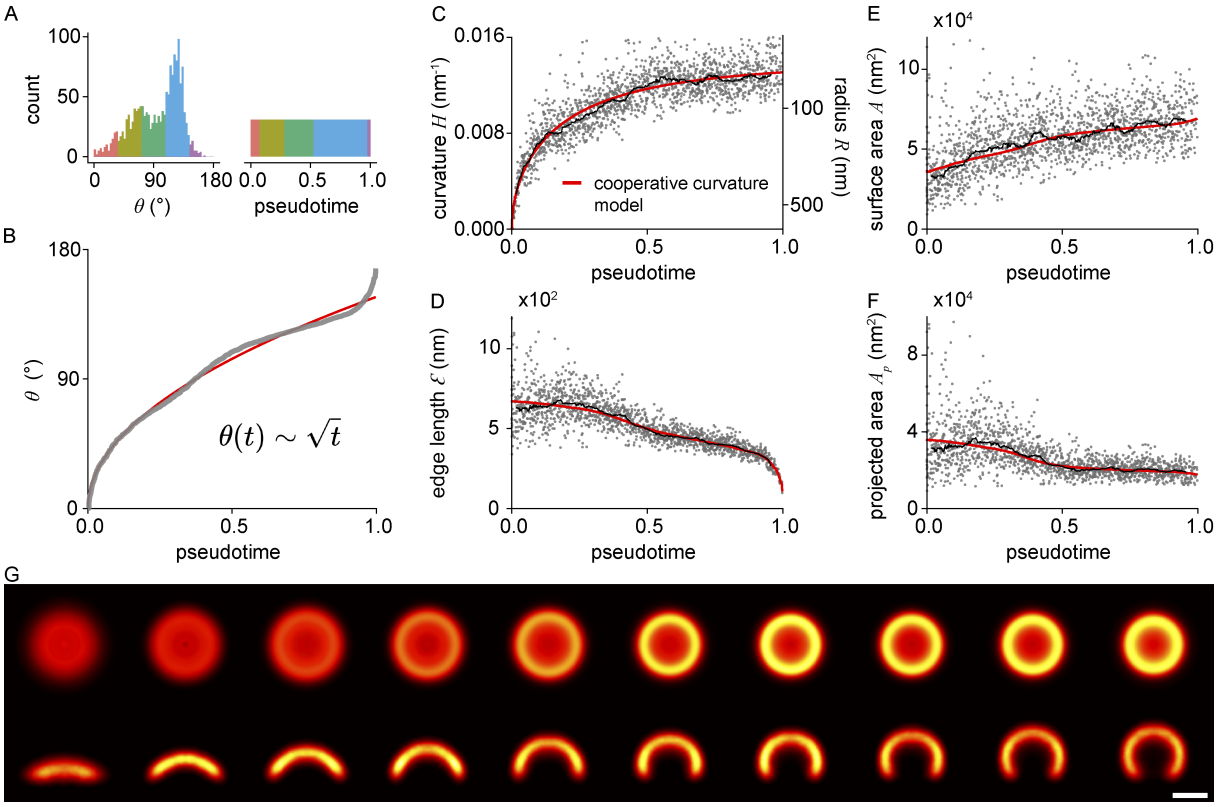

Figure 1.10: Pseudotime-resolution of clathrin coat evolution. (A): The number of events belonging to a closing angle θ. Many counts indicate a long-lived state. (C-F): By binning an equal number of events and assigning each bin the same ratio of the total pseudotime, the pseudotime resolution of closing angle, curvature, edge length, surface area and projected area was achieved. Overlaid is the prediction of the CoopCM in red. (G): Averages of superresolution microscopy images for the different endocytic stages. Each bin corresponds to the same number of snapshots. Different endocytic sites are rescaled to the average radius. The scale bar represents 100 nm. Figure taken from Mund et al. (2023).

To be able to answer questions about coat remodelling in detail, knowledge about the mechanical properties of clathrin and clathrin-coats is necessary. However, measuring these properties turns out to be quite challenging. One analysis done by Albert J. Jin and Nossal (2000) used statistical analysis of electron micrograph images to infer the flexural rigidity of the individual clathrin legs. Using assumptions on the geometry of a single leg, this can be mapped to a predicted bending rigidity of a clathrin coat, see 2.2. However, it was not yet possible to measure the bending rigidity of a pure clathrin coat. What was measured was the bending rigidity of CCVs using AFM (Albert J. Jin, Prasad, et al. 2006), yielding values of $\kappa_{CCV} \sim 200 - 500\,\mathrm{k_B T}$, which are about a factor of 10 higher than the combined stiffness of the membrane and the clathrin stiffness equivalent[6]. Interpretations have diverged from this point. Some follow the line of thought that therefore the clathrin coat itself is much stiffer than the flexural rigidity would indicate, arguing for the CCM (Saleem et al. 2015). Others argue that the clathrin coat itself has a low bending rigidity of approximately $10 - 30\,\mathrm{k_B T}$, and therefore the connecting layer (including the *AP2*) must bind to both the membrane and the clathrin in a way that the combined bending rigidity is increased. See (Albert J. Jin, Prasad, et al. 2006), using the theory of elasticity of composite shells. A list of the most important quantities for clathrin and its related structures can be found in Table 1.1.

[6] Calculated as explained in 2.2.

Quantity	Value	Reference
Membrane bending rigidity	$\kappa_{\text{Membrane}} \sim 10 - 25\ k_BT$	(Rawicz et al. 2000; Evans et al. 1990)
CCV membrane bending rigidity	$\kappa_{\text{CCV}} \sim 100 - 300\,\text{k}_\text{B}\text{T}$	(Albert J. Jin, Prasad, et al. 2006)
Binding energy per triskelion	$E_{\text{binding}} \sim 5 - 30\ k_BT$	(Den Otter et al. 2011; Nossal 2001; Saleem et al. 2015)
Clathrin arm length	$L_{\text{arm}} \sim 54 \pm 6\,\text{nm}$	(Albert J. Jin and Nossal 2000)
Clathrin-clathrin grid distance	$L_{\text{grid}} \sim 18.4 \pm 2.0\,\text{nm} := L_0$	(Sochacki et al. 2021; Vigers et al. 1986; Crowther et al. 1981)
Flexural rigidity of a clathrin arm	$EI_{\text{arm}} \sim 35\,\text{k}_\text{B}\text{T}\,\text{nm}$	(Albert J. Jin and Nossal 2000)
Flexural rigidity of a clathrin bond	$EI_{\text{bond}}/EI_{\text{arm}} \sim 16 \pm 6$	(Albert J. Jin and Nossal 2000)
Triskelia in a CCV	$N_{\text{CCV}} \sim 60 - 140$	(J. Heuser et al. 1985)
Triskelia in a clathrin cage	$N_{\text{CC}} \sim 40$	(Vigers et al. 1986)
Radius of a CCV	$R_{\text{CCV}} \sim 60 - 350\,\text{nm} = 3.24 - 19.02\,\text{L}_0$	
Curvature of a CCV	$H_{\text{CCV}} \sim 16 - 2.8\,\mu\text{m}^{-1} = 0.3 - 0.05\,\text{L}_0^{-1}$	
Radius of a clathrin cage	$R_{\text{CC}} \sim 30\,\text{nm} = 1.63\,\text{L}_0$	(Vigers et al. 1986)
Curvature of a clathrin cage	$H_{\text{CC}} \sim 32\,\mu\text{m}^{-1} = 0.61\,\text{L}_0^{-1}$	
Lipid bilayer thickness	$t_{\text{LB}} \sim 5\,\text{nm} = 0.27\,\text{L}_0$	(Albert J. Jin, Prasad, et al. 2006)
Clathrin coat thickness	$t_{\text{Clathrin}} \sim 4.5\,\text{nm} = 0.25\,\text{L}_0$	(Albert J. Jin, Prasad, et al. 2006)
Clathrin-lipid-connecting layer thickness	$t_{\text{conn. layer}} \sim 18\,\text{nm} = 0.97\,\text{L}_0$	(Albert J. Jin, Prasad, et al. 2006; Vigers et al. 1986)
Membrane thickness of a CCV	$t_{\text{CCV}} \sim 27\,\text{nm} = 1.46\,\text{L}_0$	(Albert J. Jin, Prasad, et al. 2006)
Mean lifetime of a CCV	$\tau_{\text{CCV}} \sim 30 - 120\ s$	(Loerke et al. 2009)
Mean lifetime of an aborted CCP	$\tau \lesssim 20\,\text{s}$	(Tom Kirchhausen 2009)
Mean area per triskelion	$A_{\text{triskelion}} \sim 433\,\text{nm}^2 = 1.28\,\text{L}_0^2$	(Sochacki et al. 2021; Vigers et al. 1986; Crowther et al. 1981)
Projected leg-leg angle (2D) in coat	$\phi = 120.0 \pm 13.3°$	(Sochacki et al. 2021)
Leg-leg angle (3D) in coat	$\theta = 119.3 \pm 13.0°$	(Sochacki et al. 2021)
Projected leg-leg angle (2D) single clathrin	$\phi = 120.0 \pm 37.0°$	(Albert J. Jin and Nossal 2000)

Table 1.1: Reference numbers for clathrin and related structures. Listed are from top to bottom: The bending rigidity of a typical lipid bilayer membrane. The bending rigidity of the coat of a CCV, including the cellular membrane, the layer of adaptor proteins, bound cargo and the bound clathrin coat. The combined bending rigidity will be greater or equal to the sum of the individual bending rigidities, depending on the binding between the layers (see (Albert J. Jin and Nossal 2000)). The binding energy per triskelion, which is released upon binding of the triskelion to the clathrin-coat or membrane. The values given are no measurements but theoretical estimates coming from different approaches. It is therefore not really clear if this is the binding energy for inclusion into the clathrin-lattice, binding to the membrane, or a combination of both. The length of a clathrin arm. This is not the distance to the first partner in a clathrin-lattice. This would be the clathrin-clathrin grid distance, which is the distance two different bound clathrins are apart in a typical coat. This distance defines the length scale L_0. The flexural rigidity of a clathrin arm in the unbound state. Multiple bound arms are described by the flexural rigidity of a clathrin bond, consisting of multiple arms. Depending on the geometry, the relative factor between the flexural rigidities can change, see (Albert J. Jin and Nossal 2000). To relate the flexural rigidity to the bending rigidity, see 2.2. The involved clathrin triskelia in a clathrin coated vesicle or in a clathrin cage. The typical radii and curvatures of clathrin coated vesicles or clathrin cages. The typical thicknesses of membranes. Between the cellular membrane and the clathrin coat, which are both of similar thickness, a layer of connecting proteins is placed, sometimes including cargo, which comprises the major thickness of the combined CCV membrane. The lifetime of a CCV until it is dissolved. Aborted CCPs live much shorter. The average area per triskelion can be found from L_{grid} and the fact that any triskelion occupies half the area of a regular hexagon with side length L_{grid}. The two leg angles for clathrin in a lattice, once projected in a top-down view (restricted to exactly 120°) and once the real inter-leg angle. For a bent surface, this is strictly less than 120°. At last the projection angle for free clathrin.

Chapter 2

Theoretical Background

2.1 Membrane Bending Energy

The first description of a membrane's bending energy is now usually attributed to Wolfgang Helfrich, who described it in his 1973 paper (Helfrich 1973) during his time at F. Hoffmann-La Roche in Basel. However, already in 1970, Peter Canham, independently of Helfrich, published a version of this bending energy (Canham 1970), so now it is also referred to as the **Helfrich-Canham bending energy**. In its modern form, it states that the energy of a bent smooth membrane described by the manifold $\mathcal{M}$, is given by[1]

$$E = \int_{\mathcal{M}} \left\{ \kappa \left(H - \overline{H} \right)^2 + \kappa_G K \right\} \mathrm{d}A \tag{2.1}$$

where H is the **mean** curvature, K is the **Gaussian** curvature, $\overline{H}$ is the **spontaneous** curvature and κ and κ_G are material parameters called the **bending** and **saddle-splay** moduli.

For any point on a one-dimensional differentiable curve, it is possible to fit a tangential circle to that point, called an *osculating circle*. The curvature of the line at that point is then referred to as the inverse of that radius, $k = 1/R$.

For a two-dimensional smooth manifold, it is again possible to fit an osculating circle at any point of the manifold. However, depending on the rotation of the circle, its radius may change. By rotating the circle around, one can find the minimal and maximal radius, R_1 and R_2, which define the *principal curvatures* $k_1 = 1/R_1$ and $k_2 = 1/R_2$. From these two principal curvatures, one then defines the *mean curvature* H

$$H := \frac{k_1 + k_2}{2} = \frac{1}{2}\left(\frac{1}{R_1} + \frac{1}{R_2} \right) \tag{2.2}$$

as the average of the curvatures and then the *Gaussian curvature* K

$$K := k_1 k_2 = \frac{1}{R_1 R_2} \tag{2.3}$$

as the product of the curvatures. A convex membrane has both $H, K > 0$. A saddle is characterised by $K < 0$, and H can take on values positive, zero and negative.

[1] There are different conventions on the prefactors of κ. For this thesis, we have adopted the variant without a factor of 2.

For a closed manifold $\mathcal{M}$, the integral over the Gaussian curvature is **constant**. This statement is the *Gauß-Bonnet theorem*, and it states that the constant is proportional to the *Euler-characteristic* $\chi(\mathcal{M})$ of the manifold. It states that

$$\int_{\mathcal{M}} K \, \mathrm{d}A = 2\pi\chi(\mathcal{M}) \tag{2.4}$$

and therefore the Gaussian curvature only changes if the topology of the observed surface changes. For our purposes, the Euler characteristic of the sphere is $\chi = 2$. As long as the membrane only bends, the contribution of the Gaussian curvature stays constant over the whole system. This does not mean that locally the energy cannot change, but the overall contribution stays constant. When a vesicle buds off, however, the Gaussian contribution becomes important, as the topology changes.

2.1.1 Spherical Membrane

For our purposes, we are interested in the special case of a membrane that can fit onto a sphere. In that case, the two principal curvatures are identical and equal to the inverse radius R, i.e. $k_1 \equiv k_2 := \frac{1}{R}$. Then we find $H = \frac{1}{R}$ and $K = H^2$ and the curvature is constant. In this case, the Helfrich Hamiltonian (2.1) assumes the simpler form derived in A.8 given by

$$E(A, H) = \kappa' \left(H - \overline{H}'\right)^2 A + \sigma A \tag{2.5}$$

where $\sigma_P \propto \overline{H}^2$ and acts as an additional effective *pseudo surface tension*.

When we describe the bending of membrane during CME, the local curvature is small, so the membrane can be approximated as flat, i.e. with vanishing spontaneous curvature $\overline{H} = 0$, as a clathrin coated vesicle has a radius of order $R \sim 100\,\mathrm{nm}$, where the membrane has a radius of order $R \sim 10\,\mu\mathrm{m}$ and can therefore be regarded as quasi-flat. In that case, the pseudo surface tension $\sigma_P \to 0$, and we are left with the most simple formulation of the Helfrich-energy with just a shifted bending rigidity.

As for the description of the clathrin coat, the σ_P-term can be absorbed into a modification of the polymerisation energy per clathrin.

2.1.2 Discrete Formulation of Helfrich Hamiltonian

Our later code is formulated in terms of single clathrin nodes. We want a large coat to resemble a continuous membrane, therefore we need a discretisation of the Helfrich energy suitable to our case. To compare to other discretisations, see e.g. Zhu et al. (2022). For the clathrin-dynamics, we want a local energy formulated in terms of an energy E_i for every i-th clathrin hub, which, when accumulated over the entire coat, reproduces the Helfrich energy.

The correct limiting behaviour is formulated in terms of large clathrin number at constant area

$$\lim_{N\to\infty,\, A=\text{const.}} \sum_i^N E_i = \int_A \mathrm{d}A \, \kappa \left(H - \overline{H}\right)^2 \tag{2.6}$$

This local energy E_i is given by

$$\boxed{E_i \approx \frac{4\kappa}{L_0^2} \frac{A_{\mathrm{Cl}}}{M_{\max}} \sum_{j=1}^{M_i} (\chi_j - \chi_0)^2} \tag{2.7}$$

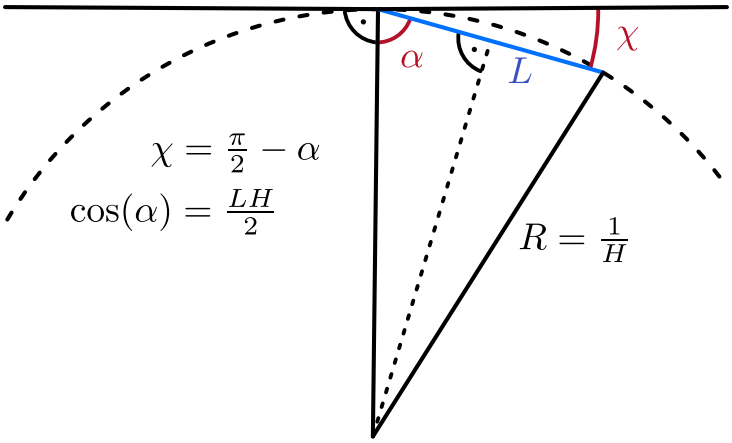

Figure 2.1: Pucker angle and its relation to the leg length.. The pucker angle χ describes how far a leg (blue) is dipped away from the the tangential plane (solid line) at the sphere (dashed line). Using simple trigonometry, χ can be brought into relation to the length L of the leg, see Equation (2.8).

for a lattice where each hub has exactly M_i and at most $M_{\max}$ legs that connect to a partner. The area of a single clathrin is A_{Cl} and χ is the so-called *dipping angle* or *pucker angle* which describes how far a leg is dipped into the tangential plane, see Figure 2.1.

To show that this is the correct microscopic energy, we have to relate the pucker angle to the leg length L. By taking the sin of $\chi = \pi/2 - \alpha$ from Figure 2.1, we get

$$\chi_j \approx \sin(\chi_j) = \sin(\pi/2 - \alpha_j) = \cos(\alpha_j) = \frac{L_j H}{2} \tag{2.8}$$

that relates the pucker angle χ_j of the jth leg to the length of the jth leg L_j. Then

$$(\chi_j - \chi_0)^2 = \left(\frac{L_j H}{2} - \frac{L_0 H_0}{2}\right)^2 \approx \left(\frac{L_0}{2}\right)^2 (H - H_0)^2$$

if we assume that, on average the leg-length L only deviates by small amounts from the equilibrium length L_0. If the coat becomes densely connected, $M_i \to M_{\max}$ and the sum becomes independent of j and cancels with the $1/M_{\max}$, and we reach the known Helfrich-energy:

$$\begin{aligned} E_{\mathrm{micro}} =& \frac{4\kappa}{L_0^2} \frac{A_{\mathrm{Cl}}}{M_{\max}} \sum_{j=1}^{M_{\max}} \left(\frac{L_0}{2}\right)^2 (H - H_0)^2 \\ =& \kappa (H - H_0)^2 A_{\mathrm{Cl}} \end{aligned}$$

The Helfrich bending rigidity κ is then connected to the microscopical spring constant for the pucker angle χ by:

$$\boxed{k_\chi = \frac{2}{3} \frac{A_{\mathrm{Cl}}}{L_0^2} \kappa} \tag{2.9}$$

with $M_{\max} = 6$ in our case.

2.1.3 Two Membrane Energies

For our purpose, we need to know how the combination of two membranes behaves with regard to their curvature. For later naming conventions, think about a combined Helfrich

energy of the form

$$E(H) = A\left\{\kappa_C(H - H_0)^2 + \kappa_M H^2\right\} \tag{2.10}$$

The first term describes the effective bending energy of the clathrin coat with the effective bending rigidity κ_C. The second part describes the membrane bending energy, with vanishing spontaneous curvature. We can rewrite this energy to the form

$$\begin{aligned} E(H) =& A\left(\kappa_C + \kappa_M\right)\left\{\left(H - \alpha H_0\right)^2 + \alpha H_0^2\left(1 - \alpha\right)\right\} \\ =& A\kappa_{\text{tot}}\left(H - H_{\text{min}}\right)^2 + A\sigma \end{aligned} \tag{2.11}$$

with effective parameters

$$\boxed{\kappa_{\text{tot}} := \kappa_C + \kappa_M \quad , \quad \alpha := \frac{\kappa_C}{\kappa_C + \kappa_M} \in [0,1] \quad , \quad H_{\text{min}} = \alpha H_0 \quad , \quad \sigma := \kappa_{\text{tot}} H_0^2 \alpha(1-\alpha)} \tag{2.12}$$

The sum of the membranes behaves as a membrane with the combined bending rigidity. However, the energy-minimising curvature H_{min} is now shifted towards lower values, depending on the ratio α of coat to total rigidity, which only makes sense, since the membrane pushes the system towards low curvatures. We also see a surface-tension term appear, which becomes relevant for the case $\alpha \approx 0.5$, so $\kappa_M \approx \kappa_C$. The combined energy can be seen in Figure 2.2.

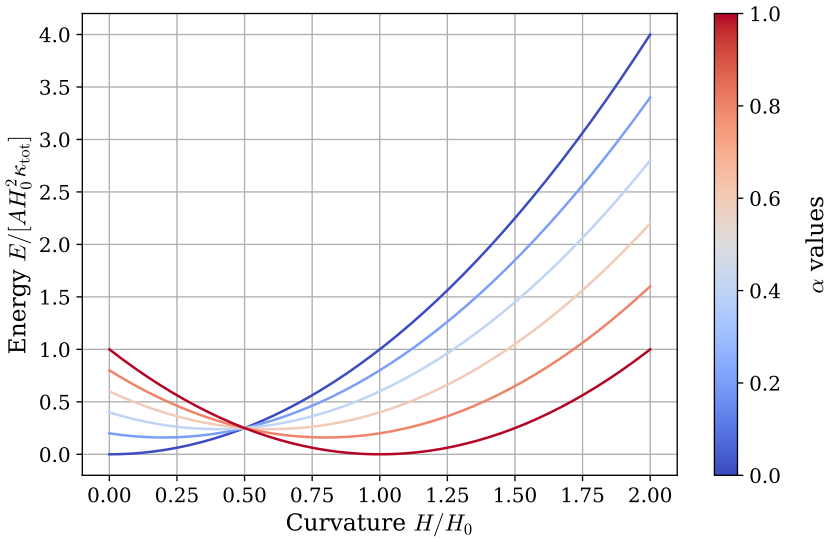

Figure 2.2: Combined bending energy of two membranes. The energy, see Equation (2.11), of two membranes kept at constant combined bending rigidity κ_{tot}. For different ratios α of the individual rigidities, the minimum of the energy shifts between $H_{\text{min}}/H_0 \in [0,1]$.

2.2 Connecting Flexural Rigidity to Bending Rigidity

As mentioned in the introduction, direct measurements of only clathrin's bending rigidity are hard. However, one may look at isolated clathrin and try to derive from the statistics

of the arm movements an analogue of the bending rigidity. For general information about the energetics of bent membranes or rods, one may refer to the classic Lifšic et al. (1986). For adaptation in biology, refer to Gittes et al. (1993).

For clathrin, we may describe the energy due to bending deformation of the isolated arms using the theory of elastic rods. There, the bending energy E_{bend} of a rod of arc length s_{tot} is described using

$$E_{\text{bend}} = \frac{1}{2} EI \int_0^{s_{\text{tot}}} [C(s) - C_0(s)]^2 \, \mathrm{d}s \tag{2.13}$$

where $C(s)$ is the curvature at arc length s. The parameter EI is called the *flexural rigidity* and has units of energy times length, usually in units of $\mathrm{k_B T\,nm}$.

By measuring the shape function $\theta(s)$ and decomposing it into *normal modes*, $\theta(s) = \frac{1}{\sqrt{2L}} \sum_{n=0}^{\infty} a_n \cos(n\pi s/L)$, one can measure the flexural rigidity through measurements of the mean squared modes

$$\langle (a_n - a_n^0)^2 \rangle = \frac{k_B T}{EI} \left(\frac{L}{\pi n} \right)^2$$

see (Albert J. Jin and Nossal 2000), which gives a value of $EI_{\text{arm}} \approx 35\,\mathrm{k_B T\,nm}$, see table 1.1. EI_{arm} is the flexural rigidity of an individual clathrin arm. In real coats, multiple arms, however, are responsible for a bond, at least two, and at most four (Crowther et al. 1981). If the substance is isotropic and elastic, the flexural rigidity may be written as

$$EI = E \cdot I$$

where E is the Young's modulus and I is the moment of inertia. A bond comprised of multiple arms differs only in its moment of inertia, i.e.

$$EI_{\text{bond}} = E_{\text{clathrin}}\, I_{\text{bond}} = E_{\text{clathrin}}\, g I_{\text{arm}} = g EI_{\text{arm}}$$

In Albert J. Jin and Nossal (2000), different possible arrangements of four arms are shown with their corresponding g-factor. For reasonable arrangements of four arms, they find $g \in [12, 26]$ with an average of $g \approx 16$. Four bonds together do not make the bond 4 times as stiff, but 4^2 times as stiff. For less arms, g is also less.

At last, we need to relate the flexural rigidity to the bending rigidity. Assume we have a coat comprised of single rods (the clathrin arms), which we want to bend in one direction. For the time being, let us assume a vanishing spontaneous curvature. Then the Gaussian curvature vanishes and $H = \frac{1}{2R}$, i.e., the energy of the bent membrane is

$$E_{\text{Helfrich}} = A \kappa_C \frac{1}{4R^2} \tag{2.14}$$

At the same time, on the rod-level, two rods need to be bent over their whole length l at curvature $C = 1/R$ to accomplish this bending, so the energy of a single grid-cell is given as

$$E_{\text{rod bending}} = 2\,\frac{1}{2} EI_{\text{bond}} \int_0^l C(s)^2 \mathrm{d}s = EI_{\text{bond}} \frac{l}{R^2} \tag{2.15}$$

As this describes the energy of a single grid-cell, we can equate (2.14) and (2.15) by choosing $A = A_{\text{grid cell}} \approx l^2$. We then get

$$\kappa_C \frac{l^2}{4R^2} \approx EI_{\text{bond}} \frac{l}{R^2}$$

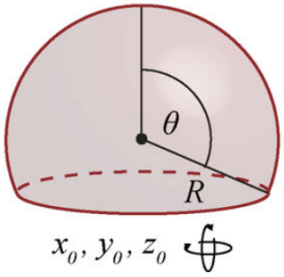

Figure 2.3: Geometry of a spherical cap. A spherical cap can be described using the *invagination angle* θ, which is sometimes also named the *closing angle*. Figure taken from Mund et al. (2023).

which finally gives, after substituting in EI_{arm}, the relation we are looking for

$$\boxed{\kappa_C \approx g\frac{EI_{\text{arm}}}{l}} \tag{2.16}$$

Looking at table 1.1, we find $EI_{\text{arm}} \approx 35\,\text{k}_\text{B}\text{T}\,\text{nm}$ and $l \approx 50\,\text{nm}$ giving values of

$$\kappa_C \approx g\ 0.7\,\text{k}_\text{B}\text{T}$$

Therefore, if the bending stiffness would be purely due to clathrin's legs bending, we would see a maximal bending rigidity of $\kappa_C^{\text{max}} \approx 15\,\text{k}_\text{B}\text{T}$.

2.3 The Cooperative Curvature Model

The cooperative curvature model (CoopCM) was introduced in Mund et al. (2023). It describes a growing clathrin coat using a *spherical cap* as shown in Figure 2.3. Such a geometry can be described using the two quantities of spherical radius R and closing angle θ. The area A of the spherical cap is found by integration:

$$A(\theta, R) = \int_0^{2\pi} \text{d}\varphi \int_0^{\theta} \text{d}\theta \sin\theta R^2 = 2\pi R^2(1-\cos\theta) \tag{2.17}$$

An important special case is the one of a flat patch, which is still described in this model in the limit of $\theta \to 0, R \to \infty$ with $R^2(1-\cos\theta)$ staying constant.

The cooperative curvature model is based on two assumptions. The first relates the growth of the area A of the spherical cap to its circumference ε:

$$\frac{\text{d}A}{\text{d}t} = k_{\text{on}}\varepsilon \tag{2.18}$$

as it assumes that new clathrin is only introduced at the brim of the cap. The second assumption relates the curvature $H = 1/R$ of the cap with radius R to the invagination angle θ:

$$\frac{\text{d}H}{\text{d}\theta} = \gamma\left(1 - \frac{H^2}{H_0^2}\right) \tag{2.19}$$

Solving the above leads to

$$H(\theta) = H_0 \tanh\left(\frac{\gamma\theta}{H_0}\right) \tag{2.20}$$

which can be used to find the area at which the curvature generation initially starts. Since $H(\theta = 0) = 0$, curvature generation starts at $\theta = 0$, and we find:

$$A_{\text{init.}} := \lim_{\theta \to 0} A(\theta) = \lim_{\theta \to 0} \frac{2\pi}{H_0^2} \frac{1 - \cos\theta}{\tanh\left(\frac{\gamma\theta}{H_0}\right)^2} = \frac{\pi}{\gamma^2} \tag{2.21}$$

This links the growth rate γ of the curvature with closing angle θ to the initial radius of the flat clathrin patch, $R_{\text{init.}} = 1/\gamma$. From this we see that the constant curvature model is the limiting case of $\gamma \to \infty$. However, the constant area model cannot be fully recovered.

While the first Equation (2.18) follows naturally from physical considerations, the second (2.19) is largely phenomenological: it successfully captures the observed behaviour, but its exact mechanistic origin remains unclear. Therefore in 2024, the CoopCM was extended to allow for a derivation of (2.19) (Frey and Schwarz 2024). It assumes an energy of the membrane-clathrin system (ignoring the connecting layer) of

$$E = \int_{\text{Mem.}} \mathrm{d}A \left[2\kappa_M H^2 + \sigma\right] + \int_{\text{Coat}} \mathrm{d}A \left[-\mu + 2\kappa_C (H - H_C)^2\right] + \zeta\varepsilon \tag{2.22}$$

The membrane's energy is comprised of the usual bending term with vanishing spontaneous curvature and a surface tension term. The coat's energy has a polymerisation energy per area μ and a spontaneous curvature H_C. The possibility of a line tension ζ taxing the edge of the coat is included. Using overdamped dynamics for the time evolution of curvature H, i.e.

$$\alpha \frac{\partial H}{\partial t} = -\frac{\partial E}{\partial H} \tag{2.23}$$

with a proportionality constant α, and using

$$\dot{\theta} \propto \frac{1}{\theta} \Rightarrow \theta(t) \sim \sqrt{t} \tag{2.24}$$

which can be derived from Equation (2.18), the time derivative can be switched to a derivative w.r.t. θ. Then one finds under the assumption $\alpha \sim \theta$ that

$$\frac{\partial H}{\partial \theta} \propto -\frac{1}{A\kappa_C} \frac{\partial E}{\partial H} \tag{2.25}$$

where A is the area of the clathrin coat, and κ_C is its bending rigidity. Under the assumptions that the polymerization energy increases linearly in H and that the coat stiffens up with a power of three

$$\begin{aligned} \mu(H) &= \frac{\mu^{\text{fin.}}}{2} \left(1 + \frac{H}{H_C}\right) \\ \kappa_C(H) &= \kappa_C^{\text{fin.}} \left(\frac{H}{H_C}\right)^3 \end{aligned} \tag{2.26}$$

this leads in leading order to the second equation of the cooperative curvature model, (2.19).

2.4 Graph Theory of a Clathrin Lattice

Even though the clathrin network might be more dynamic than initially anticipated and the clathrin-clathrin interactions can be complicated, from an abstract point of view a

clathrin-lattice can be thought of as a *graph*. This will allow us to place constraints on the possible configurations of the clathrin-lattice. For this, we ignore the next-nearest-neighbour bonds and simply look at clathrin-hubs that can be connected to other clathrin-hubs. For a simple yet comprehensive introduction to graph theory, see Wilson (2009).

Speaking in graph theoretic terms, the clathrin-hub would be called a *vertex* of the graph, and the direct bond would be called an *edge*. The *degree* of a vertex is the number of edges ending in it. Physically, clathrin always has three legs going out of it, and can have three legs coming into it. We can represent this either by having *directed legs* or simpler by noticing that if hub A is connected to B, then B is connected to A, and the relation can be expressed with an undirected single edge. In this case, if we consider a leg as an edge only if it connects two clathrin-hubs, then the degree of each vertex is $D \leq 3$. As clathrin-legs don't connect to their own hub, the graph has no *loops*. It is not observed that two hubs form multiple direct bonds between another. This corresponds to the graph not having *multiple edges*. An undirected graph without loops and multiple edges is called an *undirected simple graph*.

It is also not observed that clathrin-legs cross another in a built lattice. A graph that can be drawn without its edges crossing each other is called a *planar graph*. A completed cage is in fact also a planar graph, since it can be projected onto the plane without any two edges crossing another using, e.g., a stereographic projection. For more details on this projection method, see A.2. If in such a completed cage every vertex (clathrin-hub) has built exactly three direct bonds, then the degree of each vertex is exactly $D = 3$ and the resulting graph is said to be r-regular with $r = 3$. For our case, r-regular graphs of degree three are said to be *cubic graphs*.

A sequence of n distinct vertices v_i with $v_0 = v_n$ that are each adjacent (i.e. connected through an edge) is called an *n-cycle*. For large enough graphs, such that no 6-cycle can span the equator, the number of hexagons is equivalent to the number of 6-cycles in our graph. However, for smaller graphs, this relation is not so direct.

Every planar projection of a planar graph separates the plane into distinct regions separated by edges. These regions are called *faces*. Every such graph has an outlying face, called the *infinite face*. However, there is no canonical projection for a planar graph. As an example, think of a clathrin cage, which is a three-dimensional representation of a planar graph. This can be shown using a stereographic projection, where the clathrin cage is mapped onto the plane with no two edges overlapping. However, the choice of projection point makes the projection ambiguous.

However, it can be shown (Wilson 2009) that for any choice of planar representation of a planar graph, the number of edges E, faces F and vertices V is constant. This is essential for the following important theorem due to Euler. It is the precursor to Euler's polyhedron formula. It states that for any connected planar graph, the number of edges E, vertices V and faces F fulfil :

$$\boxed{V - E + F = 2} \tag{2.27}$$

Euler's polyhedron formula can be derived from it, because every three-dimensional convex polyhedron can be represented by a connected planar graph, and those are sometimes called *polyhedral graphs*.

For more general polyhedra, Euler's formula can be extended to

$$V - E + F = \chi \tag{2.28}$$

where χ is called the *Euler-characteristic*, that already appeared in 2.1. For our purposes, only the $\chi = 2$ case is relevant.

2.4.1 Topological Constraints

We can acquire some general statements about the possible configurations of clathrin lattices through their connection to planar graphs. This is especially easy for the possible configurations of the clathrin-cages, which correspond to connected 3-regular planar graphs. We can find the possible numbers of n-gons in such clathrin-cages.

Let F_i be the number of i-gonal faces included in a polyhedron and assume that it is possible to construct a polyhedron out of $\{F_i\,;\, i \geq 3\}$ such faces. Then every i-gon has i vertices, which are shared with two other faces, since the graph is trivalent. Therefore, the number of vertices V is:

$$V = \frac{1}{3}\sum_{i\geq 3} i \times F_i \tag{2.29}$$

Every i-gon also has i edges, which are shared with exactly one other face, so the number of edges E is:

$$E = \frac{1}{2}\sum_{i\geq 3} i \times F_i \tag{2.30}$$

which leads to the general statement

$$E = \frac{3}{2}V \tag{2.31}$$

The total number of faces is $F = \sum_{i\geq 3} F_i$. Substituting this into (2.28) leads to

$$\boxed{12 = \sum_{i\geq 3} \{6 - i\}\, F_i = 3F_3 + 2F_4 + F_5 + 0F_6 - F_7 - 2F_8 - \dots} \tag{2.32}$$

This leads to the often heard statement that a closed cage out of only hexagons and pentagons needs exactly **twelve pentagons**. Equation (2.32) is the more general statement. The number of pentagons can be increased by also including heptagons (often observed in clathrin-coats (Heuser 1980; Albert J Jin et al. 1993), or polygons of higher order.

A well known example of such a shape is the *soccer ball* or *truncated icosahedron* (TI), which has $F_6 = 20$ hexagons and $F_5 = 12$ pentagons. But also other, less regular shapes are not only possible but also found in clathrin experiments (Vigers et al. 1986).

2.5 Monte Carlo Algorithms

For simulating our clathrin model, we will employ several Monte Carlo techniques. We will therefore give a brief introduction into Monte Carlo techniques, Markov chains, Markov Chain Monte Carlo algorithms, the Metropolis Hastings algorithm and kinetic Monte Carlo techniques. Four easy-to-understand, yet comprehensive overviews for the following are given by Landau et al. (2015), Jansen (2012), Binder et al. (2010), and Hanada et al. (2022).

The basic idea comes from the Monte Carlo technique of estimating integrals using probability distributions. The following integral F

$$F = \int_\Omega \mathrm{d}x\, f(x)$$

can be evaluated by introducing a non-vanishing function $p(x)$

$$F = \int_\Omega \mathrm{d}x\, p(x)\frac{f(x)}{p(x)}$$

If $p(x)$ is a probability density function, i.e. non-negative (or in our case positive) with $\int_\Omega \mathrm{d}x\, p(x) = 1$, then the integral is the expectation value of $f(x)/p(x)$:

$$F = \left\langle \frac{f(x)}{p(x)} \right\rangle \tag{2.33}$$

for x-values sampled according to $p(x)$. If, e.g., $\Omega = [x_1, x_2]$, one might choose $p(x) = \frac{1}{x_2 - x_1}$ as a uniform distribution. If one then generates a set of random values $\{\chi\}$ of length N, the value of the integral can be approximated by

$$F \approx \frac{x_2 - x_1}{N} \sum_\chi f(\chi) \tag{2.34}$$

For computational efficiency, it becomes important how the probability distribution looks. If the integral is dominated by values of a region, then we achieve faster convergence by choosing a probability distribution peaked in that region. This is known under the keyword *importance sampling*, and will pop up again when we look at sampling configuration space. We should mention that it is a priori difficult to say which regions contribute how much to a certain integral if the integrands become reasonably complicated. This problem is elegantly solved using the Markov chain Monte Carlo method, where the sampling naturally spends most of the time at the states contributing most to the distribution.

2.5.1 Markov Chains

An important concept for our studies of ensemble-distributions are Markov chains, first introduced by Andrey Markov in 1906. For an overview, see Douc et al. (2018). A Markov chain is a process that describes the transition between states C_i out of a catalogue $\{C\} = C_1, C_2, \ldots, C_N$, i.e. the events form a *chain*. Starting from an initial event $C_{i(0)}$, the following events are $C_{i(0)} \to C_{i(1)} \to C_{i(2)} \to \ldots$. The probability for an event at every step depends only on the state of the system at the previous step. Colloquially it is often said that a Markov process has *no memory*. This can be a bit misleading and is important to clarify for later. The Markov process has a memory, as its current state will depend on its past, and therefore its future trajectory depends on its past. However, the precise way in which the current state is reached does not influence the future trajectory of the system, as all memory is stored in the current state. Speaking in terms of conditional probabilities, the Markov property reads:

$$P(C_1, C_2, \ldots, C_N | C_{N+1}) = P(C_N | C_{N+1}) \tag{2.35}$$

In the end, if our system can be described by states C_i at every step n, then there exists a probability distribution $P_n(C_i)$ of finding the system in state C_i at step n. If we know the transition probabilities $W_{ij} := P(C_i \to C_j)$, we can find the state probability distribution at the next step $n+1$ as

$$P_{n+1}(C_i) = P_n(C_i) + \sum_{j \neq i} \Big[P_n(C_j) W_{ji} - P_n(C_i) W_{ij} \Big] \tag{2.36}$$

as the difference of states transitioning into state C_i and transitioning out of state i. The upper equation is called the *master equation*. In a continuous Markov process, it describes

the time evolution of our probability distribution. By replacing the discrete step variable n with the continuous time variable t we get

$$\frac{\mathrm{d}P(C_i,t)}{\mathrm{d}t} = \sum_{j\neq i}\Big[P(C_j,t)W_{ji} - P(C_i,t)W_{ij}\Big] \tag{2.37}$$

The master equation implies conservation of the *total probability*:

$$\frac{\mathrm{d}}{\mathrm{d}t}\sum_{\alpha}P(\alpha,t) = \sum_{\alpha}\frac{\mathrm{d}P(\alpha,t)}{\mathrm{d}t} = \sum_{\alpha\beta}\Big[W_{\alpha\beta}P(\alpha,t) - W_{\beta\alpha}P(\beta,t)\Big] \equiv 0 \tag{2.38}$$

which follows by exchanging summation indices.

An important class of distributions are so-called *stationary* or *equilibrium distributions*. Such distributions do not change between steps, i.e. $P_n(C_i) = P_m(C_i)\ \forall n,m$. Many naturally occurring Markov chains (so called *regular* chains) will eventually approach such a distribution. While in theory there may be many different stationary distributions, any distribution that fulfils the so called *detailed balance condition* is certainly a stationary distribution. Detailed balance is achieved if and only if

$$\boxed{W_{ij}P(C_i) = W_{ji}P(C_j)} \quad \text{(detailed balance)} \tag{2.39}$$

and therefore, any probability distribution P together with its Markov process that fulfil detailed balance are automatically stationary.

In many contexts, the transition probabilities W_{ij} are split into a *choice* and an *acceptance* probability:

$$W_{ij} = W_{ij}^{\text{choice}}W_{ij}^{\text{accept}} \tag{2.40}$$

When using the Markov chain Monte Carlo methods, we will sample a probability distribution by hopping along the states of a Markov chain whose equilibrium distribution matches the probability distribution we want to sample. In theory, to ensure we sample the complete state space, we would need to have an ensemble of initial states from which we hop along the chain. We can avoid this by ensuring that our process is *ergodic*, i.e. every state C_i can be reached from every state C_j with a non-zero probability.

2.5.2 Markov Chain Monte Carlo

The main idea is that we can sample values out of a probability distribution $P(X)$ by using a Markov chain whose equilibrium distribution is that probability distribution, i.e. $P^{\text{Markov}}_{\text{Equation}}(X) = P(X)$. Then we can hop along the Markov chain and thereby sample our probability distribution $P(X)$ without explicitly knowing all our probabilities: It suffices to know the transition rates.

The most prominent example is the sampling of configurations of a canonical ensemble out of a Boltzmann distribution. The probabilities of being in configuration C_i is given as

$$P(C_i) = \frac{1}{Z}W(C_i)\ \ ,\ \ W(C_i) := e^{-E(C_i)/k_BT}\ \ ,\ \ Z := \sum_i W(C_i) \tag{2.41}$$

Finding the Z is cumbersome and oftentimes impossible. But in this case, the ratio of probabilities cancels the Z:

$$\frac{P(C_i)}{P(C_j)} = \frac{W(C_i)}{W(C_j)} \tag{2.42}$$

This is a significant simplification. One can now create a Markov process that is in detailed balance by ensuring that

$$\frac{P(C_i \to C_j)}{P(C_j \to C_i)} = \frac{W(C_j)}{W(C_i)} = \frac{P(C_j)}{P(C_i)} \tag{2.43}$$

meaning that the ratio of transition probabilities match the ratio of state probabilities. A very common choice is

$$P^{\text{accept}}(C_i \to C_j) = \min\left[\frac{W(C_j)}{W(C_i)}, 1\right] \tag{2.44}$$

with a suitable W_{ij}^{choice}. For a Boltzmann distribution, this simplifies to the calculation of energy differences:

$$P^{\text{accept}}(C_i \to C_j) = \min\left[\exp\Big(-\beta\Big(E(C_j) - E(C_i)\Big)\Big), 1\right] \tag{2.45}$$

When looking at spin lattices, one usually chooses W_{ij}^{choice} to be uniform if the two configurations differ by only one spin-flip and zero otherwise:

$$P(C_i \to C_j)^{\text{choice}} = \begin{cases} \text{uniform} & \text{one spin-flip difference} \\ 0 & \text{else} \end{cases} \tag{2.46}$$

Such a choice fulfils detailed balance, since in Equation (2.39) either both sides are zero or have the same constant factor. This choice of transition probabilities is called the *Metropolis algorithm*. It was introduced by Nicholas Metropolis in 1953 (Metropolis et al. 1953). The generalisation for asymmetric proposal probabilities is called *Metropolis Hastings Monte Carlo algorithm*, and later introduced by Wilfred Hastings, leading to the now-known name of *Metropolis Hastings*.

While we generate configurations out of the stationary distribution, two generated configurations are not necessarily statistically independent. To measure this one uses *autocorrelation functions*. For a quantity Q, the autocorrelation of Q is defined as

$$A_Q(\tau) := \frac{\langle Q(t)Q(t+\tau)\rangle_t - \langle Q(t)\rangle_t^2}{\langle Q(t)^2\rangle_t - \langle Q(t)\rangle_t^2} \tag{2.47}$$

The denominator simply normalizes the quantity to the standard deviation, and we subtract the squared expectation value of the quantity itself. The autocorrelation function asks how much realisations of Q at time $t+\tau$ depend on Q at t, and this is averaged over all time realisations. For an ergodic simulation, in many cases one expects statistical independence for $\tau \to \infty$ and a decay of the autocorrelation function as

$$A_Q(\tau) \sim e^{-\tau/\Theta} \tag{2.48}$$

with *autocorrelation time* Θ. However, this behaviour may differ around critical points.

2.5.3 Kinetic Monte Carlo for Markov Chains

While it is tempting to interpret the Markov chain as following the system in its time evolution, this notion is wrong! The Markov chain is only used as a tool to sample from the

equilibrium distribution, which by definition does not evolve in time and is not connected to a physical time scale. But there are simulation schemes similar to the Markov chain Monte Carlo method that allow us to simulate a physical time evolution, which equip a Markov process with a time evolution. Such methods are called *kinetic Monte Carlo* techniques, or sometimes under the name of *Gillespie algorithms*.

At first, the setup is similar to a standard MCMC setup: We again start with a set of states $\{C_i\}$ between which the system can change. But now instead of providing a transition probability $P(C_i \to C_j)$, we provide a *transition rate* k_{ij} that is linked to a physical time scale, as it has units of 1/time. Whether this catalogue of transition rates needs to be known a priori depends on the scheme.

We define the *total escape rate* r_i out of state i as

$$r_i := \sum_{j \neq i} k_{ij} \tag{2.49}$$

Then, the probability that the system is still in state C_i at time t ($X(t) = C_i$) if it was in state C_i at time $t = 0$ ($X(t = 0) = C_i$) is then given as

$$P^{C_i}_{\text{survive}}(t) := \Pr\{X(t) = C_i | X(t = 0) = C_i\} = \exp(-r_i t) \tag{2.50}$$

We now link this probability with the probability distribution of the *time of first escape*, $P^{C_i}_{\text{TFE}}(t)$, which is the time at which the first transition away from C_i takes place. The integral of $P^{C_i}_{\text{TFE}}$ from 0 to t is the probability that *any* escape has happened between $t = 0$ and t, and therefore must equal $1 - P^{C_i}_{\text{survive}}(t)$. Therefore

$$\int_0^t \mathrm{d}t'\, P^{C_i}_{\text{TFE}}(t') = 1 - P^{C_i}_{\text{survive}}(t) \Rightarrow P^{C_i}_{\text{TFE}}(t) = r_i \, \exp(-r_i t) \tag{2.51}$$

The first escape times along any path with escape rate k are distributed exponentially like

$$\boxed{P^{k}_{\text{TFE}}(t) = k \exp(-kt)} \tag{2.52}$$

with average first escape time τ being

$$\tau := \langle t \rangle = \frac{1}{k} \tag{2.53}$$

defining the time scale. This is true for any particular escape path as well as for the total escape process. From here, the kinetic Monte Carlo scheme is almost obvious. Assume we start in state C_i at time t with transition rates k_{ij} out of state C_i to states C_j. Proceed as follows:

- For every $j \neq i$, draw a time of first escape t_{ij} according to $P^{C_i}_{\text{TFE}}(t_{ij}) = k_{ij} \exp(-k_{ij} t_{ij})$.
- Only one escape can happen, which is the earliest, so choose the j with minimal t_{ij}.
- Evolve the time by $\Delta t = t_{ij}$.
- Repeat.

Such an algorithm is known as *rejection free* KMC (rfKMC). It is also known as BKL-KMC, named for its inventors, Bortz, Kalos and Lebowitz in 1975 (Bortz et al. 1975), n-fold way, Gillespie algorithm, for another advertiser of this method (Gillespie 1977), variable step size-KMC (VSSKMC), or often simply as KMC.

In practice, there are several simplifications. Usually one does not draw a time of first escape for every path, but simply samples a number uniformly between 0 and $\sum_{j\neq i} k_{ij}$ to choose a path, and then advances the system time by drawing a time of first escape using the total escape rate r_i.

We will combine the kinetic Monte Carlo approach with an open system in the sense that particles are allowed to enter (clathrin binds to the membrane) and leave (clathrin unbinds from the membrane) the system. The rate at which they bind to the system will be governed by an Arrhenius-like law

$$k \propto \gamma \exp(\beta\mu) \tag{2.54}$$

where μ is an effective polymerisation energy for binding of clathrin to the coat. This approach is therefore sometimes named *grand canonical Monte Carlo*.

2.5.4 Rejection KMC

The benefit of rejection free KMC is that every computation, i.e. every evolution-step actually *changes* the system and is not "wasted". The downside is that we need to know every transition-rate to every possible state at each computation-step. For some systems, e.g. ones with high symmetry (crystals, spin-lattices), this is feasible. For the system we are describing, it is not. The main reason is that we are looking at an *off-lattice* system. The possible state space trajectories are not limited by a lattice, which makes the computation of the whole transition rate catalogue an almost unfeasible task. Such off-lattice KMC (OL-KMC) simulations were first developed by Henkelman and Jónsson in 2001 (Henkelman et al. 2001) and use rejection based KMC, another variant of the kinetic Monte Carlo scheme. The idea here is that the system can also remain in its current state i. The possibility of simply not changing facilitates the decision process drastically. If we are in state C_i, and there are N possible pathways of changing state C_i to some other state, we can simply choose one of those ways uniformly at random, decide if the path is executed or not, and then progress the system by a time increment Δt whether or not the change actually took place. Therefore, we do not have to know every transition rate out of state C_i! We only need an upper limit on the sum of the transition rates to normalize everything. This, as well as the overall equivalence between rejection free and rejection based methods were proven by Serebrinsky in 2011 (Serebrinsky 2011). For a better understanding of how rejection-based and rejection-free KMC are connected, the proof together with some intuition is given in the appendix, see A.10.

Let us define a rejection KMC algorithm: Assume there is a set of states $\{C_i\}$ and a set of transition-rates $\{k_{ij}\}$ between these states. Then for every state C_i there exists a total escape rate $r_i := \sum_{j\neq i} k_{ij}$. In rfKMC, we need to know this total escape rate, as the state C_j to which the state C_i escapes is chosen with probability k_{ij}/r_i.

In rKMC, we can avoid having to know these r_i precisely by using a set of *overestimated escape rates* $\{w_i \geq r_i\}$.

The algorithm is then as follows:

1. We start in a system in state C_i at time t.
2. Choose a state $C_j \neq C_i$ out of the N_s states available to state C_i.
3. Advance the time by Δt drawn from $\text{Prob}(\Delta t) = \exp(-w_i N_s \Delta t)$ with mean $\langle \Delta t \rangle = \frac{1}{N_s w_i}$.

4. Accept change to state C_j with probability k_{ij}/w_i.

5. Repeat.

Importantly, the time is advanced whether or not we accept the transition. Also, independent of how good we choose our upper bound for the total escape rate, each step in an rKMC-scheme can advance the time only by $\langle \Delta t_{\text{rKMC}} \rangle \leq \langle \Delta t_{\text{rfKMC}} \rangle / N_s$. This is the trade-off for not having to compute the N_s escape rates k_{ij}.

The rKMC-scheme is often less efficient than the rfKMC-scheme. Even after N_s rKMC-steps, the average probability of having escaped the state i is

$$\text{Prob}_{\text{escape}}^{C_i} = N_s \left\langle \frac{k_{ij}}{w_i} \right\rangle_j = \frac{N_s}{w_i} \frac{\sum_{j \neq i} k_{ij}}{N_s} = \frac{r_i}{w_i} \leq 1 \tag{2.55}$$

where equality is only achieved for a perfectly chosen upper bound. So for every state C_i, if we overestimate the total escape rate by a lot, we will spend more than N_s computational steps in state C_i.

Also, N_s can be larger than necessary. If there exist states $C_{j'}$ with $k_{ij'} = 0$, then we will again spend unnecessary computational time in state C_i that the rfKMC scheme would not. The efficiency of our rKMC-scheme can be expressed as the average probability of leaving any state at any evolution step and is given by

$$\text{eff}_{\text{rKMC}} = \left\langle \frac{k_{ij}}{w_i} \right\rangle_{ij} = \frac{1}{N_s} \left\langle \frac{r_i}{w_i} \right\rangle_i \tag{2.56}$$

with 1 being the step efficiency of the rfKMC scheme.

2.6 Computational Techniques

Our software is written in `Python` using the library `JAX`, developed by Google. It is a library for numerical computing that can be used as a drop-in replacement for `numpy`, making its usage very familiar. Among others, it adds three powerful features: automatic differentiation for computing derivatives of complex (i.e. complicated, not imaginary) functions (called *autograd*), just-in-time compilation for speeding up consecutive executions of functions, and vectorisation to efficiently operate on entire arrays of grids. `JAX` was developed in the context of machine learning, and we will use some techniques often found in the context of machine learning (although not limited to that). I want to introduce the main ones briefly.

2.6.1 AutoGrad

AutoGrad, or *automatic differentiation*, is a neat feature of many machine learning libraries. It allows us to directly find the exact gradient of a function. It is one of the three common ways to calculate derivatives, alongside *finite difference derivatives* and *symbolic derivatives*. For a short and comprehensive review, see (Margossian 2019). In short, AutoGrad leverages the chain rule. Let us imagine we have a function $f : \mathbb{R}^n \to \mathbb{R}^m$ implemented into our software. The $m \times n$ Jacobian is $J_{ij} := \frac{\partial f_i}{\partial x_j}$. If f is no elementary function, it is composed out of other functions: $f(x) = (h \circ g)(x)$. Then we know that for the Jacobian, we have

$$J_f(x) = J_{h \circ g}(x) = J_h(g(x)) \cdot J_g(x) \tag{2.57}$$

More generally, if f is composed of L functions, $f = f^L \circ f^{L-1} \circ \ldots \circ f^1$, then the Jacobian is composed out of the L sub-Jacobians: $J = J_L \cdot J_{L-1} \cdot \ldots \cdot J_1$. Ultimately, any function defined within within our program is composed of elementary functions, for which the exact Jacobians are known.

As an example (taken from (Margossian 2019)), think of the log-likelihood function

$$f(y, \mu, \sigma) := \log\left(\text{Normal}(y, \mu, \sigma)\right) = -\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2 - \log(\sigma) - \frac{1}{2}\log\left(2\pi\right) \tag{2.58}$$

We can represent this function using the following sequence of operations:

$$\begin{aligned}(y, \mu, \sigma) &\to (y - \mu, \sigma) \\ &\to \left(\frac{y-\mu}{\sigma}, \sigma\right) \\ &\to \left(\frac{y-\mu}{\sigma}, \log(\sigma)\right) \\ &\to \ldots \\ &\to -\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2 - \log(\sigma) - \frac{1}{2}\log\left(2\pi\right)\end{aligned} \tag{2.59}$$

where every operation is elementary, like division, multiplication, or application of a logarithm. Such a mapping can be expressed through an *expression graph*, as can be seen in Figure 2.4. In this representation, it becomes clear how the decomposition works. As an example, to calculate the dependency of v_5 on μ, use the chain rule:

$$\frac{\partial v_5}{\partial \mu} = \frac{\partial v_5}{\partial v_4}\frac{\partial v_4}{\partial \mu} = \frac{1}{\sigma}\left(-1\right) \tag{2.60}$$

For more complicated functions, it can be advantageous to use so called *forward* or *reverse* mode, but for our purposes this is not necessary.

2.6.2 Gradient Descent

Gradient descent describes the simple idea of moving in a *cost*-landscape towards regions of lower cost. We have a *cost* or *loss*-function $L(\theta)$ for some parameters θ and try to minimize the loss-function. Since heavily used in machine learning (ML), most such libraries are well-equipped to perform gradient descent.

If posed with such a minimization problem, one usually defines a *learning rate* α and then proceeds, after choosing an initial set of parameters $\boldsymbol{\theta}_0$ and the number of optimization steps N, like this:

1. Calculate the gradient: $\boldsymbol{\nabla} L(\boldsymbol{\theta_i})$
2. Calculate $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \alpha \cdot \boldsymbol{\nabla} L(\boldsymbol{\theta}_i)$
3. If $i < N$, go to step 1.

This algorithm has a direct physical interpretation. If our loss function L is the Hamiltonian H of our system, the gradient w.r.t. parameters $\boldsymbol{\theta}$ describes a (generalised) *force* acting upon our system:

$$\boldsymbol{F} = -\boldsymbol{\nabla} H(\theta) \tag{2.61}$$

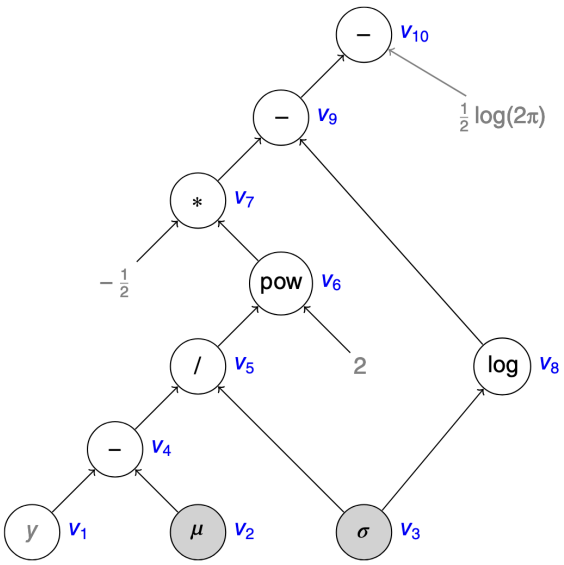

Figure 2.4: An expression graph for the log-likelihood function. The log-likelihood function can be expressed as an expression-graph which shows the order in which elementary operations, whose Jacobians are known, have to be applied in order to reach the final composed function. Each operation is indicated through v_1 to v_10. Figure taken from Margossian (2019).

Then, the algorithm described above is equivalent to integrating the equations of motion in the overdamped limit. As a reminder, in damped dynamics, the E.O.M. looks like

$$M\ddot{X} = -\nabla U(X) - \zeta\dot{X} \tag{2.62}$$

where the $\zeta\dot{X}$ term describes *friction*. In the limit of (almost) no acceleration, a solution to this equation is found as

$$\dot{X} = -\frac{1}{\zeta}\nabla U(X) \tag{2.63}$$

Thus, the updating algorithm above can be seen as a discrete integration of this acceleration-free equation of motion, where the learning rate α corresponds to $\Delta t/\zeta$.

$$\boldsymbol{\theta}_{T+\Delta t} = \boldsymbol{\theta}_T + \frac{\Delta t}{\zeta} \cdot \boldsymbol{F} \tag{2.64}$$

This explains how gradient descent works in the first place: It mimics a "ball" rolling down an energy landscape with friction, such that the ball eventually comes to rest in an energy basin. By decreasing the learning rate α, we *increase* friction, thereby increasing the chance of the ball coming to a stand in basins with lower and lower walls.

But this also allows us some physical interpretation of our system. For example, we can find the force acting upon the membrane that forces curvature in- or decrease, simply by evaluating the gradient of the energy E w.r.t. the curvature H. Or we can find the forces acting upon single particles in our grid through their elastic interactions with their neighbours. Though the resulting force is a high-dimensional vector, it can be used to find minimal-energy-configurations even in complicated grid configurations.

The upper explained method of gradient descent used the physical case of overdamped dynamics. However, faster convergence to a minimum can be achieved using *gradient*

descent with momentum, famously used in the *ADAM*-algorithm (Kingma et al. 2017). Here, the E.O.M. is the full (2.62). One then minimizes the cost function again by integrating the equations of motion, now starting with finding the velocity $\dot{X}_{i+1} = \beta\dot{X}_i - \alpha\Delta U$ and then the position $X_{i+1} = X_i + \dot{X}_{i+1}$. The introduction of β, the decay rate for the velocity, leads to a rolling average of the previous velocities, i.e. the system remembers the "directions" from which it came, but forgets them with each further iteration. ADAM extends this idea further by introducing a dynamic effective "mass" to the equations of motion.

2.6.3 The Envelope Theorem

The envelope theorem is a statement about the variation of an optimal-value-function. It states: Let $f(\boldsymbol{x}, q)$ be continouosly differentiable and let $\boldsymbol{x}^*(q)$ be the solution to

$$f(\boldsymbol{x}^*(q), q) = \max_{\boldsymbol{x}} \{f(\boldsymbol{x}, q)\} \tag{2.65}$$

Then $v(q) := f(\boldsymbol{x}^*(q), q)$ defines the optimal-value-function of f. The envelope theorem states:

$$\boxed{\frac{\mathrm{d}v(q)}{\mathrm{d}q} = \left.\frac{\partial f(\boldsymbol{x}, q)}{\partial q}\right|_{\boldsymbol{x}=\boldsymbol{x}^*(q)}} \tag{2.66}$$

i.e. the variation with respect to the optimised parameter $\boldsymbol{x}$ vanishes. To show this, note that

$$f(\boldsymbol{x}^*(q), q) = \max_{\boldsymbol{x}} \{f(\boldsymbol{x}, q)\} \;\Rightarrow\; \boldsymbol{\nabla}_{\boldsymbol{x}} f(\boldsymbol{x}, q)|_{\boldsymbol{x}=\boldsymbol{x}^*(q)} = 0 \tag{2.67}$$

and therefore

$$\begin{aligned}
\frac{\mathrm{d}v(q)}{\mathrm{d}q} &= \left.\frac{\mathrm{d}}{\mathrm{d}q}\left(f(\boldsymbol{x}, q)\right)\right|_{\boldsymbol{x}=\boldsymbol{x}^*(q)} \\
&= \left.\frac{\partial f(\boldsymbol{x}, q)}{\partial q}\right|_{\boldsymbol{x}=\boldsymbol{x}^*(q)} + \frac{\partial \boldsymbol{x}^*(q)}{\partial q} \cdot \underbrace{\boldsymbol{\nabla}_{\boldsymbol{x}} f(\boldsymbol{x}, q)|_{\boldsymbol{x}=\boldsymbol{x}^*(q)}}_{=0} \\
&= \left.\frac{\partial f(\boldsymbol{x}, q)}{\partial q}\right|_{\boldsymbol{x}=\boldsymbol{x}^*(q)}
\end{aligned} \tag{2.68}$$

Of course, we do not get anything for free: We are trading the computation of the partial derivatives $\boldsymbol{\nabla}_{\boldsymbol{x}}$ against the optimisation of f w.r.t. $\boldsymbol{x}$, which in practice usually involves the calculation of gradients.

Where this can shine is in the calculation of, e.g., forces. We will later encounter an energy function $E(\boldsymbol{x}, H)$ that depends on a (high dimensional) vector of positions $\boldsymbol{x}$ and a curvature H. When changing this curvature, it is not directly clear what the corresponding energy should be. We can however define the energy at a curvature H as the minimal energy at fixed curvature for energy minimizing $\boldsymbol{x}^*(H)$, thereby defining $E(H) := E(\boldsymbol{x}^*(H), H)$ (alternatively, one can exchange H for other auxiliary parameters).

Then, using implicit differentiation, we find

$$\begin{aligned}\left(\frac{\mathrm{d}E}{\mathrm{d}H}\right)(H) &:= \left(\frac{\mathrm{d}E}{\mathrm{d}H}\right)(\boldsymbol{x}^*(H), H) \\ &= (\boldsymbol{\nabla}_{\boldsymbol{x}} E)(\boldsymbol{x}^*(H), H) \cdot \left(\frac{\mathrm{d}\boldsymbol{x}^*}{\mathrm{d}H}\right)(H) + \left(\frac{\partial E}{\partial H}\right)(\boldsymbol{x}^*(H), H) \\ &= \left(\frac{\partial E(\boldsymbol{x}, H)}{\partial H}\right)(\boldsymbol{x}^*(H), H)\end{aligned} \tag{2.69}$$

which can be used to find the effective bending rigidity of a system.

If instead of differentiating w.r.t. H we differentiate w.r.t. bending radius R, we can find the effective pressure of the system. With

$$-AP = \frac{\mathrm{d}E}{\mathrm{d}R} = \frac{\mathrm{d}E}{\mathrm{d}H}\frac{\mathrm{d}H}{\mathrm{d}R} = -\frac{1}{R^2} 2\kappa A H \Rightarrow P = 2\kappa H^3 \quad \text{and} \quad P = \frac{-1}{A}\frac{\mathrm{d}E}{\mathrm{d}R} \tag{2.70}$$

By defining the energy per surface area $\omega := \frac{E_{\text{total}}}{A_{\text{surface}}}$ we get

$$\frac{\mathrm{d}\omega}{\mathrm{d}R} = \frac{1}{A_{\text{surface}}}\frac{\mathrm{d}E_{\text{total}}}{\mathrm{d}R} = -P(R, A) \tag{2.71}$$

Chapter 3

An Agentic Model of Clathrin Assembly in Spherical Geometry

This chapter outlines the modelling and simulation of clathrin coat growth on curved surfaces. The necessary concepts are introduced step-by-step, building upon the theory established in Section 2.

3.1 General Idea

The aim is to provide a framework to model the accumulation of clathrin onto a curved membrane and to observe the dynamics of this growth at variable curvature. By doing so, we want to address questions such as the role of pentagons in coat formation and effective parameters.

To start, we model clathrin as an object (a *hub* or *node*, often displayed as a point) with a position in 3D space. The positions are restricted to a sphere with radius R. Nodes have two states, either *active*, which means accumulated on the sphere, or *inactive*, which means not accumulated on the sphere and therefore taking no part in any interactions. Inactive nodes have no effect on the system's dynamics. The clathrin is therefore modelled as being supplied by a particle bath, similar to a grand canonical ensemble, where particle uptake is regulated through polymerisation energies or chemical potentials, and particle supply is never exhausted.

Any hub can form so-called *direct bonds* to other hubs (displayed through directed lines), which are then called the hub's *partners*. These bonds correspond to the proximal part of the clathrin legs, see Figure 1.4. Because of clathrin's geometry, every hub can form three direct bonds, i.e., every hub can have up to three direct partners.

For every partner of a hub, a *second-order* bond can be formed through the partner to a partner of the partner. Such a bond is called a *next-nearest-neighbour bond* (or short NNN-bond) and the partner of the partner is called the *next-nearest neighbour* of the hub. This NNN-bond corresponds to the distant part of a clathrin leg, again see Figure 1.4. Figure 3.1 depicts this configuration for a node with the maximum number of direct and NNN-bonds built. The naming conventions and the geometry of our model are explained in the next section.

We then equip every node with an energy function and distribute multiple nodes on a sphere to let them interact. We also equip the whole grid configuration with an energy function, such that the combination assigns an energy depending on the individual arrangements of nodes and bonds as well as on the configuration of the whole grid. With

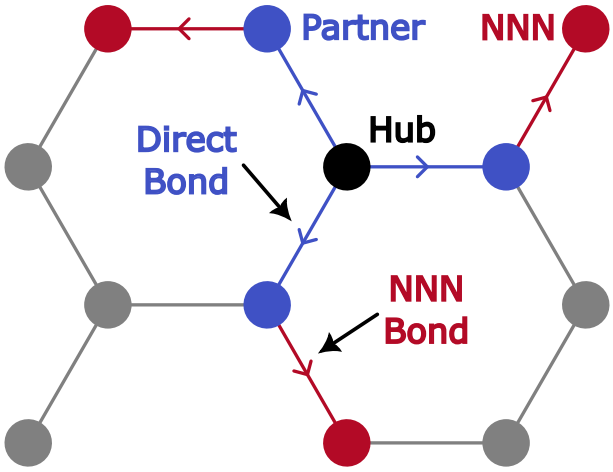

Figure 3.1: Diagrammatic structure inside a clathrin grid. Every hub has up to three partners to which direct bonds are formed (blue), and for every partner a next-nearest neighbour can be connected through an NNN-bond (red). The coloured legs all belong to the central hub (black). This reproduces the ability of clathrin to build densely connected grids with beyond-nearest-neighbour interactions.

this, we can simulate equilibrium and non-equilibrium state transitions, using either standard Markov chain Monte Carlo methods or kinetic Monte Carlo methods.

Our idea is that the system can dynamically form or remove bonds between nodes, place or remove nodes from the membrane or change the membrane curvature, to effectively lower its energy. The energy function dictates in large part how a final grid will look. For example, the typical hexagonal structure of clathrin is encoded in the energy function of the angles.

The sphere radius, being an easily accessible parameter for the curvature $H = 1/R$, can be made arbitrarily large to provide a flat environment or can be lowered to increase the curvature. On this curved surface, the hubs are then allowed to form connections to one another. The model is designed such that if node A has formed a direct bond to node B, then node B automatically forms a direct bond to node A, i.e., to be a direct partner of another node is reciprocal. So two nodes that are connected automatically have at least two bonds connecting them. Then, if node B has another partner C, then C could build an NNN-bond to A through B. Similarly, if node A has another partner D, then D could form an NNN-bond to B along A. So at most two nodes are connected by four bonds. Figure 3.2 shows an example of a perfectly connected part of a grid. Bond colours match the hub-colour of the node from which they originate. Direct bonds are drawn outward from the central line connecting two nodes, NNN-bonds are drawn more central on that line. Arrows indicate the direction away from the node from which the bond originates.

The nodes can dynamically form and remove these bonds, which constitutes a non-equilibrium state change, which is simulated using a kinetic Monte Carlo algorithm. We then also allow the single nodes to move along the sphere. The underlying assumption is that the movement of the single nodes and the bond formation and removal happen on different timescales, i.e., the coat adiabatically rearranges itself around an equilibrium configuration through node movements, while bond formations or removals constitute an exit from this equilibrium state and are therefore modelled using the kinetic Monte Carlo method. This also provides a connection to physical timescales.

It is important to emphasize that we can also dynamically add new nodes to the

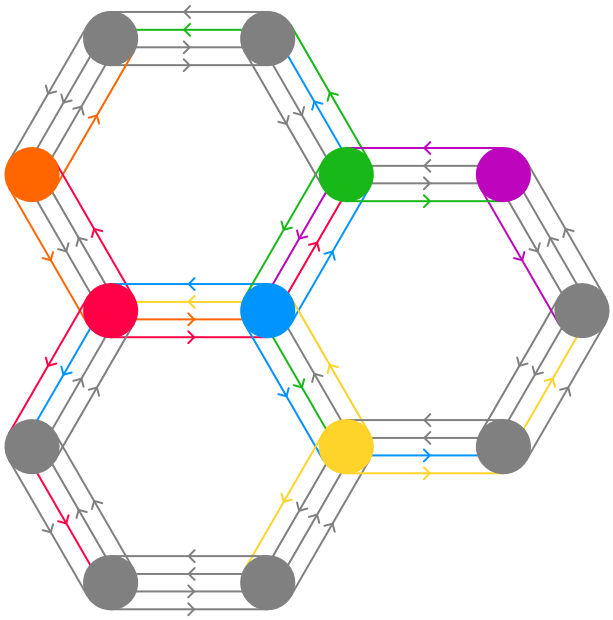

Figure 3.2: Depiction of bonds between nodes. In a typical depiction of bonds between nodes, direct bonds are drawn on the outside of the central line connecting two nodes, NNN-bonds are drawn more central. Arrows indicate the direction away from the node from which the legs originate.

system. And these nodes are *not* placed at *predetermined positions*, as in many other simulations using either an underlying grid (Frey, Bucher, et al. 2020) or fixed clathrin shapes (Den Otter et al. 2011), but rather can be placed continuously across a range of positions determined by an underlying probability function that depends on the energy of the node at the position.

3.2 Geometry & Notation

Before detailing the algorithm, it is necessary to define the geometric relations and nomenclature used. We will generally assume that the nodes are placed on a sphere of radius R that is centred in 3D space at $(0, 0, 0)$. Consider a single node where the i-th leg has length L_i. We will not indicate the overall orientation of the node, since the system is rotationally invariant. However, the pair of legs i and j enclose the angle θ_{ij}. Note that this angle is defined in 3D space, and is therefore called the *total angle*. In general, except for when $R = \infty$ (i.e., $H = 0$), the three total angles of the proximal legs will not add up to 2π. This can be seen for real clathrin in Table 1.1.

Every node at (x, y, z) defines a tangential plane touching the sphere, defined by the normal vector $\boldsymbol{n}$ of the plane, which is the normalised position vector $\hat{\boldsymbol{x}}$ of the node.

We define $\boldsymbol{V}_i$ to be the vector representing the i-th leg of the node, i.e., $\boldsymbol{V}_i := \boldsymbol{y} - \boldsymbol{x}$, where $\boldsymbol{y}$ denotes the position of the partner. In the tangential plane, the projection of $\boldsymbol{V}_i$ is the planar projected vector

$$\boldsymbol{V}_i^{\parallel} := \boldsymbol{V}_i - (\boldsymbol{n} \cdot \boldsymbol{V}_i)\, \boldsymbol{n} \tag{3.1}$$

For every angle θ_{ij}, there exists a so-called *projected angle* or *planar angle* ϕ_{ij} defined to be the angle enclosed by the planar projected vectors of the legs i and j. Furthermore,

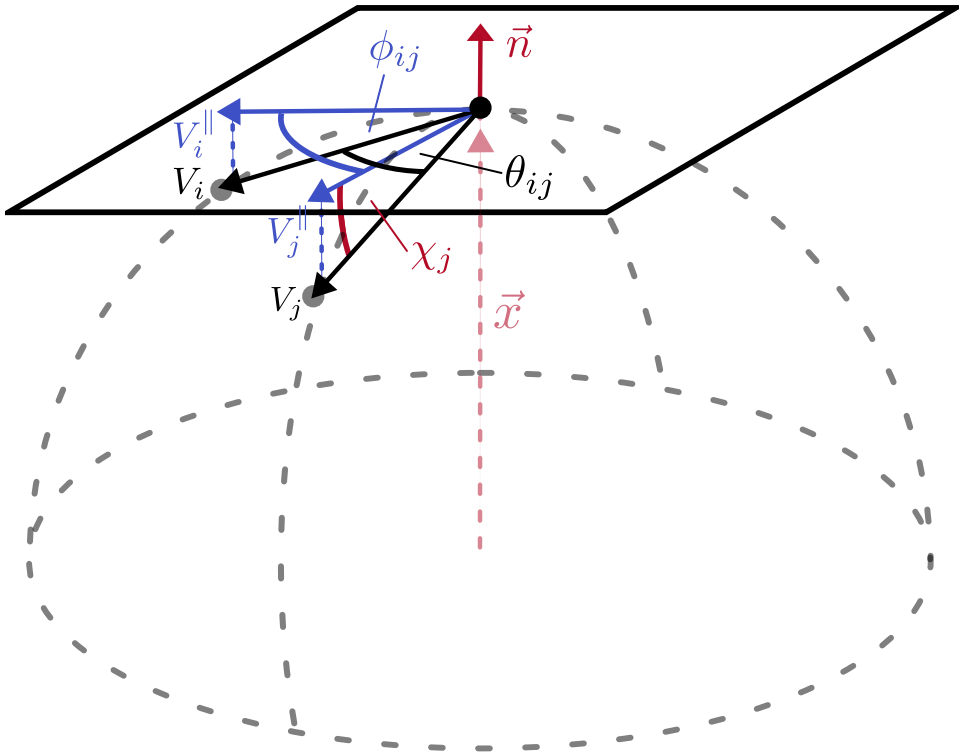

Figure 3.3: Explanation of the single node geometry. A single node is placed at position $\boldsymbol{x}$ with radius R. The normalised node position $\boldsymbol{n}$ defines its tangential plane. The vector connecting to its partner i-th is called $\boldsymbol{V}_i$. Its projection, living in the tangential plane, is called $\boldsymbol{V}_i^{\parallel}$. The connecting vectors enclose the total angle θ_{ij}, the projected vectors enclose the projected angle ϕ_{ij}. The angle between real and projected vector is the pucker angle χ_j.

every leg is associated with a so-called *dipping* or *pucker angle* (Den Otter et al. 2011) χ_i, defined to be the angle enclosed by $\boldsymbol{V}_i$ and $\boldsymbol{V}_i^{\parallel}$. Figure 3.3 illustrates these variables using the example of a typical clathrin configuration. The angles are therefore defined to be

$$\begin{aligned}\cos(\theta_{ij}) :=& \frac{\boldsymbol{V}_i \cdot \boldsymbol{V}_j}{\|\boldsymbol{V}_i\|\|\boldsymbol{V}_j\|} \\ \cos(\phi_{ij}) :=& \frac{\boldsymbol{V}_i^{\parallel} \cdot \boldsymbol{V}_j^{\parallel}}{\left\|\boldsymbol{V}_i^{\parallel}\right\|\left\|\boldsymbol{V}_j^{\parallel}\right\|} \\ \cos(\chi_i) :=& \frac{\boldsymbol{V}_i \cdot \boldsymbol{V}_i^{\parallel}}{\|\boldsymbol{V}_i\|\left\|\boldsymbol{V}_i^{\parallel}\right\|}\end{aligned} \tag{3.2}$$

If we abbreviate $d_i := \|\boldsymbol{V}_i\|$, we can rewrite

$$\cos(\chi_i) = \frac{\boldsymbol{V}_i \cdot (\boldsymbol{V}_i - (\boldsymbol{n} \cdot \boldsymbol{V}_i)\,\boldsymbol{n})}{d_i\sqrt{d_i^2 - (\boldsymbol{n} \cdot \boldsymbol{V}_i)^2}} = \sqrt{1 - \frac{(\boldsymbol{n} \cdot \boldsymbol{V}_i)^2}{d_i^2}} \;\Rightarrow\; \sin(\chi_i) = \frac{|(\boldsymbol{n} \cdot \boldsymbol{V}_i)|}{d_i}$$

from which we get

$$\cos(\phi_{ij}) = \frac{\boldsymbol{V}_i \cdot \boldsymbol{V}_j - (\boldsymbol{n} \cdot V_i)(\boldsymbol{n} \cdot \boldsymbol{V}_j)}{\sqrt{d_i^2 - (\boldsymbol{n} \cdot \boldsymbol{V}_i)^2}\sqrt{d_j^2 - (\boldsymbol{n} \cdot \boldsymbol{V}_j)^2}} = \frac{\frac{\boldsymbol{V}_i \cdot \boldsymbol{V}_j}{d_i \cdot d_j} - \sin(\chi_i)\sin(\chi_j)}{\cos(\chi_i)\cos(\chi_j)}$$

By plugging in the definition of $\cos(\theta_{ij})$, we finally find the relation between these three angles:

$$\boxed{\cos(\phi_{ij}) = \frac{\cos(\theta_{ij}) - \sin(\chi_i)\sin(\chi_j)}{\cos(\chi_i)\cos(\chi_j)}} \tag{3.3}$$

We find that for every set of $\{\phi_{ij}, \theta_{ij}, \chi_i, \chi_j\}$, only three angles are independent.

It will be interesting to look at the impact and meaning of these different angles. We think the total angles θ_{ij} should be the ones that define the energy, since they are the most physical ones. Therefore, they control both curvature and lattice geometry. However, splitting the angles up into the projection angle and the dipping angle has several advantages. The projection angle obviously is related to the hexagonal structure of the clathrin lattices, i.e., we expect ϕ_{ij} to be 120° in an ideal configuration (in the flat plane). On the other hand, the dipping angle has a direct interpretation in terms of curvature, shown in Equation (2.8), as

$$\boxed{\chi_i \approx \frac{L_i H}{2}} \tag{3.4}$$

The typical curvature evolution starts almost flat at $H_{\text{init.}} \approx 0$ and increases towards the spontaneous curvature H_0.

3.3 Grid States, State Changes and Time Evolution

An essential part of our model is the question of how the state evolution should take place. While it is hard to find time-resolved data for CME, and often other surrogates for time are used as evolution parameter, it is still important to incorporate growth in the correct framework.

The evolution process we want to observe is not happening in an equilibrium. The whole grid growth is heavily energy-driven and node additions as well as conformal changes strongly influence the energy landscape. We therefore believe a standard Metropolis Hastings Monte Carlo algorithm to be unsuitable for resolving the evolution we are interested in completely. While there are some state changes in our model for which an MH-MCalgorithm is suitable, which will be explained below, the evolution process is mainly driven by coat growth, and this happens through node addition, a process not suitable to be described through MH-MC.

Our solution will be to use a kinetic Monte Carlo algorithm (see 2.5.3) to find the suitable time evolution. To define our methodology more precisely, we will in the following sections define what constitutes a state of our system, what processes change these states and how these state changes will be implemented and why.

3.3.1 States of our System

A state is a set of parameters that let us recreate the complete grid state. We will now give these parameters and already separate them into two subsets, which will become important later on.

We define a **macrostate** α of our system[1] as a set of *nodes* $\{n_\alpha\}$ that are bound to the membrane together with a set of *direct bonds* between two nodes $\{(n_\alpha, m_\alpha)\}$

[1] Always noted using Greek letters.

and a set of *NNN-bonds* along three nodes $\{(n_\alpha, m_\alpha, p_\alpha)\}$. The bond notation means (Base-node, Partner-node) and (Base-node, Partner-node, NNN-node). To complete the necessary information needed to recreate the full grid state, we define a **microstate** i_α of macrostate α[2] by the set of positions $\{(x_n, y_n, z_n)\}$ of the nodes $\{n_\alpha\}$ together with the current curvature H_{i_α}. The combined information of macrostate α and microstate i_α holds the complete information of our current grid state. The idea of differentiating between these two sets of information is based on how they change. A macrostate change will be mediated through a kinetic process, i.e., a process for which we can define a physical rate, like the formation of bonds between two clathrins. A microstate will change simply by sampling from the thermal distribution.

3.3.2 Changes of State

We can now list all changes the system can make to change its complete state to another one. To change its current state, the grid can:

- Move a node n_α to change its position (x_n, y_n, z_n).
- Change the membrane curvature
- Form a direct bond between nodes n_α and m_α or remove a direct bond.
- Form an NNN-bond between nodes n_α and p_α along m_α or remove such a bond.
- Add a new node m_α to the set of active nodes $\{n_\alpha\}$.

Therefore, only the first two processes change the microstate $i \to j$, while all other processes change the macrostate. We suggest that the movement of nodes and changes in curvature happen on a much faster timescale than any other evolution, i.e., the node positions $\{(x_n, y_n, z_n)\}$ are always close to a local energetic minimum. For the curvature evolution this means a necessary inclusion of a flat membrane potential that dominates the coat bending rigidity in the beginning. In other words, the microstate adiabatically follows a local energetic minimum. And therefore the sampling of the microstate happens in equilibrium, which means we can use a classic MH-MC scheme for microstate sampling.

On the other hand, all other processes change the macrostate $\alpha \to \beta$, i.e., are related to changes in in the grid configuration. This is by definition an out-of-equilibrium process which drives the time evolution and is therefore necessarily connected to a timescale. This is best suited to be simulated using a KMC scheme.

3.4 Implementing Curvature Changes

An essential part of our software is the capability to change the curvature of the sphere on which the clathrin coat grows. To achieve this, we need to define a map

$$C : S(R = R_1) \to S(R = R_2),\ \boldsymbol{x}_i \mapsto \boldsymbol{x}'_i \tag{3.5}$$

mapping the positions $\boldsymbol{x}_i$ from the sphere with radius R_1 (i.e., $\|\boldsymbol{x}_i\| = R_1$) to positions $\boldsymbol{x}'_i$ on the sphere with radius R_2 (i.e., $\|\boldsymbol{x}'_i\| = R_2$). There are, of course, numerous ways to define such a map. The simplest approach would be to **rescale** every node position to the

[2] Always abbreviated using Roman letters.

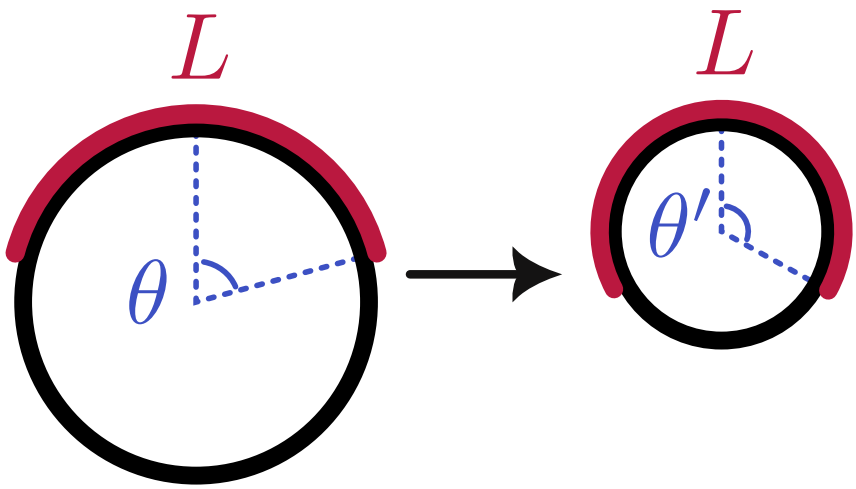

Figure 3.4: Wrapping method explained. For a piece of cloth wrapping a cylinder, decreasing the bending radius, i.e., increasing the curvature while keeping the length of the cloth constant, the wrapping angle θ increases to $\theta' < \theta$.

new radius, i.e., $\boldsymbol{x}_i' = \frac{R_2}{R_1}\boldsymbol{x}_i$. However, this would effectively amount to a simple rescaling. After such a change, every bond would be frustrated, as the average length would have been reduced. Of course it is interesting how a coat would relax in such a situation, but we believe that this might not be the most natural way to define a curvature increase. Indeed, it can be shown that such a mapping corresponds to a rescaling of the equilibrium length L_0 of the legs, and nothing else.

We propose that it is better to try to emulate a *wrapping* behaviour, shown in Figure 3.4. Think about a piece of cloth that is wrapped around a cylinder. We can think of the process like this: Imagine the cloth piece being flat, i.e., lying on a cylinder with radius $R = \infty$ or curvature $H = 0$. We can increase the curvature slightly step by step, until we have reached the target radius. During the process, the distance of each line along the symmetry axis of the cylinder stays the same along the slope of the cylinder.

In cylindrical coordinates θ, z at fixed $r = r_1$, we can define a *reference line* at $\theta_0 = 0$. Let us assume we look at the line at θ_1. Its distance to the line at θ_0 is $d(r_1) = \theta_1 r_1$. If we now increase the curvature by decreasing the radius to $r_2 < r_1$, to keep the distance fixed, we need to increase $\theta_2 > \theta_1$, i.e.

$$\theta_1 r_1 = d(r_1) \stackrel{!}{=} d(r_2) = \theta_2 r_2 \;\Rightarrow\; \theta_2 = \theta_1 \frac{r_1}{r_2} > \theta_1 \tag{3.6}$$

The lines "shift along the cylinder", i.e., increase their θ-coordinate to keep their distances in θ-direction fixed.

We want to apply the same idea to a plane wrapping around a sphere. But as is commonly known, such a mapping cannot preserve all lengths and angles: To wrap a piece of paper around a ball, we need to crumple it. Speaking in spherical coordinates R, θ, φ, maintaining the distance in θ-direction while changing R and keeping φ **fixed** will decrease the distance in φ-direction (the "crumpling").

The corresponding mapping is defined as follows: For a set of coordinates R_1, θ_1, φ_1, find the corresponding coordinates at decreased radius R_2 as

$$(R_1, \theta_1, \varphi_1) \mapsto \left(R_2, \theta_2 = \frac{R_1}{R_2}\theta_1, \varphi_2\right) \tag{3.7}$$

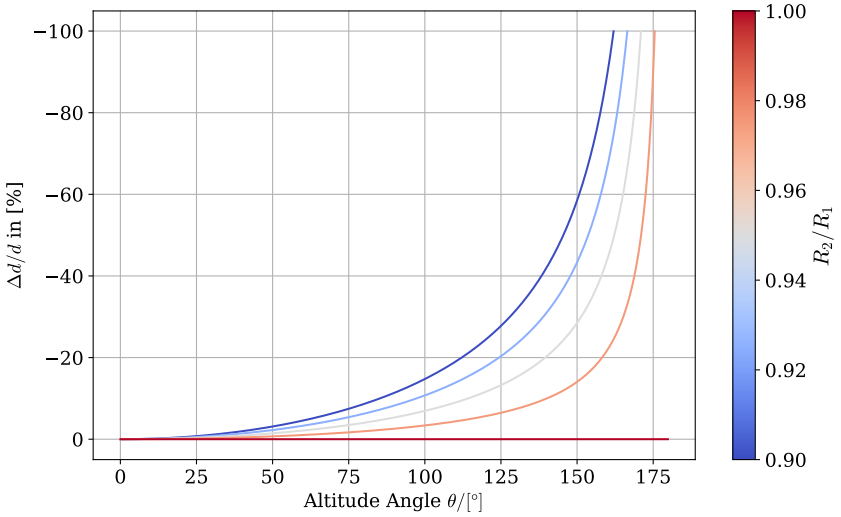

Figure 3.5: Azimuthal distance increase versus altitudinal position on the sphere. The relative azimuthal distance change according to (3.8) is shown versus the altitudinal position of the two nodes on the sphere. Larger curvature changes lead to larger decreases in the distance. The change in distance is always negative. A maximal distance change of 100% is reached when $\pi = \theta_2 = \theta_1 R_1/R_2$.

Consider two points at the same altitude θ_1 with azimuthal angles φ_1 and φ_2. They have an azimuthal distance $d(R_1) = \sin(\theta_1)R_1\Delta\varphi$. After changing the radius, $\Delta\varphi$ remains constant and we find the relative change in distance to be

$$\frac{\Delta d}{d(R_1)} = \frac{\sin\left(\frac{R_1}{R_2}\theta_1\right)}{\sin(\theta_1)}\frac{R_2}{R_1} - 1 \tag{3.8}$$

We see the relative azimuthal distance changes in Figure 3.5. The closer we are to the south pole, i.e., the larger the altitudinal angle θ, the more the azimuthal distance decreases for an increase in curvature. A maximal distance decrease is reached when the two positions would meet at the south pole, i.e., when $\theta_2 = R_1\theta_1/R_2 \overset{!}{=} \pi$. If this point is reached, the paper-crumple approach to curvature increases fails. However, this is quite physical, as at that point the sphere would have to close, as the circumference has shrunk to zero.

3.5 Energy Function

An important aspect of our model is that the macroscopic effects should arise from a microscopic energy function. Ideally, this is formulated on the level of individual nodes and their legs. The idea is that this energy function assigns each grid configuration a dedicated energy, which in turn is then used to evolve the grid state to a new one, which is explained in Section 3.3 on grid state changes.

While we have non-local effects like the exclusion energy and the membrane bending energy, we will first focus on the local node energy.

3.5.1 Node Energy

The energy is summed up from different parts. One energy contribution comes from the legs of the individual clathrin hubs. Their configuration can be optimal or suboptimal, and thus define their energy. The leg configuration of a single clathrin hub can be defined using the length L_i of every i-th leg, the pucker angle χ_i of the i-th leg as well as the projection angle ϕ_{ij} between the i-th and j-th leg, as explained in Section 3.2. For NNN-bonds, the definition of the projection angle has to be modified to be the angle between the NNN-bond and the corresponding direct bond, but in general the position of every bond can be described using the three parameters mentioned above..

Therefore we propose an energy $E(\{L_i, \chi_i, \phi_{ij}\})$ of a node that depends on the set of parameters $\{L_i, \chi_i, \phi_{ij}\}$ where each parameter is restricted by a harmonic potential:

$$E_{\text{leg}}(\{L_i, \chi_i, \phi_{ij}\}) := \sum_i \Bigg\{ k_L \left(\frac{L_i}{L_0} - 1 \right)^2 + \left[k_\chi \left(\chi_i - \chi_0 \right)^2 - E_\chi^{\text{shift}} \right] + k_\phi \left(\phi_{ij} - \phi_0 \right)^2 \Bigg\} \tag{3.9}$$

where L_0, χ_0 and ϕ_0 define *equilibrium parameters* that have to be chosen according to the situation of interest. The energy shift E_χ^{shift} allows us to shift the pucker angle energy. This will be useful when we allow the curvature to change, such that we start with equal initial conditions, more on that in Section 3.9. The spring constants κ_X have units of energy, given in $k_B T$, or energy per *radian* for the two angular constants. With regards to other units, we usually set $L_0 \equiv 1$ and measure lengths in units of the equilibrium length. The definition of the dipping angle χ does not change whether we are talking about a direct bond or an NNN-bond. For the projection angle, this is different.

The reason becomes clear once we talk about what to do with "unplaced" bonds. While physical clathrin of course always has three legs, we will not actively simulate all three of them. Rather, we want to assume that if a leg is not bound, it will fluctuate around its energetic minimum, and therefore its energy can be neglected in some approximation. While that means that the length of an unplaced leg can always be chosen to be optimal, for the angular energy this is not directly given.

Let us first think about the ϕ energy. For the hexagonal structure we usually choose $\phi_0 = 120°$. Then if only one leg is placed, the other two can always arrange themselves such that every leg-pair encloses an angle of $\phi = \phi_0$, and therefore the ϕ energy can be chosen to be zero for only one placed leg. For two placed legs with enclosed angle ϕ_{ij}, the minimal energy configuration for the system is such that the third leg positions itself at an angle of $(2\pi - \phi_{ij})/2$ with respect to the other two legs. For a configuration with two placed legs, we will therefore include the energy contribution of the unplaced leg in the above mentioned form. For three placed legs, the energy will simply be given by the above-mentioned formula with a sum over all three leg-pairs.

For the χ energy, we currently do not include the energy for unbound legs, i.e., we only sum over existing legs.

3.5.2 Understanding the Node Energy

It is worth thinking about the meaning of each term in the single node energy. First of all, is such an energy term suitable to describe our system, and is it necessary? As explained in 3.3, the microstate of our grid is described by the set of angles $\{\chi_i, \phi_{ij}\}$, the set of

leg lengths $\{L_i\}$ and the curvature $H = 1/R$ of our system. We know from Equation (3.4) that at fixed curvature, the set of lengths $\{L_i\}$ is equivalent to the set of pucker angles $\{\chi_i\}$. Therefore, at *fixed* curvature, it would suffice to have either the stretching energy of the pucker angle energy. But precisely for changing curvatures, it is necessary to include *both* contributions into the energy such that our system can sense the curvature. It should also be noted that through Equation (3.3) a description in terms of $\{\chi_i, \phi_{ij}\}$ is equivalent to any description replacing either $\{\chi_i\}$ or $\{\phi_{ij}\}$ with $\{\theta_{ij}\}$. We can fill the terms in Equation (3.9) with some more intuition.

1. The first two terms are responsible for the scale of our grid and for the curvature generation. While the first term is much more responsible for the scaling of the system than the second, still both contribute. Using some appropriate approximations, we will show that the second term is responsible for curvature generation.

2. The third term describes an energy that has more to do with the overall configuration of the grid. It is not easily possible for the grid to minimise this energy in a straightforward way. This can be already seen from the definition of ϕ_0, which is entirely determined through the "base geometry" of the lattice, i.e., the hexagonal structure as the usually assumed configuration. We call this part of the energy the conformational energy.

To illuminate the first statement, let us look at the first two terms of (3.9). We can use the small angle approximation together with Equation (3.4) to replace $\chi_i = L_i H/2$. Then we find

$$E_{L+\chi} = k_L \left(\frac{L_i}{L_0} - 1\right)^2 + k_\chi \left(\frac{L_i H}{2} - \frac{L_0 H_0}{2}\right)^2 - E_\chi^{\text{shift}} \tag{3.10}$$

The dual impact of this energy can be seen by looking at the cases of fixed or changeable curvature. Let us first assume we look at a change of macrostate, i.e., we allow the curvature to change. In this case, it is reasonable to assume that the length will be on average around the equilibrium length, i.e., $L_i \approx L_0$. Then the energy per leg then simplifies to (up to the constant shift)

$$E_{\text{Curvature change}}(H) = \frac{k_\chi L_0^2}{4}(H - H_0)^2 = \frac{k_\chi L_0^2}{4R_0^2}\left(\frac{R - R_0}{R}\right)^2 \tag{3.11}$$

which is the usual Helfrich type energy described in Section (2.1). The factor $L_0^2/4$ is exactly the matching factor we calculated in Equation (2.7). The course of the energy as a function of H or R can be seen in Figure 3.6. The energy shift E_χ^{shift} can be used to set the energy at the initial curvature to zero, i.e., $E(H = H_{\text{init}}) = 0$, more on that in Section 3.9.

Since the Helfrich energy is defined by an integral, we can associate L_0^2 as the rough area occupied by a single clathrin, which corresponds to the infinitesimal area element $\mathrm{d}A$ in the integral, again making the identification of k_χ with the bending rigidity of the Helfrich energy more natural.

As mentioned above, the scale of the lattice (i.e., roughly the average distance between nodes) is mostly controlled through the stretching energy. However, also the pucker angle energy contributes. To see this more clearly, we can look at (3.10) (up to the constant

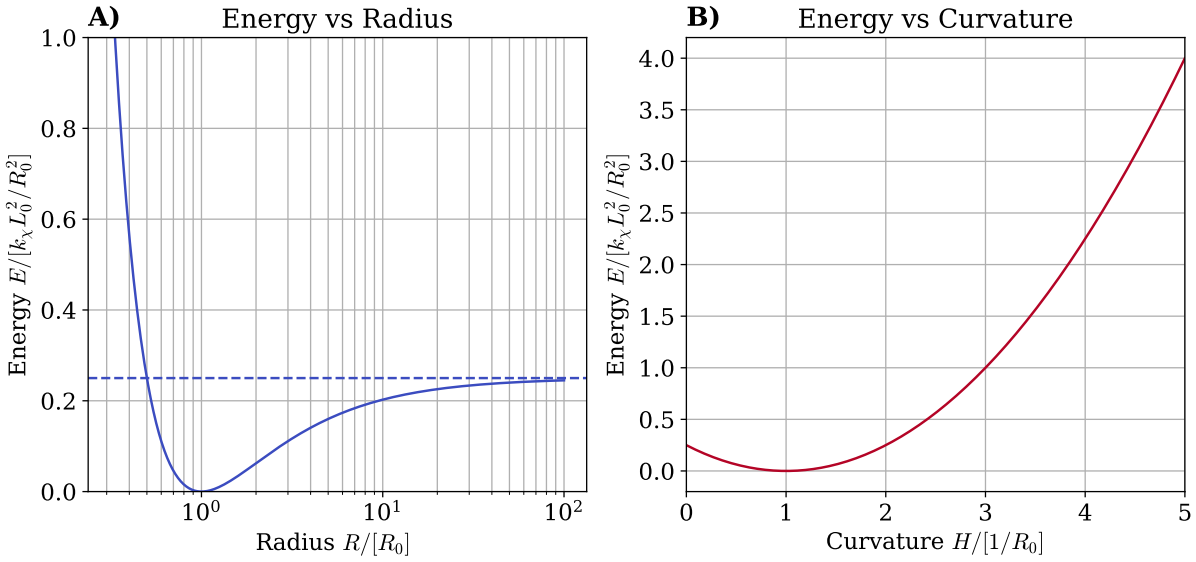

Figure 3.6: Energy during radius changes. A) The energy as a function of bending radius R. We see the minimum at $R = R_0$. The dotted line shows the asymptotic energy for a flat coat at $E_{\text{asympt.}} = \frac{\kappa_\chi L_0^2}{4R_0^2}$. B) The energy against the curvature H shows the typical parabolic Helfrich shape.

shift) at fixed curvature H:

$$\begin{aligned} E_{\text{Scale}}(L) =& k_L \left(\frac{L}{L_0} - 1\right)^2 + \frac{k_\chi H^2}{4}\left(L - \frac{L_0 H_0}{H}\right)^2 \\ =& k_L \left(1 + \eta \frac{H^2}{H_0^2}\right)\left(\frac{L}{L_0} - \left\{\frac{1 + \eta\frac{H}{H_0}}{1 + \eta\frac{H^2}{H_0^2}}\right\}\right)^2 + \text{const.} \end{aligned} \tag{3.12}$$

where we have defined $\eta := \frac{H_0^2 L_0^2}{4}\frac{k_\chi}{k_L}$, which reveals that the combination of pucker angle energy and stretching energy shifts the equilibrium leg length away from L_0, as well as modifying the stretching stiffness. The shift of the equilibrium leg length L_0 can be seen in Figure 3.7. In anticipation to Section 3.9, the order of magnitude of η for the cases we use is at $\eta \sim 10^{-2}$.

This shift in equilibrium length might come as a surprise at first. So how to understand it? The equilibrium length is the length of a leg at which two forces, the stretching force that wants to keep $L = L_0$, and the angular force of the pucker angle, which wants to send L to much larger values than L_0 to increase the pucker angle[3], equilibrate.

For very small curvature, one would need to stretch the leg very much to increase the pucker angle by very little, i.e., the force resulting from the pucker angle energy is small, i.e., the equilibrium distance is close to L_0. If the curvature gets closer to H_0, then the energy gained from stretching the leg away from L_0 becomes larger, i.e., the force grows and the equilibrium length shifts away from L_0. At some curvature, almost every leg length around L_0 ensures already an almost correct pucker angle, and therefore the force again shrinks and the equilibrium length shifts back towards L_0.

[3]If we assume we approach the correct curvature from below, i.e., start at small H and approach H_0 from $H < H_0$, then the pucker angle can always be increased if we send the partner node further away on the sphere.

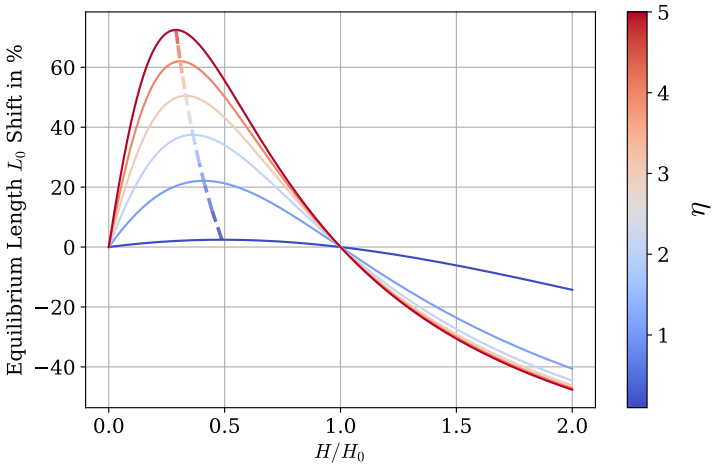

Figure 3.7: Shift of equilibrium leg length L_0. The equilibrium leg length L_0 shifts, depending on the curvature H and the relation of k_χ to k_L, measured in η. The maximum of this shift occurs at $H/H_0 = \sqrt{(\eta+1)/\eta^2} - 1/\eta$ at the value $(\sqrt{\eta+1}+1)/2$, indicated by the dashed line.

3.5.3 Exclusion Energy

We want to model every clathrin hub with an exclusion zone. We experimented with having a smooth *Lennard-Jones*-like potential. For reasons of computational efficiency, we have decided to opt for a *hard-shell potential*, meaning if two nodes cannot get closer than some σ, i.e.

$$E_{\text{excl.}}(r) = \lim_{U\to\infty} U\, H(\sigma - r) \tag{3.13}$$

with the Heaviside step function.

3.5.4 Membrane Bending Energy

We also want to include the possibility of some force pushing back against the curvature formation, and this force should ideally result from the analogue of the membrane pushing back against the clathrin. For this we want to employ the Helfrich bending energy with the form of Section 2.1.1. The simplest case is that the bent area is approximately the area occupied by clathrin. Then we can identify the area occupied by clathrin with the number of clathrin hubs polymerised on the membrane times an average clathrin area A. Therefore, we can approximate the integral for the Helfrich energy with

$$E_{\text{Helfrich, Membrane}} = \int \mathrm{d}A\, \kappa_{\text{Membrane}} \left(H - \overline{H}\right)^2 \approx \sum_{\text{Clathrin}} A_{\text{Cl}} \kappa_M \frac{1}{R^2} \tag{3.14}$$

where we have used that the spontaneous curvature $\overline{H}$ of the membrane vanishes and have identified the curvature H with the inverse sphere radius R. The sum goes over the polymerised clathrin, and A_{Clathrin} is the average area occupied by a single clathrin.

However, other forms of the bending area are possible. One could define the *smallest coat-engulfing spherical cap* as the spherical cap with smallest invagination angle θ at the given curvature H that still includes all clathrin. This has the form (compare to Section

2.3):

$$\begin{aligned} E_{\text{H, Membrane, Alt.}} =& \underbrace{\frac{2\pi}{H^2}\left(1-\cos\left[\max\left\{\theta_{\text{Clathrin}}\right\}\right]\right)}_{=A}\kappa_M H^2 \\ =& 2\pi\kappa_M\left(1-\cos\left[\max\left\{\theta_{\text{Clathrin}}\right\}\right]\right) \end{aligned} \tag{3.15}$$

It might look like this expression does not depend on the curvature. However, the curvature goes into calculating θ_{Clathrin}. We would expect that such an energy would lead to more symmetric and spherical-patch-like growth.

3.6 Effective Membrane Rigidity

We expect that the system of the clathrin coat can (at least to some degree) be described using an *effective bending rigidity* κ_C. By this we mean that the macroscopic energy can be described as a function of H:

$$E(H) = \kappa_C\left(H - H_{\text{min}}\right)^2 A \tag{3.16}$$

where A is the area the coat has grown to. From Equation (3.11) we can assume that if no other effects were to play a role this macroscopic bending rigidity *could* be expressed as

$$\kappa_C \approx \frac{L_0^2}{A_{\text{Cl}}}\frac{\langle N_{\text{leg}}\rangle}{4}k_\chi \tag{3.17}$$

since the bending energy *per leg* is described using $\kappa_\chi/4$, and we can assume that L_0^2 sets the area scale for clathrin.[4] N_{leg} is the number of legs per node, i.e., to how many neighbours and next-nearest neighbours the node has connected. This number is bounded by $1 \leq N_{\text{leg}} \leq 6$. We assume that the increase of the average number of bonds per node will increase the effective bending rigidity of the coat.

In Section 2.1.3 we defined the *bending rigidity ratio* $\alpha := \frac{\kappa_C}{\kappa_C+\kappa_M}$, where κ_M is the membrane's underlying bending rigidity. In a simple *two membrane model*, this ratio controls where the energy minimising curvature H_{min} of the combined membrane is at, with $H_{\text{min}} = \alpha H_0$, with H_0 being the energy minimising curvature of the coat alone, see Section 2.1.3. By substituting in Equation (3.17), we find α as a function of the average number of legs per clathrin $N_{\text{avg.}}$ and the ratio of dipping angle stiffness κ_χ to membrane bending stiffness κ_M:

$$\alpha = \left[1 + \frac{4}{\langle N_{\text{leg}}\rangle}\frac{\kappa_M}{\kappa_\chi}\right]^{-1} \tag{3.18}$$

which can be seen in Figure 3.8. We can see that the ratio of κ_M/κ_χ defines the scale on which α varies, and the increase of average leg number per clathrin increases α. The largest increase can be seen for $\kappa_M \approx \kappa_\chi$ (corresponding to the physical case), with $\Delta\alpha \approx 0.4$.

For a growing coat, we expect the average number of legs to increase as the coat becomes better connected and the ratio of nodes at its border relative to the total node number decreases. This would result in an increased value of α, which shifts the energy minimising curvature H_{min} towards H_0, the spontaneous curvature of clathrin. If the

[4] To be precise, when calculating the area occupied by a single clathrin one arrives at approximately $1.25L_0^2$, but we neglect this small difference for the time being.

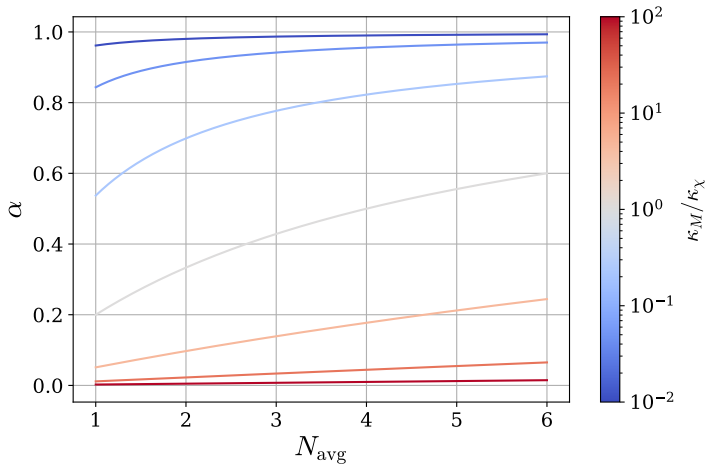

Figure 3.8: Effective bending rigidity α against stiffness ratio. The effective bending rigidity α is shown against the realisable average bond numbers per node for different ratios of the membrane bending rigidity κ_M against the pucker angle harmonic parameter κ_χ. From Section 2.1, κ_χ should relate to the clathrin coat's bending rigidity.

membrane's bending rigidity κ_M is larger than the coat's initial bending rigidity κ_χ, then an increase of $\alpha \to 0.5$ will increase the effective surface tension $\sigma = H_0^2 \kappa_{\text{tot}} \alpha(1-\alpha)$. For initial κ_χ larger than κ_M, an increase of α will be towards values larger than 0.5, therefore decreasing the effective surface tension σ. Looking at Table 1.1 and connecting the flexural rigidity to a membrane stiffness, see Appendix 2.2, we see that we expect an initial ratio of $\kappa_M/\kappa_C \gtrsim 1$.

3.7 Microstate Changes

As explained in 3.3, we want to equilibrate and sample a microstate i using a Metropolis Hastings Monte Carlo scheme (explained in Section 2.5). For this, given a certain microstate i_α, we need to choose a new microstate j_α. Let us first discuss how node changes contribute to this new state j_α. We select a node n_α from the macrostate at random and change its position (x_n, y_n, z_n) by some small displacement $(\Delta x_n, \Delta y_n, \Delta z_n)$, which then defines the new microstate j_α. For both microstates, we can compute the energy using the energy function from Section 3.5 to define the energy difference $\Delta E_{ij} := E_j - E_i$ between both states. Then the proposed state change from i_α to j_α is accepted with probability

$$\text{Prob}_{\text{accept}} = \min\left(1,\, \exp\left(-\beta \Delta E_{ij}\right)\right) \tag{3.19}$$

where the choice of the minimum ensures a faster convergence towards thermal equilibrium.

If the macrostate α consists of $N_\alpha = \#\{n_\alpha\}$ active nodes, then N_α such proposed state changes constitute a *single Monte Carlo step*. To claim that the system has equilibrated and the final microstate sampled is indeed drawn from an equilibrium distribution, we have to perform a suitable number of MC steps. This number depends on the temperature and the distribution of positional changes $\boldsymbol{\Delta x}_n$ proposed. For more details, see A.5.

For the curvature change, we decided to let the curvature evolve using again a Metropolis algorithm. After N_α state changes for the node positions, i.e., after one Monte Carlo step, we sample a curvature change $\Delta H \in [-\Delta H_{\max}, \Delta H_{\max}]$ and calculate the energy

change ΔE resulting from that curvature change. The proposed curvature change is then accepted using the Metropolis criterion:

$$\text{Prob}_{\text{accept}} = \min\left(1, \exp\left(-\beta \Delta E(\Delta H)\right)\right) \tag{3.20}$$

However, that this yields the correct curvature evolution is not clear a priori. As mentioned before, the state evolution in a Metropolis algorithm is not directly connected to the time evolution. However, we assume that the system's energy as a function of curvature H will be in a local minimum, and what we want to observe is the shift of that minimum, which is seen as the average curvature evolution. As long as the proposed curvature changes are small enough that the system will most likely not leave its energy basin during the relaxation steps, we expect the curvature to follow a random walk constrained by a potential well. If the energy basin moves because of the time evolution of our system, and the curvature changes are large enough for the system to re-equilibrate around the new minimum, then the Metropolis evolution should be able to capture the real-time dynamics of the curvature evolution.

3.8 Macrostate Changes

A central part of our model is the implementation of the KMC algorithm, as this is the way in which we simulate the changes of macrostate, the main driver of our time evolution. We had to decide between a rejection-free scheme (where *all* transition rates from one state to others must be known) and a rejection-based scheme. We determined that the latter was more appropriate.

While rejection-free schemes are usually computationally more efficient, since each computational step surely evolves the system to another state, improving sampling statistics, they come with the burden of creating a catalogue of transition rates from every state to every other state. This is often facilitated if the systems under consideration have large restrictions, e.g., through symmetries, such as in lattice systems (for an overview, see (Landau et al. 2015)). For example, in a 2D square spin lattice with nearest-neighbour interactions, every spin can only belong to 5 classes, being defined by having 0, 1, 2, 3 or 4 neighbouring spins being anti-aligned with the central spin. Therefore, the catalogue of possible transition rates is highly restricted.

But for other systems, finding this catalogue in a finite amount of time can be hard or even impossible. Usually, those systems are described as *off-lattice* systems (for reference, see (Ruzayqat et al. 2018)) and for those, a rejection-based KMC algorithm can sometimes be the only reasonable choice.

3.8.1 Defining the Event Catalogue

Even though we do not have to find the catalogue of all possible transition rates, we still have to find the catalogue of all possible events that change a macrostate. We differentiate kinds of events based on how we will calculate the transition rate for that event. In the catalogue of macrostate changing events, we define five different kinds of events:

1. Direct bond placement to a new node
2. Direct bond placement to an existing node
3. Direct bond removal

4. Next-nearest-neighbour bond placement to an existing node

5. Next-nearest-neighbour bond removal

We will sometimes refer to the first process as the *continuous process* and to the others as the *discrete processes*. For every macrostate α, we can define a catalogue of which events are possible. This catalogue is a set C containing five subsets, c_1 to c_5. An element $n \in c_i$ signifies that the node n can perform process i.[5] The number of escape processes N_α^{proc} from state α is then defined to be

$$\boxed{N_\alpha^{\text{proc}} := \sum_{i=1}^{5} \#c_i} \tag{3.21}$$

which is used to define the time evolution.

3.8.2 Defining Transition Rates

To define the time evolution, we need to define how we calculate the transition rate $k_{\alpha\beta}$ between two macrostates α and β. As we will see, we face the challenge that the estimation of this rate for the *continuous process*, the bond placement to a new node, will be quite different and harder than the one for the *discrete processes*.

Let us therefore start with the definition of the transition rates between states α and β related through one of the processes $2, 3, 4$ or 5. These processes are discrete in the sense that for every process, a clear energy change $\Delta E_{\alpha\beta}$ can be defined by simply taking the difference between the total energy of states α and β. We therefore take the simple ansatz of an *Arrhenius-like* law to define the transition rate as

$$\boxed{k_{\text{discrete}}^{\alpha\beta} := \gamma \ \exp\left(-\beta\, \Delta E_{\alpha\beta}\right)} \tag{3.22}$$

with the introduction of a *base rate* γ with units of 1/time. In general, this base rate could depend on the kind of process we are looking at, so instead of γ we would have four different γ_i. But for simplicity we will assume that all γ_i are in fact equal.

3.8.3 Estimating Rates, Node Addition

The more complicated process is the node addition, also called the *continuous process*. Here, the bond is formed to a node that is not yet part of the grid, i.e., its position is undetermined and we cannot directly assign an energy to this process. We assume we are not diffusion-limited, i.e., whenever a bond *wants* to form, it can do so. So the rate with which this process will happen will be proportional to the amount of space this new node has to settle down.

Let us imagine that there exists only a small piece of area ΔA_i onto which a new node could settle down. The rate with which the new node settles down will be proportional to the energy gain the system experiences, i.e., $\propto \exp\left(-\beta E_{\text{acc.}}(x_i)\right)$, where $E_{\text{acc.}}(x_i)$ is the accumulation energy that gets released when a node settles down at x_i.

[5] The elements are defined in a way in which it is clear with which partner node n performs process i.

Therefore, for many of such pieces of area ΔA_i, the total rate of node accumulation is

$$\begin{aligned} k_{\text{cont.}} &= \sum_i k_i = \gamma_{\text{add.}} \sum_i \Delta A_i \, \exp\left(-\beta E_{\text{acc.}}(x_i)\right) \\ &\rightarrow \boxed{k_{\text{cont.}} = \gamma_{\text{add.}} \int \mathrm{d}A \, \exp\left(-\beta E_{\text{acc.}}(x)\right)} \end{aligned} \tag{3.23}$$

where again $\gamma_{\text{add.}}$ is a base rate. However, $\gamma_{\text{add.}}$ is not the base rate γ from the discrete processes! This can already be seen from its units, which are one over an area per unit time. For the calculation of the addition rate $k_{\text{cont.}}$, the used accumulation energy is too complicated. It will in general be of the following form:

$$E_{\text{acc.}}(x) = -\Delta E_{\text{pol,hub}} - \Delta E_{\text{pol,leg}} + E_{\text{leg}}(L, \vartheta, \chi) + E_{\text{excl.}}(x) \tag{3.24}$$

The first two terms are the polymerisation energies which get released when a hub is built into the grid or respectively a leg binds to another leg. The third contribution is the leg energy. We assume that the leg is constrained by a harmonic potential both in its preferred length L and in its preferred angular orientation ϕ and χ. This energy has the form (compare Equation (3.9))

$$E_{\text{leg}}(L, \phi, \chi) = k_L \left(\frac{L}{L_0} - 1\right)^2 + \left[k_\chi (\chi - \chi_0)^2 - E_\chi^{\text{shift}}\right] + k_\phi \left(\phi - \phi_0\right)^2 \tag{3.25}$$

where k_ϕ, k_L and k_χ are the three spring constants. The last term, the exclusion energy, is the one making this integral hard to evaluate. It is by definition an exclusion potential which can be chosen differently, either as a (modified) Lennard-Jones potential or as a hard cut-off potential. For many computations in the Monte Carlo scheme, a hard-shell potential is easier, as we do not have to deal with microscopic acceptance rates, as they are simply set to zero. However, for optimisation techniques like the ones explained in Section 2.6.2, it is important that the gradient of the energy never vanishes completely. For the code at hand we therefore chose a hard-shell potential whenever possible, and for optimisation of the energy, we employed a differentiable potential that mimics the hard-shell.

Evaluating the integral (3.23) is hard since

1. The neighbourhood configuration for every node is different.
2. We are not dealing with simple Gaussians, but rather shifted Gaussians with finite borders and multiplied linear terms.

However, we can employ a trick, which not only remedies this problem. Because a second problem of the process of integrating a new node into the code is to actually choose the position of this new node if the process is accepted. We will now propose a solution to this problem as well as the problem of calculating the addition rate raised at the end of the previous section.

It is clear that the higher the rate for a position x, the higher the probability of choosing this position should be. Therefore we propose to choose a position x by sampling a random position out of the probability distribution (with the correct normalization)

$$\text{Prob}_{\text{full}}(x) = \frac{1}{Z_{\text{full}}} \exp\left(-\beta E_{\text{acc.}}(x)\right) \tag{3.26}$$

that is the integrand of the rate integral (3.23). But again, we face a similar problem. The probability distribution effectively chooses a position out of the plane by probability (3.26) except for the places where other nodes occupy space. To sample from such a distribution is computationally costly. Therefore, we propose the following solution:

We define the analytic part of the accumulation energy as

$$E_{\text{ana.}}(x) := E_{\text{acc.}}(x) - E_{\text{excl.}}(x) \tag{3.27}$$

and the **base rate** $k_{\text{cont.}}^{\text{base}}$ as

$$\boxed{k_{\text{cont.}}^{\text{base}} := \gamma_{\text{add.}} \int \mathrm{d}A \, \exp\left(-\beta E_{\text{ana.}}(x)\right)} \tag{3.28}$$

If a bond formation event to a new node is chosen as the process, we will preliminarily accept it with probability

$$\text{Prob}_{\text{acc},1} = \frac{k_{\text{cont.}}^{\text{base}}}{w} \tag{3.29}$$

where w is the upper limit on the rates. Whether or not accepted or rejected, we evolve the time normally using w. If rejected, we simply leave the process. If accepted, we proceed to the second step. There, we choose a random position out of the 2D plane according to

$$\text{Prob}_{\text{ana.}}(x) = \frac{1}{Z_{\text{ana.}}} \exp\left(-\beta E_{\text{ana.}}(x)\right) \tag{3.30}$$

which is much easier to sample. For details on the sampling, see the Appendix A.1. Then, in a third step, we check if the chosen position x lies within the exclusion zone of any already existing node. The probability of lying outside of such an exclusion zone is

$$\text{Prob}_{\text{outside}} = \frac{\int \mathrm{d}A \, \exp\left(-\beta E_{\text{acc.}}(x)\right)}{\int \mathrm{d}A \, \exp\left(-\beta E_{\text{ana.}}(x)\right)} = \frac{k_{\text{cont.}}}{k_{\text{cont.}}^{\text{base}}} \tag{3.31}$$

compare to eqs. (3.28, 3.23). In another acceptance step, we only accept the proposed process if the chosen position lies outside of any exclusion zone. Then, the probability of accepting the complete process is

$$\boxed{\text{Prob}_{\text{acc,tot.}} = \frac{k_{\text{cont.}}^{\text{base}}}{w} \frac{k_{\text{cont.}}}{k_{\text{cont.}}^{\text{base}}} = \frac{k_{\text{cont.}}}{w}} \tag{3.32}$$

which is the true probability according to our kinetic Monte Carlo scheme. With this method, we avoid estimating the integral completely, and only choose a random position at which to position the new node if the process is accepted in the first place. Since $k_{\text{add.}}/w \approx k_{\text{add.}}^{\text{base}}/w \ll 1$, the bottleneck is the first acceptance criterion, since only very few of these proposed processes will be accepted in the first place. Therefore, it is very advantageous to avoid having to calculate random positions or integrals until this bottleneck is passed in the first place. We will later explain how we can optimise the simulation to limit the effect of said bottleneck.

3.8.4 Estimating Rate Limits

In order for our algorithm to work, we need upper bounds w for our rates for the different processes. This can be achieved quite systematically. Let us first look at the discrete

Process	Rate	Maximal Rate
DBPEN	$\gamma \exp\left(-\beta \Delta E\right)$	$\gamma \exp\left(2\beta[E_{\text{pol,leg}} + E_{\chi}^{\text{shift}}]\right)$
NNNBP	$\gamma \exp\left(-\beta \Delta E\right)$	$\gamma \exp\left(\beta[E_{\text{pol,leg}} + E_{\chi}^{\text{shift}}]\right)$
DBREM	$\gamma \exp\left(-\beta \Delta E\right)$	undetermined
NNNREM	$\gamma \exp\left(-\beta \Delta E\right)$	undetermined

Table 3.1: Rate estimates for different processes. For some of the different discrete processes within our system, we can estimate the maximal rate at which they happen. For others, while a true maximum does not exist, we can still find an effective maximum by looking at reasonable parameter regions.

transitions. As a reminder, they are *direct bond placement to an existing node* (DBPEN), *next-nearest neighbour bond placement* (NNNBP), *direct bond removal* (DBREM) and *next-nearest neighbour bond removal* (NNNREM). The rates for these processes are given in Table 3.1.

The upper bound for the rate with which bonds are placed is easily found. The energy change ΔE is given by the release of polymerisation energy $E_{\text{pol,leg}}$, either once or twice, depending on whether we place a direct or an NNN-bond, and the energy contribution the leg gives. If it is optimally placed (which is only possible at the correct curvature), then the leg energy contributes E_{χ}^{shift} per leg. Therefore, the maximum rate is achieved by only considering the polymerisation energy plus the pucker angle energy shift.

The removal processes on the other hand do not have a strict upper bound, because bonds can be arbitrarily suboptimally placed, which would increase the bond energy E_{bond} to arbitrary amounts. However, it is still possible for us to place a reasonable upper bound on these rates. For this, we have to remember two things: First of all, a bond has to be placed first, and it is increasingly unlikely that a bond is placed with $E_{\text{bond}} \gg E_{\text{pol,leg}} + E_{\chi}^{\text{shift}}$. Second of all, after a bond is placed, we assume that the coat reshuffles to adiabatically follow a (local) energy minimum. While it is still possible that the system finds itself in a very constrained situation where the bond energy is far larger than the polymerisation energy, we believe it is reasonable to assume that this will not be the case for a relevant number of events.[6] We therefore suggest that a reasonable upper bound for the removal rate is again found at

$$k^{\text{rem}} \leq \gamma \exp\left(2\beta[E_{\text{pol,leg}} + E_{\chi}^{\text{shift}}]\right) \tag{3.33}$$

To double-check that the assumptions we made here are actually valid for the process in question, we will usually output the acceptance rates (i.e., k/w) for every process in question. As mentioned before, we assumed that the rate constant γ is equal for the different processes. We would therefore propose an upper bound for the rate of discrete processes of

$$\boxed{k_{\text{discrete}}^{\text{max}} = \gamma \exp\left(2\beta\left[E_{\text{pol,leg}} + E_{\chi}^{\text{shift}}\right]\right)} \tag{3.34}$$

Now we are tasked with finding an upper bound for the rate of bond placement to a new node (DBPNN), i.e., the continuous process. An upper bound for this is given by

[6] As explained in Section 3.8.7, we will anyway take into account a small fraction of events that are under-represented since they are accepted with probability $> 100\%$. We monitor this number and accept it for small values.

the base rate from Equation (3.28):

$$\boxed{k_{\text{cont.}} \leq k_{\text{cont.}}^{\text{base}} \equiv k_{\text{cont.}}^{\max}} \tag{3.35}$$

which therefore serves the double role as also the maximal rate for the node addition process. We can better express the base rate in the following way:

$$\begin{aligned} k_{\text{cont.}}^{\text{base}} =& \gamma_{\text{add.}} \int \mathrm{d}A \exp\left(-\beta E_{\text{ana.}}(x)\right) \\ =& \gamma_{\text{add.}} \exp\left(\beta\left[E_{\text{pol,hub}} + 2[E_{\text{pol,leg}} + E_{\chi}^{\text{shift}}]\right]\right) \\ & \int \mathrm{d}A \, \exp\left(-\beta[E_{\text{leg}}(L,\phi,\chi) + E_{\chi}^{\text{shift}}]\right) \end{aligned} \tag{3.36}$$

where the leg energy has the form of Equation (3.25), and we added and subtracted an E_{χ}^{shift} in the exponent under the integral. Since in such a process the curvature is constant, the pucker angle χ is a function of L, i.e., $\chi \equiv \chi(L)$, see Equation (3.4). Therefore, this integral can be solved by integrating it over the plane. We define its solution as the function $A_{\text{eff.}}(\phi, \chi_0, L_0, k_\phi, k_\chi, k_L)$ called the **effective area**:

$$\boxed{A_{\text{eff.}}(\phi, \chi_0, L_0, k_\phi, k_\chi, k_L) := \int \mathrm{d}A \, \exp\Big(-\beta\Big[E_{\text{leg}}\Big(L, \phi, \chi(L)\Big) + E_{\chi}^{\text{shift}}\Big]\Big)} \tag{3.37}$$

As the name suggests, this is the effective area size onto which a clathrin can bind next to an already present clathrin hub. With this, we find the expression for the base rate

$$k_{\text{cont.}}^{\text{base}} = \gamma_{\text{add.}} e^{\beta(E_{\text{pol,hub}}, +2[E_{\text{pol,leg}} + E_{\chi}^{\text{shift}}])} A_{\text{eff.}} \tag{3.38}$$

Since $A_{\text{eff.}}$ has units of area, we know that $\gamma_{\text{add.}}$ is a rate constant divided by an area. It therefore seems reasonable to define

$$\gamma_{\text{add.}} := \gamma c_0 \tag{3.39}$$

where γ is the rate constant seen in the discrete processes and c_0 can be thought of as an area density.[7] This again gives us another expression for the base rate[8]:

$$\begin{aligned} k_{\text{add.}}^{\text{base}} =& \underbrace{\gamma \exp\left(2\beta\Big[E_{\text{pol,leg}} + E_{\chi}^{\text{shift}}\Big]\right)}_{k_{\text{discrete}}^{\max}} e^{\beta E_{\text{pol,hub}}} A_{\text{eff.}}(L_0, k_L, k_\phi, \beta) c_0 \\ =& k_{\text{discrete}}^{\max} \exp\left(\beta E_{\text{pol,hub}}\right) A_{\text{eff.}}\, c_0 \end{aligned} \tag{3.40}$$

which shows that it is proportional to the maximal discrete rate. It seems reasonable to assume that $A_{\text{eff.}} c_0 \lesssim 1$, meaning that not more than one or two clathrins are usually available per binding area. However, this does not limit any further reasoning and is only a guide on how to tune the parameter $A_{\text{eff.}} c_0$. The general upper bound w on the rates is found by taking the maximum of $k_{\text{cont.}}^{\text{base}}$ and $k_{\text{discrete}}^{\max}$:

$$w = \max\left\{c_0 A_{\text{eff.}} e^{\beta E_{\text{pol,hub}}}, 1\right\} \; k_{\text{discrete}}^{\max} \tag{3.41}$$

[7] c_0 is not the actual area density of clathrin above the membrane, since the effect of it would be limited, like $\frac{c_0 c_\infty}{c_0 + c_\infty}$. It is more like c_0 being an effective parameter that incorporates multiple effects, like how available clathrin is above the effective area, if it is well aligned (such that it can easily form bonds) or would need to rotate, and so on. All of these effects will reduce the rate with which clathrin will bind to the effective area, which is important further on.

[8] We see again here that the energy shift E_{χ}^{shift} is best understood as a shift in leg polymerisation energy.

This case differentiation hints at two interesting physical limits. We can either have a coat where the bond formation to new nodes dominates, because the polymerisation energy of the clathrin hubs is large, or a case where bond formation to already existing nodes in the coat dominates. The equilibrium case is achieved if the effective clathrin number fulfils $A_{\text{eff.}}c_0 \stackrel{!}{=} \exp(-\beta E_{\text{pol,hub}})$. This defines what we call the *equilibrium effective clathrin number* $A_{\text{eff.}}c_0^{\text{equil.}}$ defined to be

$$\boxed{A_{\text{eff.}}c_0^{\text{equil.}} := \exp(-\beta E_{\text{pol,hub}})} \tag{3.42}$$

With this, we define the *dynamic connectivity* ϵ as the ratio of the effective clathrin numbers:

$$\epsilon := \frac{A_{\text{eff.}}c_0}{A_{\text{eff.}}c_0^{\text{equil.}}} = A_{\text{eff.}}c_0 \exp(\beta E_{\text{pol,hub}}) \tag{3.43}$$

If $\epsilon > 1$, then the bond formation to new nodes dominates the grid growth. If, however, $\epsilon < 1$, than then the building of connections between already existing nodes will dominate the growth process, hence the name *dynamic connectivity*. We therefore find the following final two expressions:

$$\boxed{k_{\text{cont.}}^{\text{base}} = \epsilon\, k_{\text{discrete}}^{\max}} \tag{3.44}$$

$$\boxed{w = \max\{\epsilon, 1\}\ k_{\text{discrete}}^{\max}} \tag{3.45}$$

Coming back to the inclusion of new nodes into the grid. As mentioned before, the bottleneck in this process is the preliminary acceptance step, where the node addition is accepted with preliminary probability

$$\text{Prob}_{\text{acc, 1}} = \frac{k_{\text{cont.}}^{\text{base}}}{w} = \frac{\epsilon k_{\text{discrete}}^{\max}}{\max\{\epsilon, 1\}\, k_{\text{discrete}}^{\max}} = \min\{\epsilon, 1\} \tag{3.46}$$

which now becomes clear to be a real bottleneck. From our experience, ordered grid growth happens for values of $\epsilon \sim 10^{-6}$, which turns the acceptance probability for new node inclusion very low. We will later provide a trade-off to increase this acceptance probability.

3.8.5 Oversampling Equivalent Transitions

There is still a problem with how to define what constitutes an *event* and what a *rate* means. Simply put, if we only look at bond placements, the events "node i forms a bond to node j" and "node j forms a bond to node i" are **equivalent**. A rate k in its physical sense, however, means how likely the event "the legs of node i and j meet to form a bond" is.

In our code, we create a list of possible events by going through every node and for every node going through every formed or not-yet-formed leg and deciding what it can do. Therefore, the event that the rate k describes appears twice in that list, once for node i and once for node j.

To make this clear, a little example. Let us think of a system in a state A that can transition into two states, B and C. C is our case of interest, think of it as forming a bond between node i and j, and B could be "any other event happens":

$$\text{Rate}_A(X) = \begin{cases} q & X = B \\ k & X = C \end{cases} \tag{3.47}$$

For such a system, the average time it needs to escape state A to either B or C is

$$\langle T_{\text{exit}} \rangle = \frac{1}{q+k} \tag{3.48}$$

In the rKMC scheme, this is handled through $w = q + k$, either state being chosen by probability $1/2$ and accepted by q/w or k/w, and time being progressed at each step by $1/2w$. The average number of steps needed is then

$$\langle N_{\text{exit}} \rangle^{-1} = \frac{1}{2}\frac{q}{w} + \frac{1}{2}\frac{k}{w} = \frac{1}{2} \tag{3.49}$$

and also in the rKMC scheme, the total time elapsed is T_{exit}. Importantly, the ratio with which event B or C happens is given by

$$\text{Ratio}_{BC} = \frac{q}{k} \tag{3.50}$$

But now, let us inflate our state space by new states C', of which we think as physically equivalent to state C:

$$\text{Rate}_A(X) = \begin{cases} q & X = B \\ \tilde{k} & X = C \\ \tilde{k} & X = C' \\ \quad \vdots & \\ \tilde{k} & X = C^{N-1} \end{cases} \tag{3.51}$$

where we allowed the rate k to be modified to rate $\tilde{k}$. In this case, the overestimated escape rate $\tilde{w}$ is now

$$\tilde{w} = q + N\tilde{k} \tag{3.52}$$

and therefore the average single event acceptance rate decreases. This can be seen by the average number of steps needed to escape state A now being

$$\langle N_{\text{exit}} \rangle^{-1} = \frac{1}{N+1}\frac{q}{\tilde{w}} + \frac{N}{N+1}\frac{\tilde{k}}{\tilde{w}} = \frac{1}{N+1} \tag{3.53}$$

Now we come to the important metrics. The ratio with which event B happens to event C (or equivalent C's) happens now is

$$\tilde{\text{Ratio}}_{BC} = \frac{\frac{1}{N+1}\frac{q}{\tilde{w}}}{\frac{N}{N+1}\frac{\tilde{k}}{\tilde{w}}} = \frac{q}{N\tilde{k}} \tag{3.54}$$

and now becomes dependent on N, the multiplicity of the state C. Also the single step time changes to

$$\langle \Delta t_{\text{step}} \rangle = \frac{1}{(N+1)}\frac{1}{\tilde{w}} = \frac{1}{(N+1)(q+N\tilde{k})} \tag{3.55}$$

and therefore the escape time becomes

$$\langle T_{\text{exit}} \rangle = \frac{1}{q+N\tilde{k}} \tag{3.56}$$

In order for (3.51) to describe the same process as (3.47), we need to match (3.54, 3.56) with (3.50, 3.48). This is true if and only if

$$\boxed{\tilde{k} := \frac{k}{N}} \tag{3.57}$$

The take-away message is: If we have a physical event that appears with multiplicity N in our catalogue of events to choose from, we have to divide the true rate by N to still replicate the same process.

Applied to our system, this case happens whenever we try to form or remove a direct bond to nodes that exist in the coat, i.e., it does not apply to formation of direct bonds to new nodes. Here, we sample the event "nodes i and j form a bond" with multiplicity $N = 2$, since the catalogue of events that is created by looking at the legs of nodes, and therefore the upper event can be chosen either through a leg of node i or node j. The same is true for a bond removal between said nodes.

3.8.6 Increasing Acceptance Rate

If we implement the rKMC scheme as explained above, the efficiency of the algorithm, i.e., the average acceptance probability, will be quite low. As a reminder, it is defined as the rate q_{ij} for a transition $i \to j$ over $w \geq \sum_{j \neq i} q_{ij}$. With such an acceptance rate, we step forward in time by $\langle \Delta t \rangle = \frac{1}{Nw}$, where N is the number of processes.

But it turns out we can increase the acceptance rate of our algorithm while keeping the statistics the same by using a lower w. We will now define w to be an upper bound over the single rates, not the sum:

$$w \geq q_{ij} \; \forall \, j \tag{3.58}$$

and we denote with w^* the old upper bound over the sum over all rates:

$$w^* \geq \sum_j q_{ij} \tag{3.59}$$

We can now say that we choose a process j uniformly with probability $1/N$ and accept it with probability $q_{ij}/w \leq 1$. Correspondingly, we have to step forward in time by $\langle \Delta t \rangle = \frac{1}{Nw}$ instead of $\frac{1}{Nw^*}$.

Does this create the same process? The ratio with which event j is chosen over event k is

$$\text{Ratio}_{jk} = \frac{\frac{1}{N}\frac{q_{ij}}{w}}{\frac{1}{N}\frac{q_{ik}}{w}} = \frac{q_{ij}}{q_{ik}} \tag{3.60}$$

and is independent of w (or w^*). The average number of steps N_{exit} we need to leave state j is

$$\langle N_{\text{exit}} \rangle^{-1} = \sum_{j \neq i} \frac{1}{N} \frac{q_{ij}}{w} = \frac{1}{N} \frac{r_j}{w} \tag{3.61}$$

and per step we increase the time by $\langle \Delta t_{\text{step}} \rangle = \frac{1}{Nw}$. Therefore, the average time until we leave state i is

$$\langle \Delta T_{\text{exit}} \rangle = \frac{Nw}{r_j} \langle \Delta t \rangle = \frac{Nw}{r_j} \frac{1}{Nw} = \frac{1}{r_j} \tag{3.62}$$

3.8.7 Maximal Rates and Code Efficiency

Any proposed process in the KMC scheme is associated with a rate k and is accepted with probability

$$\text{Prob}_{\text{accept}} = \frac{k}{w} \tag{3.63}$$

where w is the maximal rate appearing in our system. This maximal rate is a priori unbounded, because a bond can be placed arbitrarily poorly, i.e., be associated with an arbitrarily high energy, and therefore its removal rate can be arbitrarily high.

However, we assume that between every change of macrostate, the system adiabatically relaxes the microstate, which keeps the probability of highly suboptimally placed bonds very low. As explained in Section 3.8.4, the realistic candidates for a maximal rate are the rate for bond formation to existing nodes and for bond formation to new nodes. With our previous definitions of the equilibrium effective clathrin number and the dynamic connectivity ϵ, this maximal rate is given by

$$w = \max(1, \epsilon)\ k^{\max}_{\text{discrete}} \tag{3.64}$$

Even though the value of ϵ can be chosen freely, from our experience values of $\epsilon \ll 1$ produced nicely ordered grids, as will be seen in the results. However, the usually proposed rates for bond formation between existing nodes are much smaller than the maximal possible rate, usually by two or more orders of magnitude.

It is therefore possible to increase the efficiency of our code by reducing the maximal rate below $k^{\max}_{\text{discrete}}$. We define the oversampling rate ρ as

$$w = \rho\ \max(1, \epsilon)\ k^{\max}_{\text{discrete}} \tag{3.65}$$

This introduces the possibility of processes being accepted with a probability $P > 1$, which introduces an error, as these events are then under-represented in the final system. But it is possible to keep the number of events with acceptance probability $P > 1$ low compared to the total number of proposed events. Therefore, it is possible to find a sweet spot between a small statistical error with large increase of code efficiency.

As mentioned before, the code efficiency is described by

$$\text{eff} := \frac{\langle k \rangle}{w} \tag{3.66}$$

Therefore, decreasing w will automatically increase the code efficiency. From our experience, the speed-up is on the order of factors of 2 to 200. A main driver of coat growth is the inclusion of new nodes to the lattice, which as previously mentioned is preliminarily accepted with probability

$$\text{Prob}_{\text{acc},1} = \frac{\epsilon\, k^{\max}_{\text{discrete}}}{w} \tag{3.67}$$

While before this ratio was given as $\min\{1, \epsilon\}$, it is now given as

$$\text{Prob}_{\text{acc},\,1} = \frac{\min\{1, \epsilon\}}{\rho} \tag{3.68}$$

which demonstrates the efficiency gain of the oversampling ratio ρ, and sets a sensible boundary with $\rho \leq \epsilon$.

Free Parameters	Unit	Meaning	Reference Value
γ	time^{-1}	The scale for the rates. Defines the timescale.	1
β	$\text{k}_\text{B}\text{T}^{-1}$	The inverse temperature. Sets the scale for the energy and defines which energy differences are important.	1
$E_{\text{pol,leg}}$	$\text{k}_\text{B}\text{T}$	The leg polymerisation energy (per leg) that gets released if a bond with another leg is formed.	$7\,\text{k}_\text{B}\text{T}$
E_χ^{shift}	$\text{k}_\text{B}\text{T}$	The shift in the pucker angle energy to keep $E_\chi(H = H_{\text{init}}) = 0$. To be understood as a shift in polymerisation energy.	-
$E_{\text{pol,hub}}$	$\text{k}_\text{B}\text{T}$	The hub polymerisation energy that gets released if a hub bonds to the membrane.	$1\,\text{k}_\text{B}\text{T}$
L_0	L_0	The equilibrium length of a clathrin leg.	1.
ΔL_{max}	L_0	The maximally expected deviation of the leg length.	$0.1\,\text{L}_0$.
k_L	$k_B T$	The stretching stiffness of a single clathrin leg. Defined to keep ΔL_{max}.	$800\,\text{k}_\text{B}\text{T}$
ϕ_0	rad	The ideal projection angle.	120°.
$\Delta\phi_{\text{max}}$	rad	The maximally expected deviation of the projection angle.	37°.
k_ϕ	$\text{k}_\text{B}\text{T}$	The bending stiffness of the clathrin legs in the tangential plane of the sphere. Defined to keep $\Delta\phi_{\text{max}}$.	$20\,\text{k}_\text{B}\text{T}$
χ	rad	The dipping or pucker angle. Depends directly on the chosen preferred curvature.	-
k_χ	$\text{k}_\text{B}\text{T}$	The bending stiffness of the clathrin legs in direction of the pucker angle.	$\kappa_\phi = 20\,\text{k}_\text{B}\text{T}$
H_{init}	L_0^{-1}	The initial curvature of the membrane, i.e., the inverse of the radius R of the sphere.	-
H_0	L_0^{-1}	The preferred curvature of the coat.	$0.5\,\text{L}_0^{-1}$
A_{Cl}	L_0^2	The area occupied by a single clathrin.	$1.3\,\text{L}_0^2$
κ_{Membrane}	$\text{k}_\text{B}\text{T}$	The membrane bending rigidity.	$20\,\text{k}_\text{B}\text{T}$.
σ	L_0	The exclusion radius for the hard shell exclusion potential.	$0.7\,\text{L}_0$.
ϵ	1	The dynamic connectivity. Defines whether internal connections or inclusion of new nodes dominate the growth process.	2×10^{-6}
ρ	1	The oversampling rate.	1×10^{-5}
N_{relax}	1	The number of Monte Carlo steps the system takes after a successful macrostate transition to find an energy minimum/to adiabatically follow the energy minimum.	50

Table 3.2: The free parameters for our simulation. Listed above are the parameters together with their units and their interpretation/effect on the system. Many parameters reference values can be estimated from Table 1.1.

3.9 Constraining the Free Parameters

With effectively 14 free parameters, summarised in Table 3.2, the model is *a priori* too unconstrained to be understood in a reasonable amount of time. It is therefore necessary to constrain the free parameters to reasonable values and then probe the system around these values. We will now put forward a hierarchy in which we can constrain the parameters and in doing so explain their influence.

3.9.1 Rate-Scale

The scale for the rates γ has no real influence on the model and only controls the timescale on which everything happens. Since everything in the code except for the time-propagation only depends on ratios of rates, the base rate always cancels. In our results, times are given in units of $1/\gamma$.

3.9.2 Temperature

The temperature T is included in the model only in the form of the inverse temperature $\beta := 1/k_B T$, which is used both for equilibrium Monte Carlo and kinetic Monte Carlo! The temperature effectively controls how important *energy differences* are. All rates for processes with energy difference ΔE are computed like $k = \gamma e^{-\beta \Delta E}$. If we have two processes (named "fast" and "slow") with energy differences ΔE_{fast} and ΔE_{slow} (with $\Delta E_{\text{fast}} \ll \Delta E_{\text{slow}}$), then the relative acceptance of these two processes is

$$\frac{k_{\text{fast}}}{k_{\text{slow}}} = \exp\left(-\beta(\Delta E_{\text{fast}} - \Delta E_{\text{slow}})\right) = \exp\left(\beta|\Delta E_{\text{fast}} - \Delta E_{\text{slow}}|\right) \tag{3.69}$$

So of course, the faster process will be accepted more often, but how much more depends on the (inverse) temperature. At high T, low β, energy differences become negligible, and all processes happen (almost) equally likely.[9]

However, all energies can be rescaled to units of $k_B T$, and therefore it is possible to set $\beta \equiv 1$. Therefore, all energies in our results are given in units of $k_B T$. For easier access to our model, we will, however, still use the fact that we can tune β.

3.9.3 Pucker Angle Energy Shift

The easiest parameter to fix next is the pucker angle energy shift. This parameter is in the model more for convenience, because we want to look at two kinds of processes. Either processes at *fixed curvature* or processes at curvature $H_{\text{init}} \neq H_0$, where the curvature usually grows towards H_0. Therefore, the pucker angle energy $E_\chi = k_\chi(\chi - \chi_0)^2$ will usually be positive at the initial configuration. But if we were to now change, e.g., k_χ, then the initial energy offset would also change, thereby potentially altering the rates drastically. To keep such unwanted effects at a minimum, we have introduced the pucker angle energy shift E_χ^{shift}, which we define to be such that the pucker angle energy at the initial curvature H_{init} is zero:

$$E_\chi(H = H_{\text{init}}) \equiv 0 \;\Rightarrow\; \boxed{E_\chi^{\text{shift}} = k_\chi \left(\chi_{\text{init}} - \chi_0\right)^2 \approx \frac{k_\chi L_0^2}{4}\left(H_{\text{init}} - H_0\right)^2} \tag{3.70}$$

[9] There are, however, restrictions: Some processes might not happen at all as they are impossible in certain grid configurations.

Usually, we will start at an almost flat initial curvature $H_{\text{init}} \approx 0$, where this energy shift becomes independent of the precise initial parameters, $E_\chi^{\text{shift}} \approx k_\chi \frac{L_0^2 H_0^2}{4}$.

This definition means that around the initial curvature H_{init}, the pucker angle energy will be around zero. But for small deviations of the curvature from the ideal curvature H_0, the pucker angle energy will give more and more negative contributions. This can be seen in Figure 3.11, which depicts the pucker angle bending energy for different values of k_χ for initial curvature $H_{\text{init}} = 0$. Through the definition of E_χ^{shift}, the energy at $H = 0$ always starts at zero, making scenarios with different k_χ easier to compare.

For most of the upcoming parameters, we will consider the initial configuration, where the pucker angle energy will deliver a contribution around zero for averagely placed legs. It is also worth mentioning that the shift in pucker angle energy can be understood as a **shift in leg polymerisation energy**. Absolute energy values do not play a role except in the form of a polymerisation energy, and therefore the shift of the pucker angle energy is important.

3.9.4 Leg Polymerisation Energy

Next in the constraining hierarchy is the leg polymerisation energy. This is the energy that is released for every leg that is bound to another leg. This means effectively that every leg that is placed contributes an energy

$$E_{\text{leg}} = E_{\text{angular}} + E_{\text{stretching}} - E_{\text{pol,leg}} \tag{3.71}$$

Let us assume a fixed curvature H around the initial curvature, such that the energy of the pucker angle is about zero for an otherwise ideally placed leg. Then an optimally placed leg has $E_{\text{angular}} + E_{\text{stretching}} = 0$. For such a leg to be thermally stable, we need the polymerisation energy of the leg to be larger than $k_B T$, i.e., $E_{\text{pol}}\ \beta \gtrsim 1$. Otherwise, optimally placed legs will dissolve thermally. The only mechanism to then build a stable grid would be if enough nodes randomly assemble to curve the membrane such that the energy gain from the pucker angle can stabilize the node addition. But this is not the behaviour we are interested in. Looking at Table 1.1, we see a total binding energy between $5 - 30\,\text{k}_\text{B}\text{T}$ per clathrin triskelion. In our model, this corresponds to $E_{\text{pol,hub}} + 6E_{\text{pol,leg}}$.

The polymerisation energy influences how costly bonds can be to still have a realistic chance of being accepted. If we define $\Delta\phi$ and ΔL as the deviation of the parameters ϕ and L of some optimal value, then the value $\Delta\phi_{\text{max}}$ or ΔL_{max} of still likely outliers is linked through the spring constants k_ϕ and k_L to the polymerisation energy. However, later, we will invert this relation and define the spring constants k_ϕ and k_L in such a way that for every polymerisation energy, the maximal deviations $\Delta\phi_{\text{max}}$ and ΔL_{max} are kept constant. We do this since we know from experiments (see (Albert J. Jin, Prasad, et al. 2006)) the numerical range of ΔL_{max} and $\Delta\phi_{\text{max}}$. We therefore constrain the geometry rather than the numerical value of the spring constants. In other words, the rate with which bonds at $\Delta\phi_{\text{max}}$ and ΔL_{max} are built stays constant.

3.9.5 Spring Constants

The spring constants are the parameters for the harmonic potentials for either the length L of legs or the angles ϕ and χ, even though the pucker angle will take on a special role in this discussion. In combination with the polymerisation energies they determine the range in which these parameters, now abbreviated simply with X, are realistically chosen.

For the time being, assume that X is a dimensionless number, i.e., given in units of some scale. For lengths, this is the equilibrium length L_0. The generic potential for X has the form

$$E(\Delta X) = k_X \, (\Delta X)^2$$

where $\Delta X := X - X_0$ with reference to some equilibrium value. Let us define $\Delta X_{\max}$ to be the deviation of X to X_0 that is realistically expected to occur. Then the energy of such a configuration is such that it is on the verge of becoming thermally unstable. This means[10]

$$E(\Delta X_{\max}) - E_{\text{pol,leg}} \approx k_B T = \frac{1}{\beta}$$

We can take this to be the definition of $\Delta X_{\max}$. For fixed maximal deviation $\Delta X_{\max}$, the spring constant k_X becomes a function of temperature β and polymerisation energy $E_{\text{pol,leg}}$. Plugging in the definition of $E(\Delta X_{\max})$ and rewriting, we get

$$\boxed{k_X := \frac{1 + \beta E_{\text{pol,leg}}}{\beta \Delta X_{\max}^2}} \tag{3.72}$$

This allows us, independently of the chosen polymerisation energies or temperature, to define spring constants that enforce some geometric configuration. From our experience, we gathered the impression that, e.g., values of $\Delta\phi_{\max} \approx 35°$ and $\Delta L_{\max} \approx 0.2 L_0$ lead to geometrically sound grid growths.

We can connect our maximal deviation $\Delta X_{\max}$ to the measured fluctuations of the projection angle ϕ and the leg length L of clathrin by Albert J. Jin and Nossal (2000), which are displayed in Table 1.1. For the projection angle, a standard deviation of $\Delta\phi_{\text{std.}} = 37°$ was measured for a single clathrin. Naturally, this angular deviation is a lot smaller if the clathrin is built into the grid. But our Hamiltonian describes a single free clathrin. For the stretching of the legs, both in free clathrin and bound clathrin, a typical deviation of around 10% of the equilibrium length was found. This sets our chosen value of $\Delta L_{\max} = 0.2 L_0$ on the right order of magnitude, maybe slightly too large. It is interesting to note that we have a clear relation between the polymerisation energy of clathrin legs and geometric constraints to the grid at fixed temperature. These are depicted in Figure 3.9. The deviation of the legs of roughly 20% corresponds to the light-blue line. For typical polymerisation energies on the order of $\sim 5\,\mathrm{k_B T}$ this corresponds to a spring constant of $k_L \sim 200\,\mathrm{k_B T}$. The angular deviation is measured in radian. A $\Delta\phi_{\max} \sim 35°$ corresponds to $\Delta\phi_{\max} \sim 0.6\,\text{rad}$, i.e., the red curve in Figure 3.9, where we expect a spring constant of $k_{\text{angular}} \sim 20\,\mathrm{k_B T}$.

Going back to the polymerisation energies, one may ask the question of what influence these polymerisation energies retain if by definition the maximal deviation of the parameters is kept constant? But because the overall energy of any bond is lowered, the overall acceptance rate for bonds placed more optimally than $\Delta X_{\max}$ is increased. To see this, despite the fact that the spring constants change, we can look at the energy of a bond at parameter ΔX. The energy of a leg with said parameter is given by

$$E_{\text{leg}} = E(\Delta X) - E_{\text{pol,leg}} \quad \text{with} \quad E(\Delta X) = k_X \Delta X^2 \quad \text{with} \quad k_X = \frac{1 + E_{\text{pol,leg}}\,\beta}{\beta\,\Delta X_{\max}^2} \tag{3.73}$$

[10] Whether we have a larger factor in front of $E_{\text{pol,leg}}$ is debatable, but for an order of magnitude, it suffices.

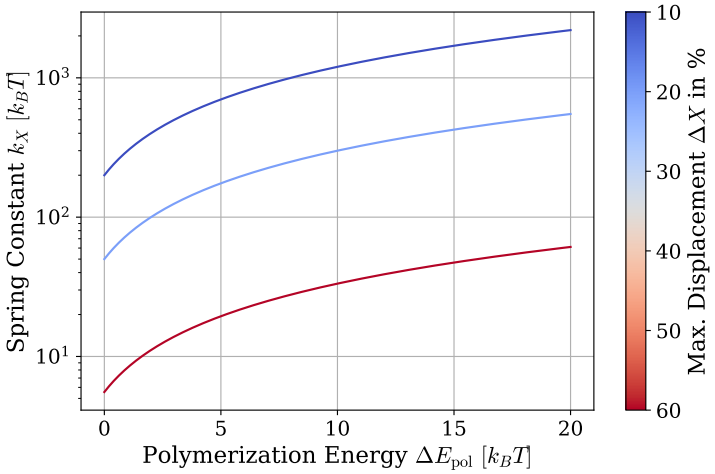

Figure 3.9: Spring constants against polymerisation energy. For a relative deviation of ΔX indicated by colour, the dependency of the spring constant κ_X is shown against the polymerisation energy.

and therefore, for two polymerisation energies E^A_{pol} and E^B_{pol} we find the differences in the leg energies to be

$$\begin{aligned}
\Delta E_{\text{leg}} =& E^A_{\text{leg}} - E^B_{\text{leg}} = k^A_X(\Delta X)^2 - E^A_{\text{pol}} - \left[k^B_X(\Delta X)^2 - E^B_{\text{pol}}\right] \\
=& -\left(E^A_{\text{pol}} - E^B_{\text{pol}}\right) + (\Delta X)^2\left[\frac{1+E^A_{\text{pol}}\beta}{\Delta X^2_{\text{max}}\beta} - \frac{1+E^B_{\text{pol}}\beta}{\Delta X^2_{\text{max}}\beta}\right] \\
=& -\left(E^A_{\text{pol}} - E^B_{\text{pol}}\right) + \left(\frac{\Delta X}{\Delta X_{\text{max}}}\right)^2\left(E^A_{\text{pol}} - E^B_{\text{pol}}\right) \\
=& -\Delta E_{\text{pol}}\left(1 - \left(\frac{\Delta X}{\Delta X_{\text{max}}}\right)^2\right)
\end{aligned} \tag{3.74}$$

with $\Delta E_{\text{pol}} = E^A_{\text{pol}} - E^B_{\text{pol}}$. We can normalize everything by E^B_{pol} to find

$$\frac{\Delta E_{\text{leg}}}{E^B_{\text{pol}}} = -\left(\frac{E^A_{\text{pol}}}{E^B_{\text{pol}}} - 1\right)\left(1 - \left(\frac{\Delta X}{\Delta X_{\text{max}}}\right)^2\right) \tag{3.75}$$

This relation is shown in Figure 3.10.

3.9.6 Pucker Angle Spring Constant

The pucker angle χ controls how sensitive the system is with regards to the curvature. As a quick recap, χ can be quickly computed as

$$\chi_i \approx \frac{L_i H}{2} \tag{3.76}$$

where L_i is the length of the i-th leg and $H = 1/R$ is the curvature. For on average well-restrained leg lengths, we can assume $L_i \approx L_0$ which yields $\chi \approx L_0 H/2$. If we want to increase the system's sensitivity towards the curvature, we have to increase k_χ, the spring constant associated with the harmonic potential in the dipping angles:

$$E_\chi := k_\chi\left(\chi - \chi_0\right)^2 \tag{3.77}$$

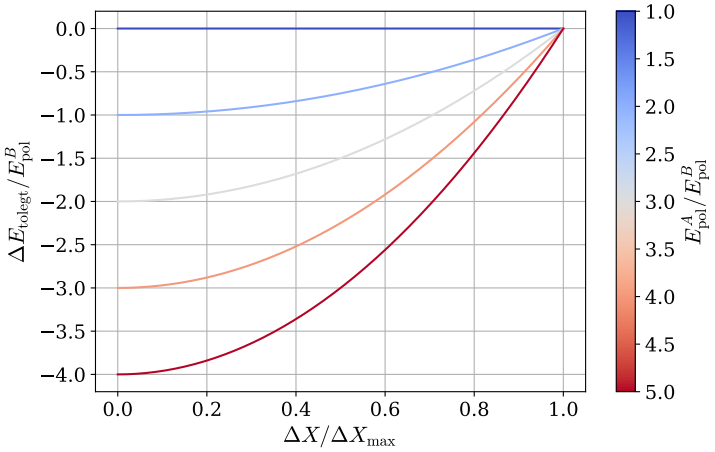

Figure 3.10: Increased polymerisation energies lead to decreased energies. Even though the maximal deviation $\Delta X_{\max}$ is kept constant for different polymerisation energies, more optimal configurations $\Delta X < \Delta X_{\max}$ receive a lower energy for larger polymerisation energies, showing the effect of the polymerisation energies.

In order for us to not completely change all the energetics if we change the χ spring constant, we have shifted the energy in such a way that the initial energy resulting from the dipping angle is always zero, i.e., $E_\chi(H = H_{\text{init}}) \equiv 0$. For this, we remember that the maximum energy to be drawn from the pucker angle is

$$\Delta E_\chi^{\max} = k_\chi \left(\chi_{\text{init}} - \chi_0\right)^2 = \frac{k_\chi L_0^2}{4} \left(H_{\text{init}} - H_0\right)^2 \tag{3.78}$$

which will be subtracted for every placed bond, therefore setting the initial energy tax from the pucker angle to zero. At the ideal radius, the energy gain per bond resulting from the pucker angle would be precisely (3.78). Changing k_χ would have no impact on the initial energy setting and would simply lower the energy well in which the system can settle at the ideal angle. This can be seen for different values of k_χ in Figure 3.11.

We see that if the curvature starts to go in the right direction, i.e., $H \nearrow H_0$, the energy gain per bond increases, and can, for high enough k_χ, reach values around the chosen polymerisation energy. It can therefore be entirely possible to see a **cooperative** behaviour. Bonds start to bend the membrane, through which the addition of new nodes becomes more likely, through which the membrane gets shifted even more.

So we can have two cases, either of which will guide the system to the correct curvature. Either the maximum possible energy gain through the pucker angle is not that significant compared to the leg polymerisation energy. Then only the total effect of many bonds will bend the membrane, nothing more. Or the single-bond energy gain is on the order of the polymerisation energy, in which case we expect to see an increase in the rate with which nodes become connected.

3.9.7 Hub Polymerisation Energy

The hub polymerisation energy is the energy released if a clathrin hub is placed on the membrane. This necessarily goes hand in hand with a bond formation, so the polymerisation energy necessarily released when a clathrin hub is placed on the membrane is $E_{\text{pol, new node}} = E_{\text{pol,hub}} + 2E_{\text{pol,leg}}$.

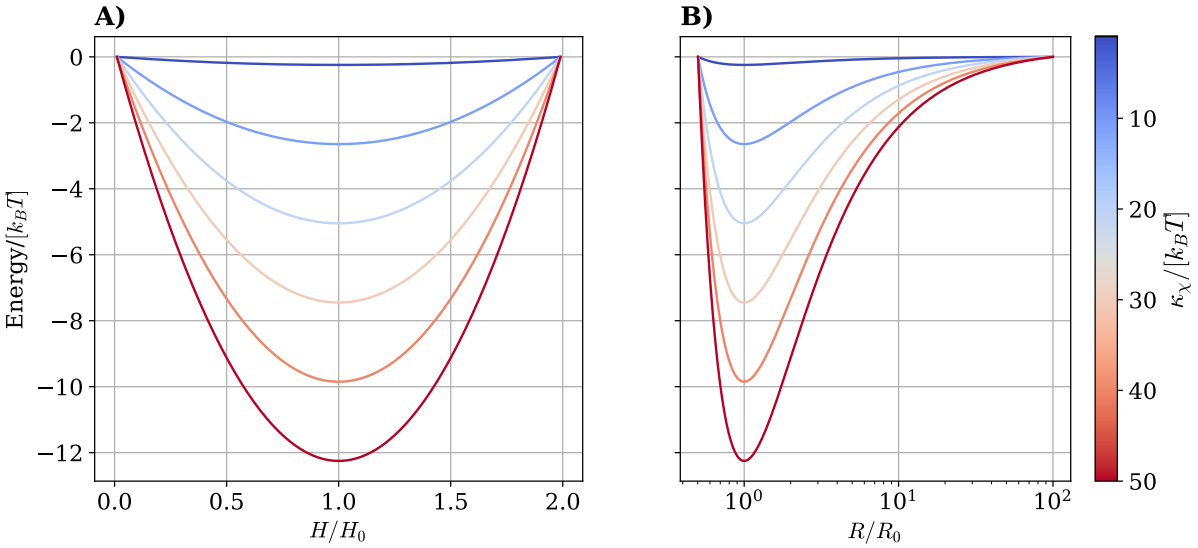

Figure 3.11: The dipping angle energy vs. curvature and bending radius. The energy resulting from the dipping angle potential is shown against the curvature (A) and against the bending radius (B). The energy follows Equation (3.77) where the initial energy (3.78) is set to zero by subtracting it. Therefore, all curves starting from $H = 0$ or $R \to \infty$ have the same initial energy of zero, and the energy well at the ideal curvature H_0 is deepened for higher values of k_χ.

If we have already constrained the legs to be thermally stable if ideally placed, then a new node ideally placed is automatically thermally stable, even with $E_{\text{pol,hub}} = 0$. The way in which we have defined our system, the hub polymerisation energy also has no immediate effect on the node addition rate. Since the maximal node addition rate $k_{\text{add.}}^{\max}$ is defined through the effective clathrin number $A_{\text{eff.}} c_0^{\text{equil.}} = \exp(-\beta E_{\text{pol,hub}})$ and the dynamic connectivity ϵ, see Section 3.8.4, the hub polymerisation energy is not important for the rate.

However, the hub polymerisation energy plays a role in the stability of nodes. A node is less likely to be removed with a high hub polymerisation energy, since the overall energy loss for removing a node is increased. But again, the leg polymerisation energy already plays a role here. So a value of $E_{\text{pol,hub}} > 0$ means that even very suboptimally placed nodes will not be removed from the grid.

3.9.8 Effective Clathrin Number and Dynamic Connectivity

The effective clathrin number is a quantity appearing in the calculation of the rate for node addition, introduced in Section 3.8.4. It has units of an area times an area-density. The area stems from the integration of the probability density function of where a new node can be placed, and therefore corresponds to the effective area onto which a new clathrin can bind. The area density c_0 necessarily arises when we try to connect the rate for node addition with the base rate γ. Both parameters can be combined. They appear finally in the expression for the maximal rate for node addition:

$$k_{\text{node addition}}^{\max} = A_{\text{eff}} c_0 \; e^{\beta E_{\text{pol,hub}}} \; k_{\text{bond formation}}^{\max} \tag{3.79}$$

The ratio of $k_{\text{node addition}}$ to $k_{\text{bond formation}}$ determines whether our system is dominated by the process of node addition or internal bond formation. For equal rates, we can define

the *equilibrium effective clathrin number* $A_{\text{eff}}c_0^{\text{equil.}}$ to be

$$\boxed{A_{\text{eff}}c_0^{\text{equil.}} := \exp\left(-\beta E_{\text{pol,hub}}\right)} \tag{3.80}$$

and define a new parameter ϵ, the **dynamic connectivity**, to be

$$\boxed{\epsilon := \frac{A_{\text{eff}}c_0}{A_{\text{eff}}c_0^{\text{equil.}}} = \frac{k^{\max}_{\text{node addition}}}{k^{\max}_{\text{bond formation}}}} \tag{3.81}$$

which controls whether the node addition rate ($\epsilon > 1$) or the bond formation rate dominates ($\epsilon < 1$) during grid growth:

$$\boxed{k^{\max}_{\text{node addition}} = \epsilon\, k^{\max}_{\text{bond formation}}} \tag{3.82}$$

Hence the name dynamic connectivity. From our experience, low values of $\epsilon \sim 10^{-6}$ lead to coordinated grid growth. This concludes the discussion about the restriction of parameters.

Chapter 4

Results

We have now built up all machinery that explains how we aim to model the clathrin coat during CME, and are ready to analyse the results. I will introduce this chapter with a section on the general data we can produce for any grid and the metrics we analyse. After that, I will go through different cases where the system produced some interesting results.

4.1 Metrics and Visualisations

At this point we can simulate a clathrin coat by running our model. However, the processes that shape the evolution and final form of our coat are highly complex and interwoven. Let us therefore get familiar with common parameters we look at and visualisation methods we use.

To investigate a clathrin coat, we usually start by growing it. This process starts by *defining* the grid

```
grid = Grid(params)
```

where `params` are the *physical* parameters together with *unphysical* parameters like the *oversampling rate* ρ defined in Table 3.2 the and *initial conditions* like the initial grid spacing. For simplicity, we start every growth process with an initial hexagon.

After defining the grid, we simulate its evolution

```
final_grid, data_dict, time_stacked_batched_nodes = grid.
    evolve_grid_till(steps=evolution_steps)
```

and gather the grid data simultaneously in the two objects `data_dict` and `time_stacked_batched_nodes`. The latter carries raw information about the clathrin nodes in our system over time, the former has more refined data of our growth process.

How would we look at a grid? One possibility to look at the grid state is as a two-dimensional projection. In this depiction, we project every node using a *stereographic projection* (see Appendix A.2) onto a plane. Because the stereographic projection is singular around one pole, we actually project the northern and southern hemisphere separately onto two different planes. To connect to the spherical geometry, we mark with concentric circles the lines with horizontal angles $\theta = \pi/4$ and $\theta = \pi/2$. The nodes representing clathrin are shown as points. Around every point the exclusion zone defining the exclusion potential (see Section 3.5.3) is indicated by a circle. The bonds between nodes are shown as arrows, originating from their node. The arrow placement is as explained in Figure 3.2. For completeness, unconnected (direct) bonds are shown in light

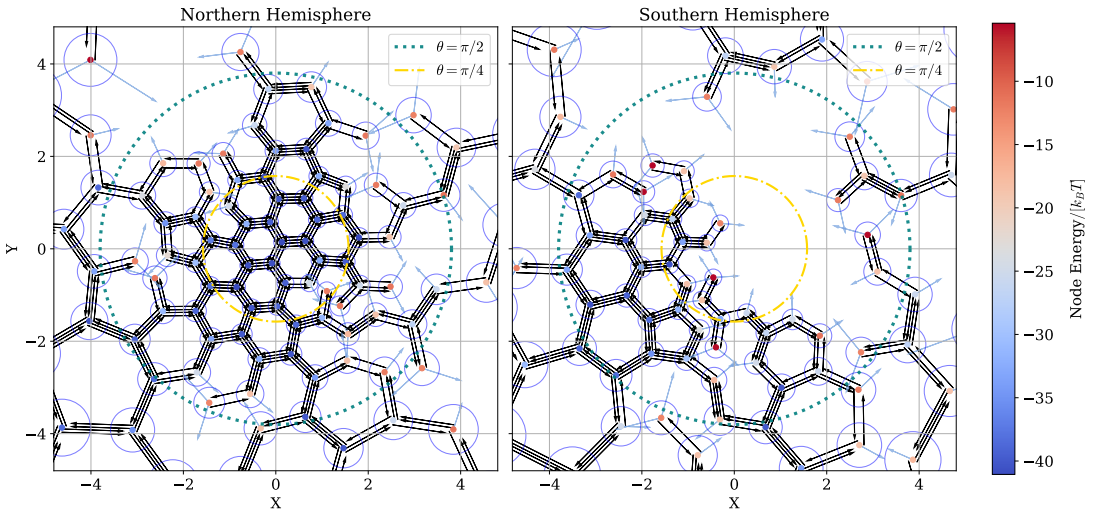

Figure 4.1: Two-Dimensional Grid Visualisation. A grid can be visualised by projecting the northern and southern hemisphere to a plane. Nodes representing clathrin hubs are depicted as points. Their energy is shown by colour. Legs are depicted using arrows originating from their node, in the way explained in Figure 3.2. We usually mark the $\theta = \pi/4$ and $\theta = \pi/2$ horizontal angle circles. Unconnected legs are represented in light blue, and are placed symmetrically around the existing legs. The exclusion areas are marked by blue circles around the black nodes.

blue. They are placed symmetrically around the existing bonds. An example of such a depiction is shown in Figure 4.1.

Another option to visualise the grid is as a three-dimensional direct representation. This is best done via a 3D rendering software such as `blender`. The grid configuration can be easily exported as a compressed numpy `.npz` file using

```
grid.export_blender_data(folder_path = "folder_path")
```

An example script of how to import the data into blender using a custom `bpy` script is provided in the GitHub repository. Nodes are again shown as points. Instead of showing all bonds, we usually only show direct bonds. Since they are always bidirectional, we only show bonds between nodes as a volumetric line. An example of such a depiction is shown in Figure 4.2.

During grid growth, data is constantly being gathered. This way we can look at several metrics such as the *energy*, the number of *bonds* between different clathrins or the number of formed polygons during growth. The usual parameter to plot this against is *time* itself. An example of such a collection of data can be seen in Figure 4.3 for a single grid. For this chosen grid, we can see that the overall energy of the system decreases during growth, together with the relative energy per clathrin (i.e., per system size), telling us that the growth is driven by a downward energy process. The bonds per clathrin tell us how interconnected our clathrin grid is at any given moment. We can also directly read off the number of polygons in the coat.

For some questions we do not allow clathrin removal during grid growth. For such processes, the clathrin number can serve as a pseudo-time. The advantage is that we

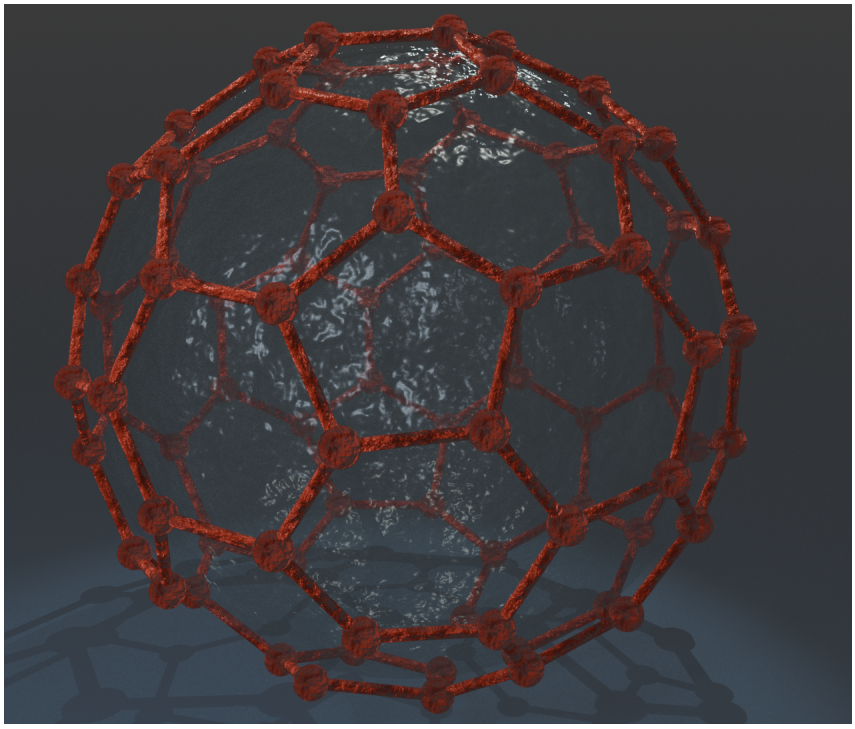

Figure 4.2: 3-Dimensional Grid Visualisation. A grid can be visualised directly in three-dimensional space. This can either be very diagrammatic or in a rendered fashion, like presented here. Nodes are again represented as points. To not overload the visualisation, only direct bonds are shown as lines.

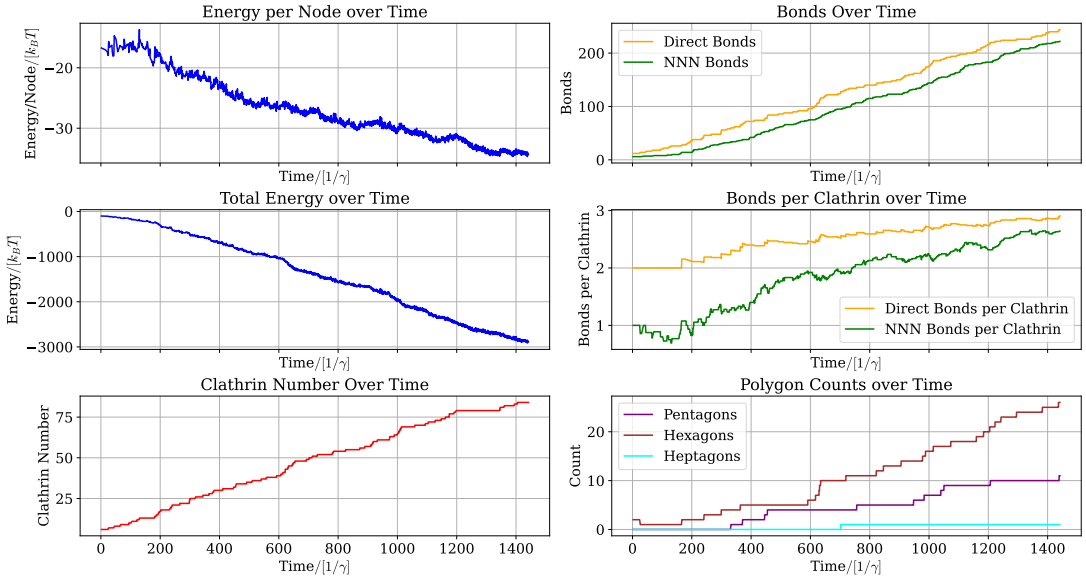

Figure 4.3: Parameter evolution during grid growth. This set of parameters allows us to evaluate a single grid evolution over time. We see that the system moves down the energy landscape, with the relative energy per node decreasing. While the total bond number increases as the grid grows, the relative bond number per clathrin also increases, showing that the grid becomes denser. In the chosen process, the clathrin number increases monotonically over time, as no bond removals are allowed. Finally, we can look at the number of penta-, hexa- and heptagons in the grid to check if it fulfils Euler's polyhedron formula.

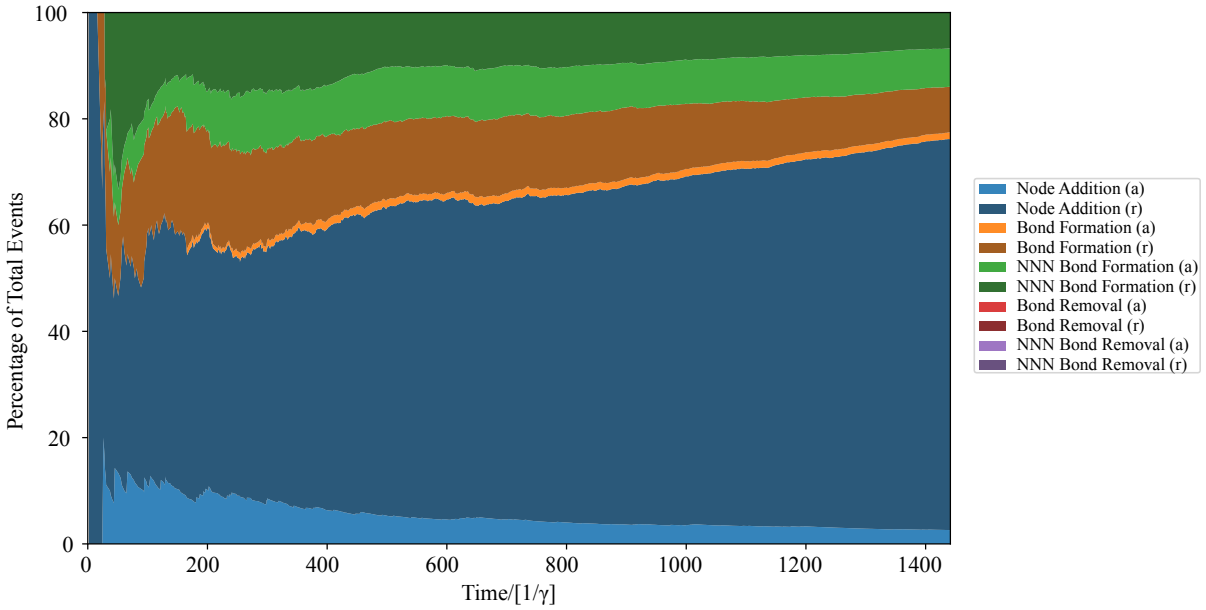

Figure 4.4: Example process contribution over time. The contribution of the five different grid growing processes over time can be visualised through this stacked plot. Every proposed process is added and normalised to unity. Then each process is marked by its colour, and the accepted (a) and rejected (r) processes are indicated by lighter or darker shades.

expect many properties of the system to depend on the number of clathrins involved, not on the absolute value of elapsed time (which only correlates with the total clathrin number). For other questions, we can also let the grid curvature evolve in time. Then a natural additional plot would be curvature H vs. time t.

To understand the dynamics of the growth process, we can also plot the contributions of different processes during grid growth, as can be seen in Figure 4.4. In that example, we can see that the initial grid growth process is dominated by proposed node additions, the only possible process when starting from an initial hexagon and not allowing node or bond removals. After the first accepted node additions (light blue), bond formation processes are suggested (green and brown). After an intermediate phase where bond formation processes dominate, the node addition processes take over again when the grid approaches its final closed state.

While understanding the distribution of processes during our growth process is necessary, it is also important to know the distribution of our proposed probabilities with which processes are accepted, which can be seen in Figure 4.5. Here, every accepted event with its probability ($P > 0$) is shown in a histogram sorted by its process type. In the example shown, we can see a peak around zero for very unlikely (though not ruled out) processes. We also see a peak around $P = 0.2$ for node addition events. This is the preliminary acceptance probability. Since for this process $\epsilon = 2 \times 10^{-6}$ and $\rho = 10^{-5}$, it is given by

$$\mathrm{Prob}_{\mathrm{acc},1} = \frac{\epsilon}{\rho} = 0.2$$

Because we are using *oversampling* for faster simulations, we also have to look at how many events are accepted with a probability $P > 1$, and would therefore be underrepresented in the grown grid. These processes are shown in the inset. Especially for the investigation

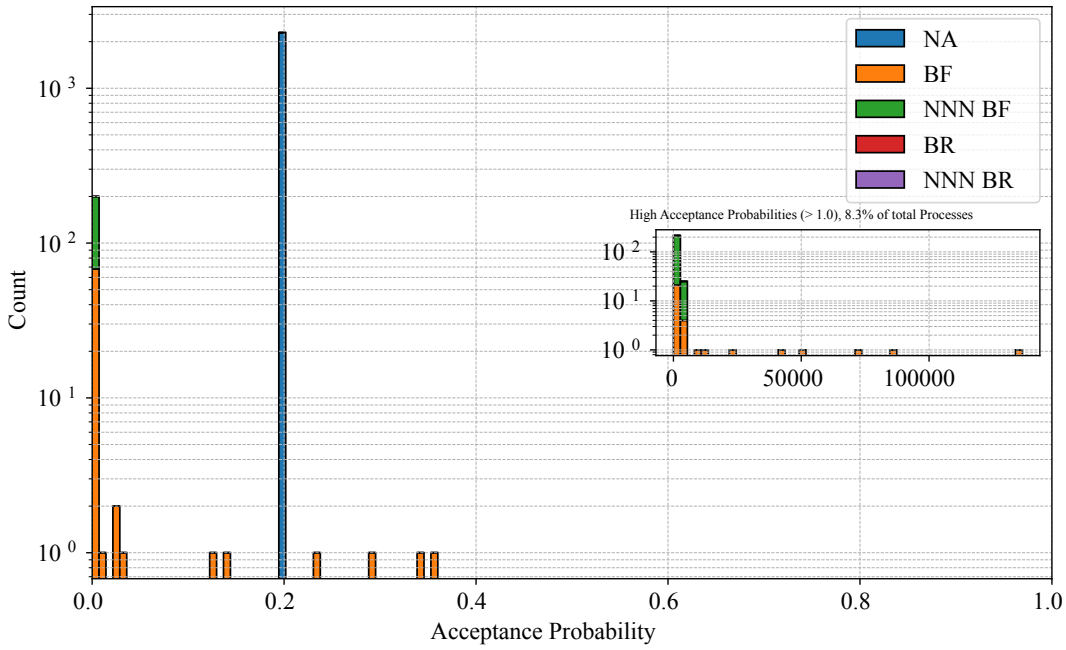

Figure 4.5: Example distribution histogram of acceptance probabilities. The proposed acceptance probabilities during grid growth sorted by process (for a growth w.o. node or bond removals) as a histogram. The inset shows the *oversampled* events with probability $P > 1$ and their relative share of the total number of processes.

of grid properties during growth, we should keep this percentage low so as not to skew the growth process.

4.2 Sanity Checks

To demonstrate the validity of our approach, we will present some proofs of concept regarding the legitimacy of the algorithm in the following section.

4.2.1 Brownian Motion of Free Clathrin Pairs

In our model, clathrin does not bind to a specific location on the membrane but just *to the membrane*. Its location is subject to Monte Carlo reshuffling. For a free clathrin, this should give rise to movement akin to Brownian motion on a sphere, i.e., in $d = 2$ dimensions. While the sphere is compact, as long as we do not look at excessively long observation times, times long enough for the particle to travel around the sphere, the compactness should not alter any results beyond the standard 2D expectation. This can be achieved by simply turning the sphere very flat.

We can test whether the motion does indeed follow Brownian motion quite easily. While our code does not allow a single particle to exist on the membrane (as this would not have a partner and is therefore deemed *inactive*), we can circumvent this problem by looking at a pair of stiffly connected clathrin nodes and simply look at the time evolution of the node pair's position.

We can analyse their motion and thereby determine the mean-squared displacement (MSD). The mean squared displacement at lag-time τ for a time-series $x(t)$ is defined to

be

$$\mathrm{MSD}(\tau) := \left\langle (x(t+\tau) - x(t))^2 \right\rangle_t \tag{4.1}$$

For Brownian motion in d dimensions, the diffusion constant D is linked to the MSD, first derived by Einstein in 1905 (Einstein 1905).

$$\mathrm{MSD}(\tau) = 2d\,D\,\tau \;\Rightarrow\; \log\left(\mathrm{MSD}(\tau)\right) = \log(2d) + \log(D) + \log(\tau) \tag{4.2}$$

Therefore, taking the log, assuming perfect diffusion, we expect a linear dependence with slope 1 whose y-intercept allows the determination of the diffusion constant D. In Figure 4.6 we can see the mean squared displacement for two independent node pairs in a log-log plot against the lag-time τ.

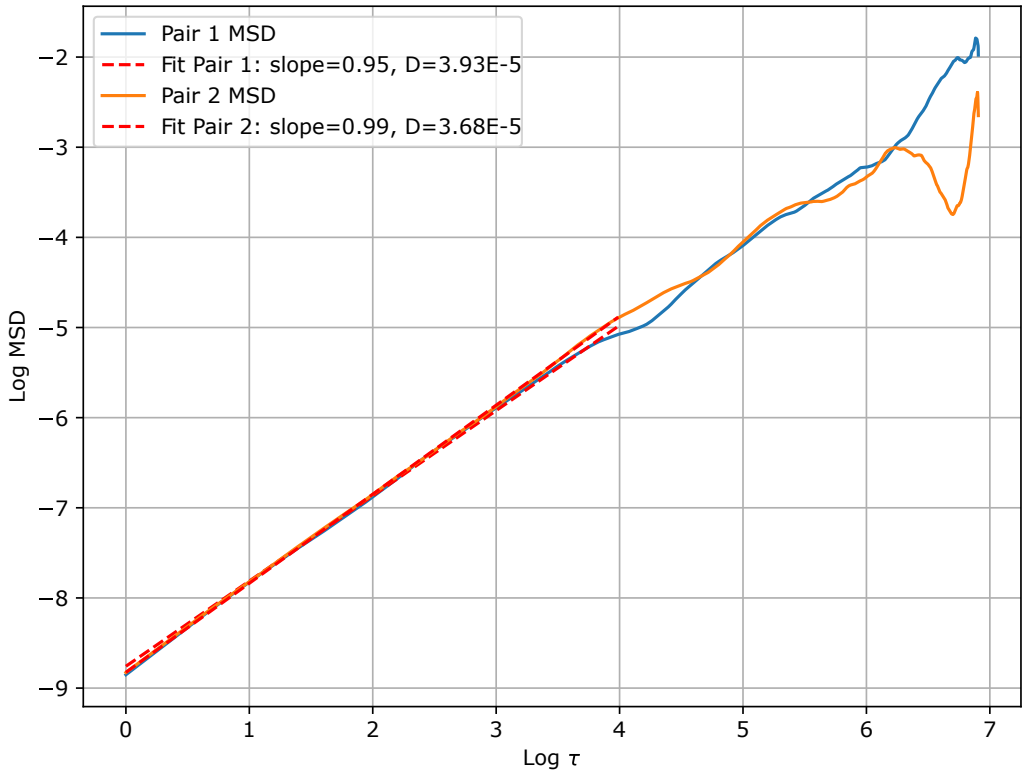

Figure 4.6: MSD of clathrin node pairs showing Brownian motion. The MSD of two coexisting node pairs displaying Brownian motion, indicated by the slope of ≈ 1. Both pairs display a similar diffusion constant, revealed by the y-axis displacement.

For both pairs, the slope is slightly below one, signalling an almost normal diffusion (Brownian motion). We can speculate whether we should see two different diffusion rates depending on whether we observe movement in the direction of the connection between the two nodes or orthogonal to it. One could also consider whether the presence of other pairs (with repulsive potentials) can impair diffusion.

4.2.2 Growing Regular Polyhedra

We hypothesised that for a fixed curvature, i.e., a sphere of fixed radius, on which the grid grows, there should exist a set of grid configurations that minimise the energy. And while it

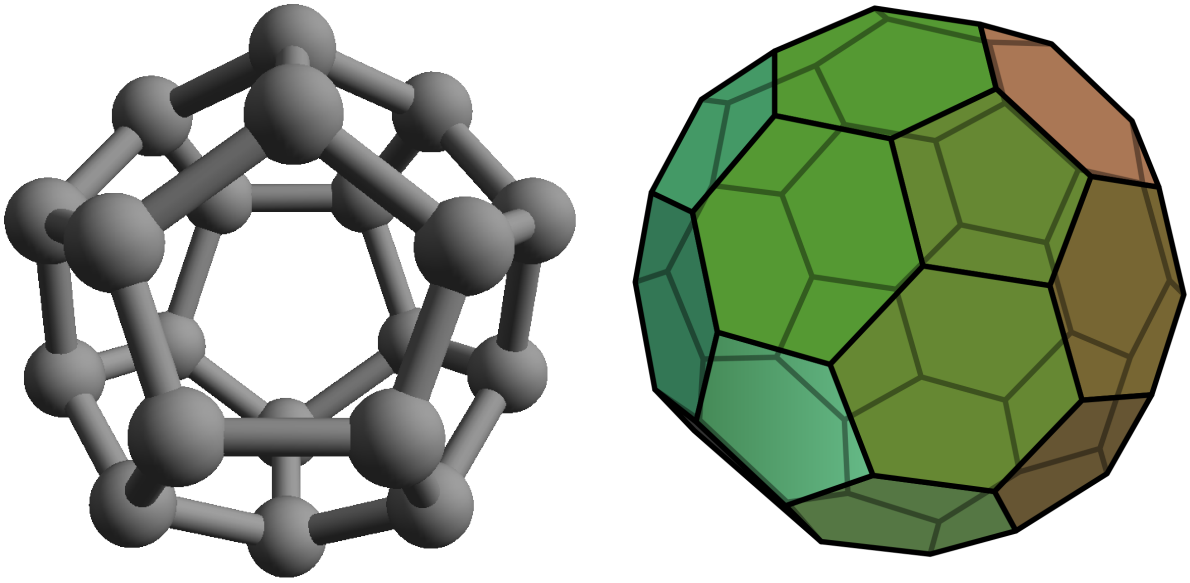

Figure 4.7: The dodecahedron and truncated icosahedron. Shown are the two candidates for energy minimising configurations at certain radii. Left: The dodecahedron, comprised of only 12 regular pentagons. Right: The truncated icosahedron composed of 12 regular pentagons and 20 regular hexagons. Every pentagon is circled by five hexagons, and at every vertex, two hexagons and a pentagon meet. Both figures taken from Wikimedia; left: By Perditax - Own work, CC0, right: By Cyp - Own work, CC BY 4.0.

is unlikely that a dynamically growing grid, using only local interactions, actually reaches this optimal configuration, we wanted to investigate whether under the right circumstances our code could somewhat reliably produce configurations close to the optimal ones.

For that, we need some candidates for such *optimal* configurations. We have to keep in mind that an optimal configuration at a certain curvature might be very suboptimal at another curvature. Therefore, let us first think about what are candidates for an optimal configuration, and then about under which circumstances these configurations really minimise the energy.

Every node individually prefers to be included in three hexagons. However, we know that a closed cage has to include at least 12 non-hexagons. Since every non-hexagonal polygon will increase the energy, the lowest energy configuration should be a coat with exactly 12 pentagons and otherwise only hexagons. Since any asymmetry in one of the polygons increases the energy, the minimal energy configuration should be comprised of regular polygons.

These restrictions naturally lead us to the Archimedean and Platonic solids as regular, convex polyhedra. The two polyhedra out of this class composed solely of pentagons and hexagons are the *dodecahedron* (12 pentagons, 0 hexagons) and the *truncated icosahedron* (12 pentagons, 20 hexagons). The two are depicted in Figure 4.7. In nature, these shapes are often found in carbon atoms as C_{20} and C_{60}, which are called *fullerenes*, first discovered in 1985 (Kroto et al. 1985), but are also found for clathrin-coated vesicles (Fotin et al. 2004).

While both are highly regular, the dodecahedron possesses a very high curvature compared to the truncated icosahedron (TI). Using the formula to relate the number of nodes to the preferred curvature (see Appendix A.3), the dodecahedron has a curvature of $H_D \approx 0.7/L_0$, while the TI has $H_{TI} \approx 0.4L_0^{-1}$. The higher the curvature, the more

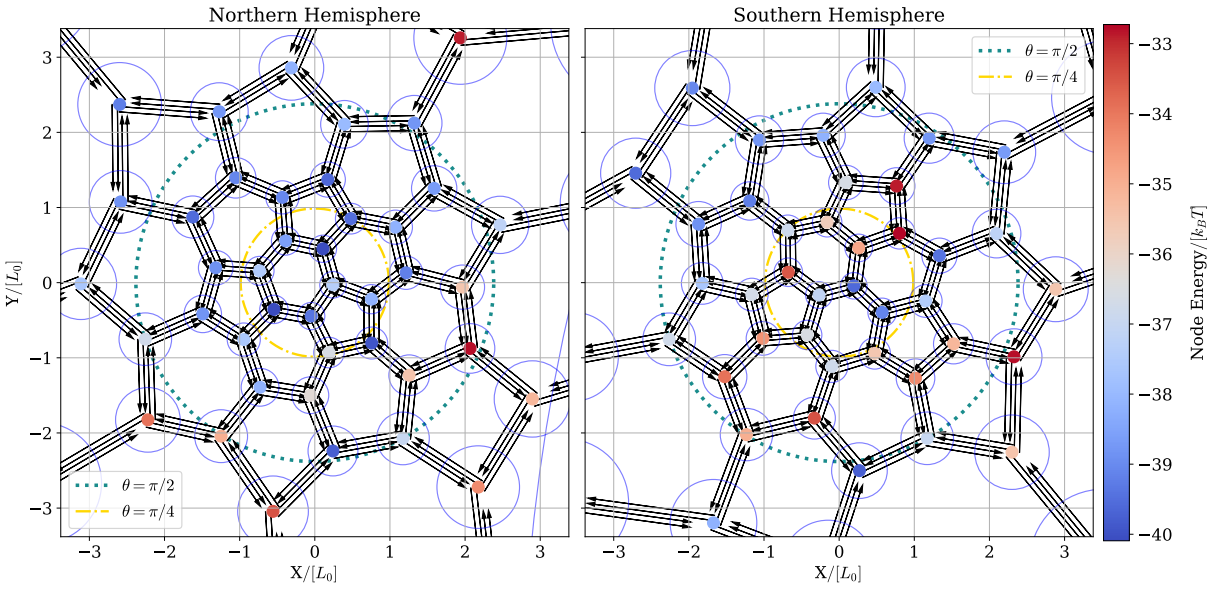

Figure 4.8: A grown grid approaches a truncated icosahedron. A grid grown close to the curvature ideal for a truncated icosahedron forms a closed cage that closely resembles a truncated icosahedron in structure and regularity.

important finite-size effects become. Since we would like to keep them small, we will concentrate on the TI.

If our code produces energy-driven grid configurations, given the correct parameters, our model should be able to build grids similar to the TI. The necessary condition for this is that the grid has the correct curvature, i.e., the sphere on which it grows has the correct radius to fit the correct number of nodes on it.

The dynamics of the growth can have an impact on the outcome. If, e.g., the potential for the angular energy is chosen too small, i.e., deviations from the preferred angles are punished less, then more irregular grids will grow. The details of how we choose these potentials will be discussed in the next section. The rate at which bonds and nodes are removed also plays a role. If the grid has many possibilities to form the preferred bond, it becomes more likely that at some point the preferred bond is actually formed.

However, when we look at the grids we have grown at curvatures close to the TI-curvature, we see that even without node and bond removals the grid grows grid conformations very close to the TI. An example of such a grid is shown in Figure 4.8, with additional evolutionary data provided in the appendix in Figure B.1.

The grid shown in the example was grown at $H = 0.42/L_0$, so slightly above the ideal curvature. It consists of 58 nodes and forms 12 pentagons and 19 hexagons. The slight irregularity in the number of hexagons is most likely due to the slightly too large curvature. As we can see from the placement of the pentagons, for the majority of the growth processes the form followed the TI perfectly. Only at the closing stage does the pentagonal distribution deviate from the ideal one. However, based on our observation of numerous grid growths, this is naturally the most difficult part of the evolution. It is spatially very restricted, depends heavily on the precise configuration of the coat and is prone to build-out defects that can only be remedied through node or bond removal.

We also have the ability to access the individual node's energy directly, which is indicated by colour in Figure 4.8. It is interesting to note that nodes with higher energy are not necessarily only the nodes involved in pentagons, but are nodes that are involved in the formation of lattice configurations that deviate from the TI. This shows that it is energetically possible to identify suboptimally placed nodes in the lattice. And therefore the number of high-energy nodes is significantly higher on the southern hemisphere where the closure happens, and a suboptimal lattice configuration has emerged.

4.3 Defining a Differentiable Minimal Energy

Of course, the precise value of the total energy depends heavily on the grid configuration, and since our system is at a finite temperature, the energy of any configuration will be increased with respect to its *zero-temperature limit*. By that expression we mean the following: For any given macrostate of the grid, the energy still depends heavily on the microstate, i.e., the positions $\boldsymbol{x}$ of its nodes. We can define the zero-temperature limit of this energy as the minimal energy over the node positions:

$$E_{\mathrm{T}=0} := \min_{\boldsymbol{x}}\{E(\boldsymbol{x})\} \tag{4.3}$$

Though it is hard to find this energetic minimum analytically, we can use the fact that we can calculate the forces $\boldsymbol{F}$ on the node positions $\boldsymbol{x}$ by using automatic differentiation:

```
force = jax.grad(energy(coordinates_2d))
```

By following the force trajectories (i.e., performing a noiseless gradient descent, see 2.6.2) the node positions will essentially settle in an energetic minimum. Note that the differentiation is performed with respect to the 2D coordinates, i.e., we calculate forces in the tangential plane of every node.

With this idea, we can define an effective curvature- and macrostate-dependent energy $E_{\mathrm{eff}}(H, \mathrm{config})$ as the zero-temperature limit of the position-dependent energy:

$$E_{\mathrm{eff}}(H, \mathrm{config}) := \min_{\boldsymbol{x}}\{E(\boldsymbol{x}, H, \mathrm{config})\} \tag{4.4}$$

In the following, we will often drop the explicit mention of the grid configuration.

One of our main goals is to find the effective energy of a coat. This will obviously heavily depend on the realised grid configuration. However, using the zero temperature limit of the energy, we minimise this dependency. Our hope is that the so defined energy becomes more tractable during the growth of a grid.

4.3.1 Effective Bending Energies of Coats

As explained in 2.1, we designed our energy function with the goal in mind that the final coat shows as an emergent property a Helfrich-like bending energy. Since the curvature is constant over our coat, we want to test if there exists a function $E_{\mathrm{eff}}(H)$ that only depends on the curvature of the coat H that approximately describes the curvature-dependent energy of our coat, and whether that function has the Helfrich form

$$\boxed{E_{\mathrm{eff}}(H) = \kappa_C (H - H_{\mathrm{min}})^2 A} \tag{4.5}$$

For this, we can look at the minimised energy of our system at different curvatures H, which is shown in Figure 4.9 A) for a representative example. There one can see the presumably quadratic dependency of the energy on the curvature.

We will later try to fit the quadratic Helfrich energy to these data. We already remarked that it seems that the energy minimising curvature H_{min} is shifted away from the spontaneous curvature of clathrin, $H_0 = 0.42/[L_0]$ in this example.

If we try to fit Equation (4.5) to the energy, we introduce the effective parameter κ_C which we call the *effective coat bending rigidity*. A is a measure of the area of the clathrin coat, and H_{min} is the *energy minimising curvature*. While we would initially assume that H_{min} will be equal to H_0, the spontaneous curvature of clathrin, we will soon see that the two can (and will) differ in general.

The scale for A is set by $N_{\text{Clathrin}}\, L_0^2$, where N_{Clathrin} is the number of clathrin hubs in the coat and L_0 is the equilibrium leg length. A direct way to choose A is to set it equal to the total node number N_{Clathrin} times a representative of the area occupied by a single clathrin: $A = N_{\text{Clathrin}}\, A_{\text{Clathrin}}$. The value of κ_C calculated as such will be called κ_C^{total}.

However, Equation (3.17) suggests that because of the increase of the average bond number per clathrin we will already see an increase in the effective bending rigidity. If we want to investigate whether there is an additional increase beyond the higher density of the grid, we can compensate for it by also including the average number of bonds per clathrin into the area, i.e., setting $A = N_{\text{Cl}}\, \langle N_{\text{Bonds per Cl}} \rangle\, A_{\text{Cl}} = N_{\text{Bonds}}\, A_{\text{Cl}}$. The value so calculated will be called $\kappa_C^{\text{conformal}}$, since its increase is only due to the conformation of the grid. The area occupied by a single clathrin can be approximated as $A_{\text{Clathrin}} \approx 1.3 L_0^2$. A hexagon with side length a has area $A_{\text{Hex.}} = 3a^2\sqrt{3}/2$, and a single clathrin occupies 3 times 1/6 of a hexagon with side length $a = L_0$.

4.3.2 Differentiating the Zero-Temperature Energy

While calculating the minimised energy of the grid at different curvatures and fitting the Helfrich-law to it is the most direct approach to finding the effective bending energy, it turns out that the algorithm that defines the zero-temperature limit of the energy is stable enough to be differentiated with respect to the curvature, using implicit differentiation, see Section 2.6.3. That this can be done and that it leads to stable, reproducible results is not a given! However, as can be seen in Figure 4.9 B), the gradient of the energy with respect to H follows a linear law, in accordance to the quadratic total energy. A rough estimate yields that both the energy and its gradient predict the same value of κ_C in the given example.

Why is the zero-temperature energy differentiable? The reason for this is that any *finite* curvature change necessarily changes node positions and will almost surely increase the energy of a given configuration. However, we are trying to approximate a continuous curvature change, where the system probably has enough time to adiabatically relax to around an energetic minimum again. Therefore, by sticking to the zero-temperature energy, we are always dealing with a configuration around an energetic minimum and energy changes induced through curvature changes will only depend on the curvature change, and not on node reshuffling, which should resemble the adiabatic relaxation.

This also has another advantage: Differentiating the energy at an extremum of one of its parameters directly links the *partial* derivative to the *total* derivative, as explained in 2.6.3.

Therefore, instead of calculating the energy $E(H)$ at different curvatures and per-

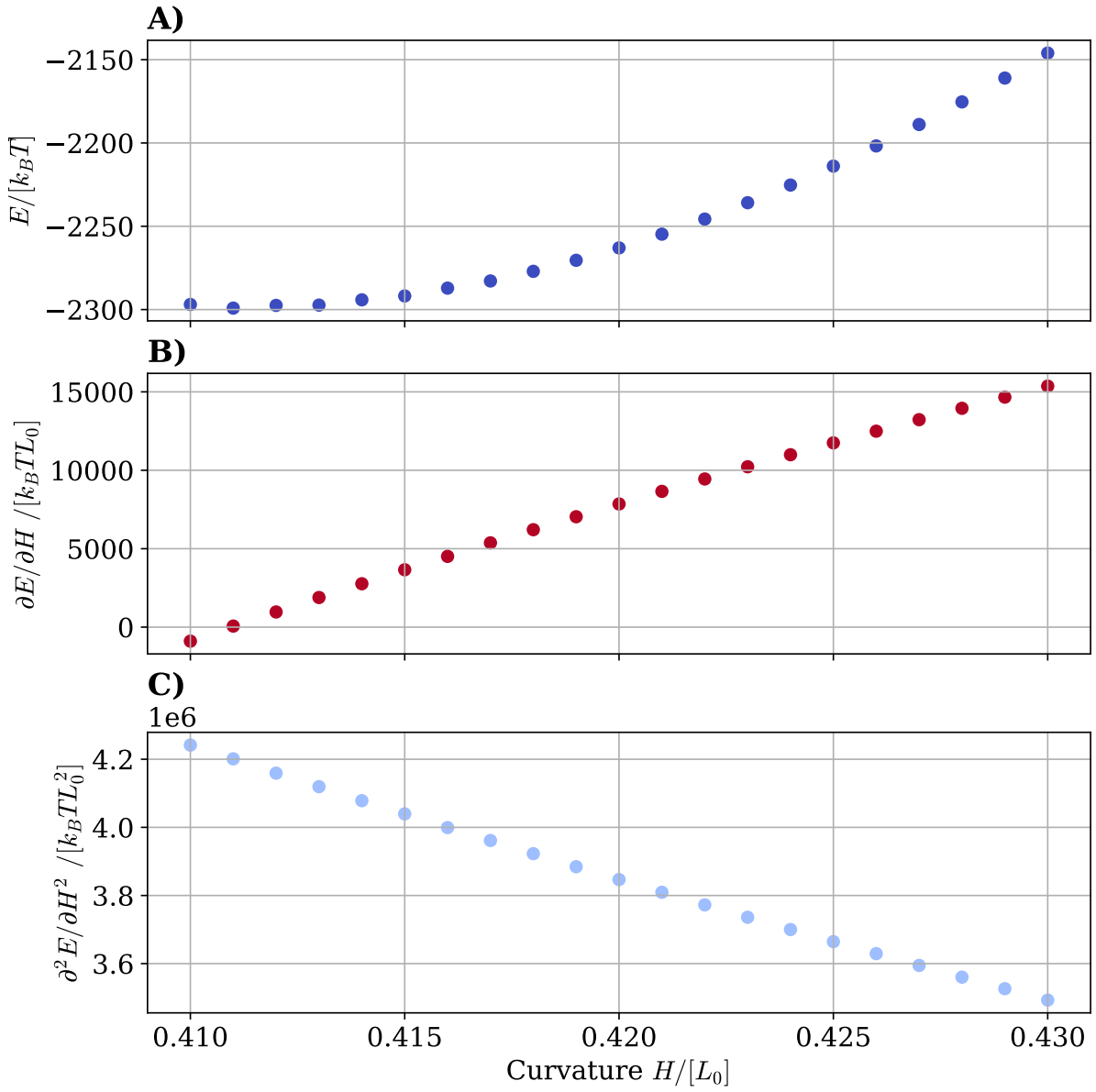

Figure 4.9: Representative examples of bending energy with its derivatives. A) For a clathrin lattice grown at preferred curvature $H_0 = 0.42/[L_0]$, the energy H after minimisation over the node positions is shown. One can see that it presumably follows a quadratic dependency in the curvature H, similar to a Helfrich-like energy. Notably, the energy minimising curvature H_{min} is shifted from the naively expected value of H_0. From the graph, a rough estimation gives $A\kappa_C \sim 3.75 \times 10^5 k_BTL_0^2$. B) The energy from A) can be locally differentiated using implicit differentiation and `autoGrad`. The gradient shows an almost linear behaviour in the curvature H. The slope is proportional to the effective bending rigidity. Again, a rough estimate yields $A\kappa_C \sim 3.75 \times 10^5 k_BTL_0^2$. C) The Hessian of the gradient, calculated again using implicit differentiation, is still stable. However, its values (being equal to twice $A\kappa_C$) overestimate the values from A) and B) by an order of magnitude, with $A\kappa_C \sim 2 \times 10^6 k_BTL_0^2$.

forming a quadratic fit, we can instead calculate the gradient of $E(H)$ with respect to the curvature H at different curvatures and perform a linear fit. This linear fit should be according to

$$\frac{\mathrm{d}E_{\mathrm{eff}}(H)}{\mathrm{d}H} = 2\kappa_C(H - H_{\mathrm{min}})A \tag{4.6}$$

and reveals information about the effective coat bending rigidity κ_C as well as about the energy minimising curvature.

This begs the question: Can we go a step further? Is it possible to calculate the second total derivative of our energy to directly read off the bending rigidity? The second total derivative of our effective energy should independent of H and be expressed as

$$\frac{\mathrm{d}^2E_{\mathrm{eff}}(H)}{\mathrm{d}H^2} = 2\kappa_C A \tag{4.7}$$

and has the advantage that it only needs to be calculated at a **single point**. This can save a lot of computation time. However, we lose information about the energy minimising curvature H_{min}. And we lose some predictive power over the errors of our quantity-estimates. The Hessian of our energy for a representative example is shown in Figure 4.9 C). We see that, while it is not constant, indicating a deviation from a pure quadratic bending energy, it is, however, relatively stable across the curvatures tested. And while it seems that the repeated differentiation procedure produced stable results, the estimated value of κ_C is larger by roughly an order of magnitude than the predictions of the energy A) and the gradient B). Our leading hypothesis is that the implementation of deriving the second total derivative actually only calculates the second partial derivative.[1]

The actual implementation of the differentiation is relatively easy thanks to `JAX`. We simply produce the gradient of the Hessian by:

```
erg_gradient_wrt_curvature = jax.grad(energy_at_curvature)
```

or

```
erg_hessian_wrt_curvature = jax.hessian(energy_at_curvature)
```

This will be our primary method to probe the bending rigidity of different grids.

4.4 Grid Growth at Fixed Curvature

Although we have the option to let the grid curvature evolve dynamically during grid evolution, we will not make use of it for the time being. This makes it easier to distinguish which effects are clearly driven by the curvature, and which are driven by the time evolution (i.e., node inclusion) of the grid. We do this by first examining grids that have been grown at a fixed curvature.

[1] Specifically, the second order total derivative of the effective energy with respect to the curvature does involve derivatives of the microscopic energy with respect to the node positions: $\mathrm{d}^2E_{\mathrm{eff}}/\mathrm{d}H^2 = \partial_H^2 E(\boldsymbol{x}^*(H), H) - (\partial_H \boldsymbol{\nabla}_{\boldsymbol{x}} E) \cdot (\mathrm{Hess}_{\boldsymbol{x}} E)^{-1} \cdot (\partial_H \boldsymbol{\nabla}_{\boldsymbol{x}} E)(\boldsymbol{x}^*(H), H)$. While $\mathrm{d}^2E_{\mathrm{eff}}/\mathrm{d}H^2$ corresponds to the relaxed energy response, i.e. κ_C as measured from the gradient derived using the envelope theorem, the second order partial derivative $\partial_H^2 E$ describes the *unrelaxed* or *frozen energy* response, i.e., how the system's energy changes under changes of curvature when the node positions are held fixed. The second term on the right hand side, involving the Hessian with respect to the node positions, is positive semi-definite and describes the decrease in energy through relaxation of the grid. The difference in the values of κ_C found through the Hessian and the gradient fit is how much bending energy the system can absorb through rearrangements of the lattice.

For this, we have generated a sample of grids grown at $H \in [0.32/L_0, 0.42/L_0]$, i.e., radii $R \in [2.4\,\mathrm{L_0}, 3.12\,\mathrm{L_0}]$. As previously explained in Section 4.2.2, the curvature $H \approx 0.4/L_0$ is one that would allow the growth of a truncated icosahedron. The precise parameters for the simulation run are shown in the appendix in Table C.1.

Since the radii vary, also the approximate number of ideal nodes to be placed on the membrane (see Appendix A.3) varies between $N_{\mathrm{ideal}} \in [56, 95]$. All grids showed a very structured initial growth. Depending how close the growth curvature was to the ideal curvature, the better the closing capability of the coat seemed to be. Some examples of the grown grids are shown in the appendix, see Figures B.2 and B.3. It is, however, hard to say whether the ability to close is increased for grids at $H = 0.4/L_0$ because it is closer to their ideal curvature or because it is closer to the curvature that allows a regular Archimedean solid to grow.

Let us understand the different effects we see for the different curvatures. We will look at average values over an ensemble of grids. For this, we have chosen either time or, in the case of growth at fixed curvature, the node number and have binned the data along it. For every bin, we have then computed an average and the standard deviation from it. Let us start by looking at the growth of the grids in terms of node numbers. This is shown in Figure 4.10. We see that all grids initially grow and then stop the growth once they have reached their final size. As predicted before, the minimal number of final nodes is around 56, while the maximum number of final nodes is around 95. We can also see that our simulation ran long enough for all grids to have reached their final size. Another interesting aspect has something to do with the kinetic Monte Carlo simulation. All grids ran for the same number of evolution steps. However, the larger the grid, the shorter the total amount of time elapsed until all evolution steps have passed. The reason for this is that the time increment Δt by which we evolve time in KMC, see Section 2.5.3, scales inversely with the total number of available escape trajectories, N_s, according to $\Delta t \sim 1/(wN_s)$. However, N_s scales with the total number of nodes in the system. Therefore, larger systems elapse a shorter amount of time during the same number of evolution steps. For an order of magnitude, we see roughly $\Delta t \sim 10^3/[1/\gamma]$ units of time difference for a difference $\Delta N \sim 40$ final nodes, giving a ratio of slow-down of approximately $-25/[1/\gamma\,\mathrm{Nodes}]$ units of time per node in the final lattice.

Next, we can look at the energetics of the growth process. The averaged total energy for the different grids is shown in Figure 4.11. For all grids, we see as expected that the growth process is highly energy driven. When we compare the evolution to Figure 4.10, we see that the energy evolution mirrors the inclusion of nodes, hinting that node inclusion is in this scenario a main driver of the energy cascade. This is to be expected since no curvature change can occur yet. As a rough estimate, the grid with $N \sim 60$ nodes arrives at a final energy of around $E_{\mathrm{fin.}} \sim -2000\,\mathrm{k_B T}$, meaning an average energy per node of $e_{\mathrm{fin}} := E_{\mathrm{fin.}}/N \sim -33\,\mathrm{k_B T}$. As shown in Table C.1, a node with three direct and three NNN-bonds releases a total of $43\,\mathrm{k_B T}$. This means that the difference, around $10\,\mathrm{k_B T}$ or 23% of the total energy, has to be stored as energy of geometric frustration. Looking at the energy per node, shown in the inset of Figure 4.11, we see that all grids equilibrate to an average energy per node of around $\sim -34\,\mathrm{k_B T}$, with grids of lower curvature (being further away from the spontaneous curvature $H_0 = 0.4/L_0$) tending towards lower average energies.

Next, we turn our attention to the polygonal structure of the grown coats. The distributions of penta-, hexa- and heptagons are shown in Figure 4.12 A) and B). We see that almost all coats reach a saturation in the number of polygons, except for the

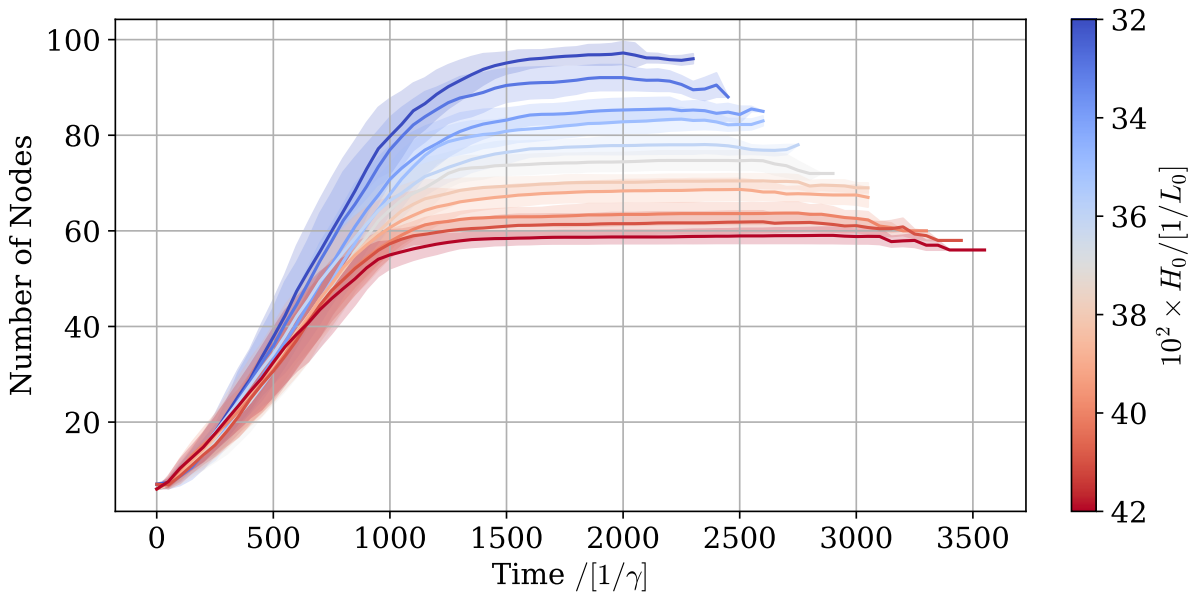

Figure 4.10: Number of nodes during growth at fixed curvature. The number of nodes incorporated into the lattice is shown against time in units of $1/\gamma$. Curvature is indicated by colour. The smaller the curvature, the more nodes are on the final sphere, and the slower the evolution in time for the same number of evolution steps is.

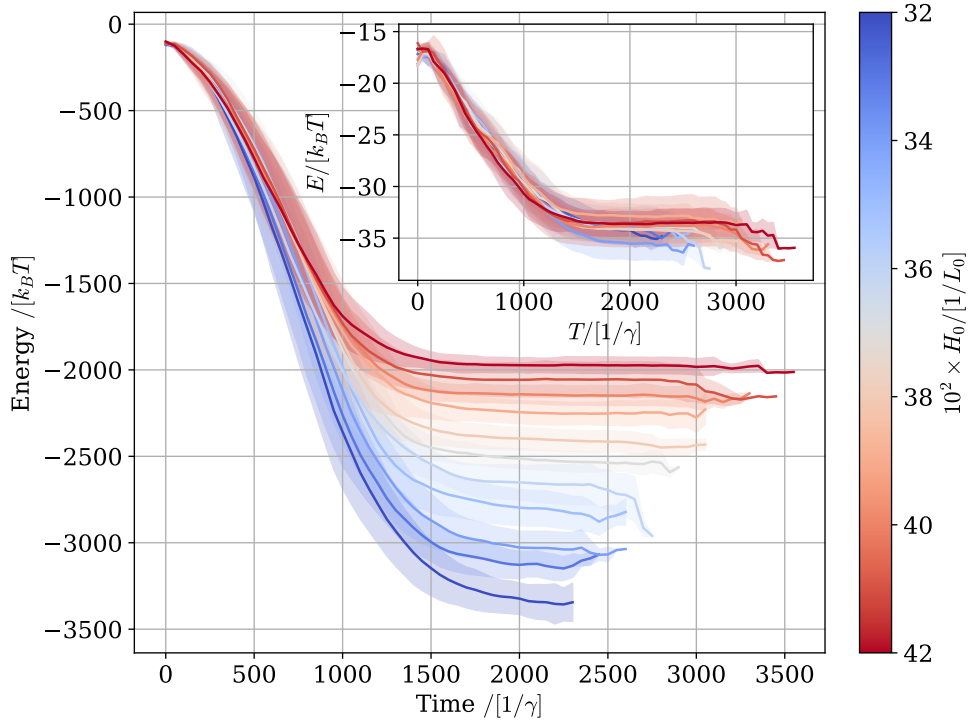

Figure 4.11: Average total energy during growth at fixed curvature. The averaged total energy of grids at different curvature, averaged is shown against evolution time. Curvature is indicated by colour. The inset shows the average energy per node.

ones grown at low curvature (blue). This correlates with the fact that these lattices also have not yet reached a saturation in the number of bonds per node, shown in Figure 4.12 C) and D). We expect the saturation time for the bond number to be later than the saturation time in the node number, and therefore, even though our grids are all at their final number of nodes, the grids at low curvature still have not yet formed all bonds. However, we can still see that grids at lower curvature tend to exceed the expected value of $N_5 = 12$ pentagons. Since for a full sphere, $N_5 - N_7 \stackrel{!}{=} 12$, we would expect that for these lattices, more heptagons are formed. And this is what we can see in Figure 4.12 B), as lattices at lower curvature form up to two heptagons to compensate the additional pentagons.

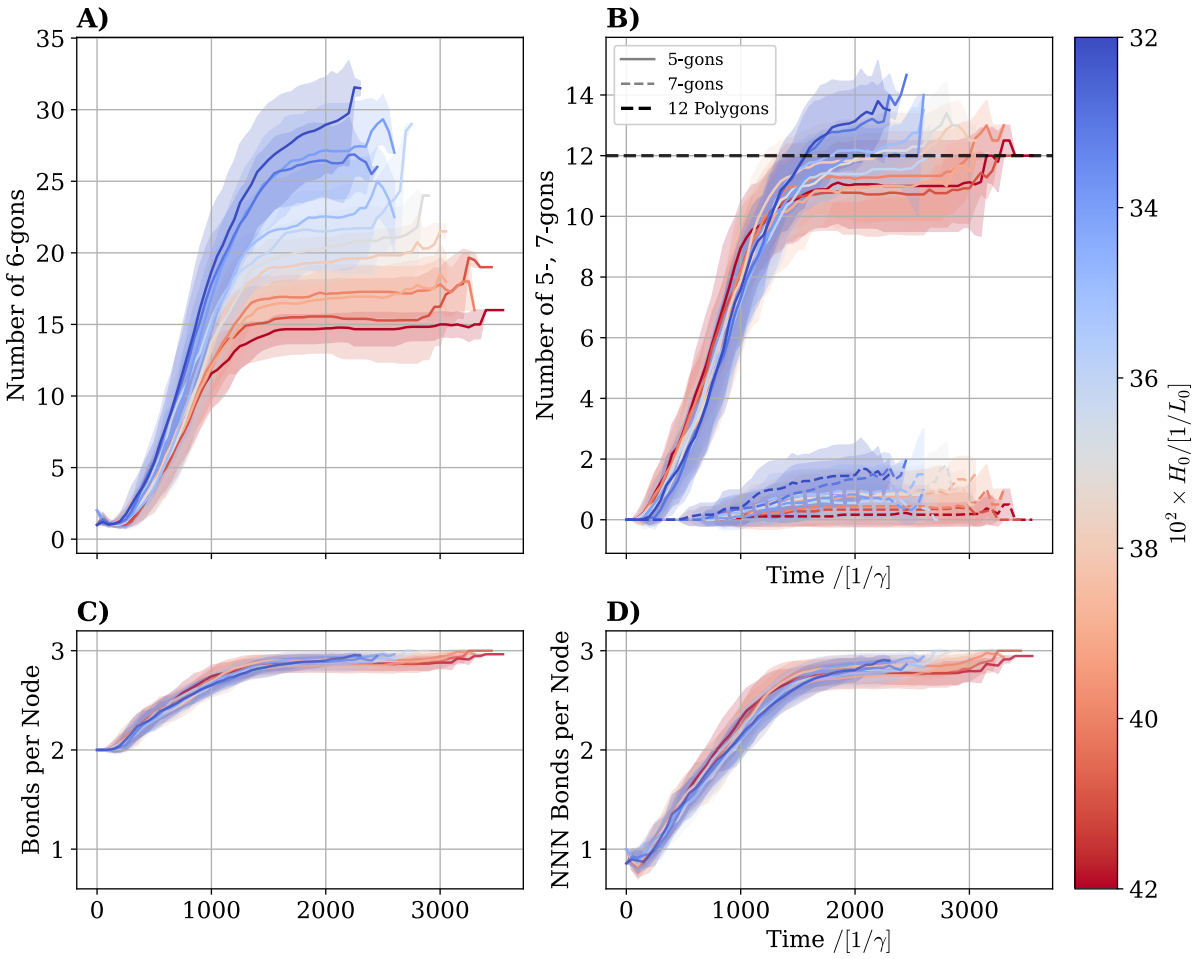

Figure 4.12: The average bond structure during growth at fixed curvature. A) The average distribution of hexagons in the lattices is shown. Curvature is indicated by colour. B) The average distribution of pentagons (solid line) and heptagons (dashed line) is shown. The expected number of pentagons of 12 is indicated by the black dashed line. C) and D) respectively show the average number of direct (C) and NNN (D) bonds per node in the lattices.

We can also use the fact that in our scenario, the node number is monotonically increasing, i.e., can be a surrogate for time. We can then look at the number of polygons as a function of the *fractional node number* n, which is defined as

$$n := \frac{N_{\text{Clathrin}}}{N_{\text{Final}}} \tag{4.8}$$

as the fraction of the current to the final node number. This is shown in Figure 4.13. Looking at A), we see that the smaller the curvature, the steeper the rate at which new hexagons are included into the coat. This makes sense, as a larger coat needs to incorporate more clathrin. If we assume that the total number of non-hexagonal polygons in the coat is restricted, then necessarily a coat at smaller curvature needs to build more hexagons during its evolution, resulting in a steeper increase. This behaviour is different for the pentagons and heptagons, shown in B). We see that the rate of inclusion along the whole growth trajectory is much more similar along the different curvatures. The coats of low curvature are however again a bit quicker in incorporating the non-hexagonal pentagons. However, this might be due to the fact that larger coats have the possibility to better fit heptagons. In Figure C), we correct for the inclusion of heptagons by showing the difference of N_5 and N_7. This is the true value that should approach $N_5 - N_7 = 12$. In C) we best see that the rate of inclusion along the growth trajectory is similar for different curvatures. This again makes sense, since for a full spherical coverage, the lattice needs $N_5 - N_7 = 12$, independent of the size of the coat.

4.4.1 Effective Bending Rigidities at Constant Curvature

As explained in Section 4.3.2, we can use our *zero-temperature energy* to find the *effective bending rigidity* κ_C of our coat. We will for now look at κ_C^{total}, i.e., we define the area of the coat A as the number of clathrins N_{Clathrin} in it times a representative of the clathrin area A_{Clathrin}, such that

$$A_{\text{Coat}} = N_{\text{Cl}} A_{\text{Cl}} \tag{4.9}$$

and by doing so, we will not correct for the increase of κ_C through higher intra-coat connectivity. As mentioned previously, $A_{\text{Clathrin}} \approx 1.3 L_0^2$.

We found that the first differentiation process, finding $\partial_H E$, was quite stable. Calculating it and fitting

$$\partial_H E(H) = 2\kappa_C (H - H_{\text{min}}) A \tag{4.10}$$

to the data reveals the *fitted* value of κ_C and for H_{min}, the energy minimising curvature. However, we want to probe the applicability of simply calculating the Hessian to find κ_C. The calculated value of the Hessian is translated to κ_C according to

$$\partial_H^2 E(H) = 2\kappa_C A \tag{4.11}$$

Figure 4.14 shows the so calculated bending rigidities κ_C and their evolution over time in A). The first striking fact we can observe is that there is a **clear increase in bending rigidity over time!** And this increase is much stronger than the mere increase we expect through the increased interconnectivity of the coat[2], spanning roughly two orders of magnitude. This stiffening seems to be only lightly correlated with the curvature at which the grids are grown. While it seems to be that grids grown at slightly lower curvatures (blue) end up at a slightly increased final value of κ_C, the difference is only small.

Again we can see that the method of calculating the Hessian does not agree with the gradient, but overestimates the slope of the gradient by at least an order of magnitude, probably due to the grid-relaxation compensating for parts of the bending energy. However, we see that the difference in both fitting methods seems to be rather constant. The

[2] As a reminder, this would increase κ_C by a maximum factor of 6/4. However, we see an increase of at least an order of magnitude.

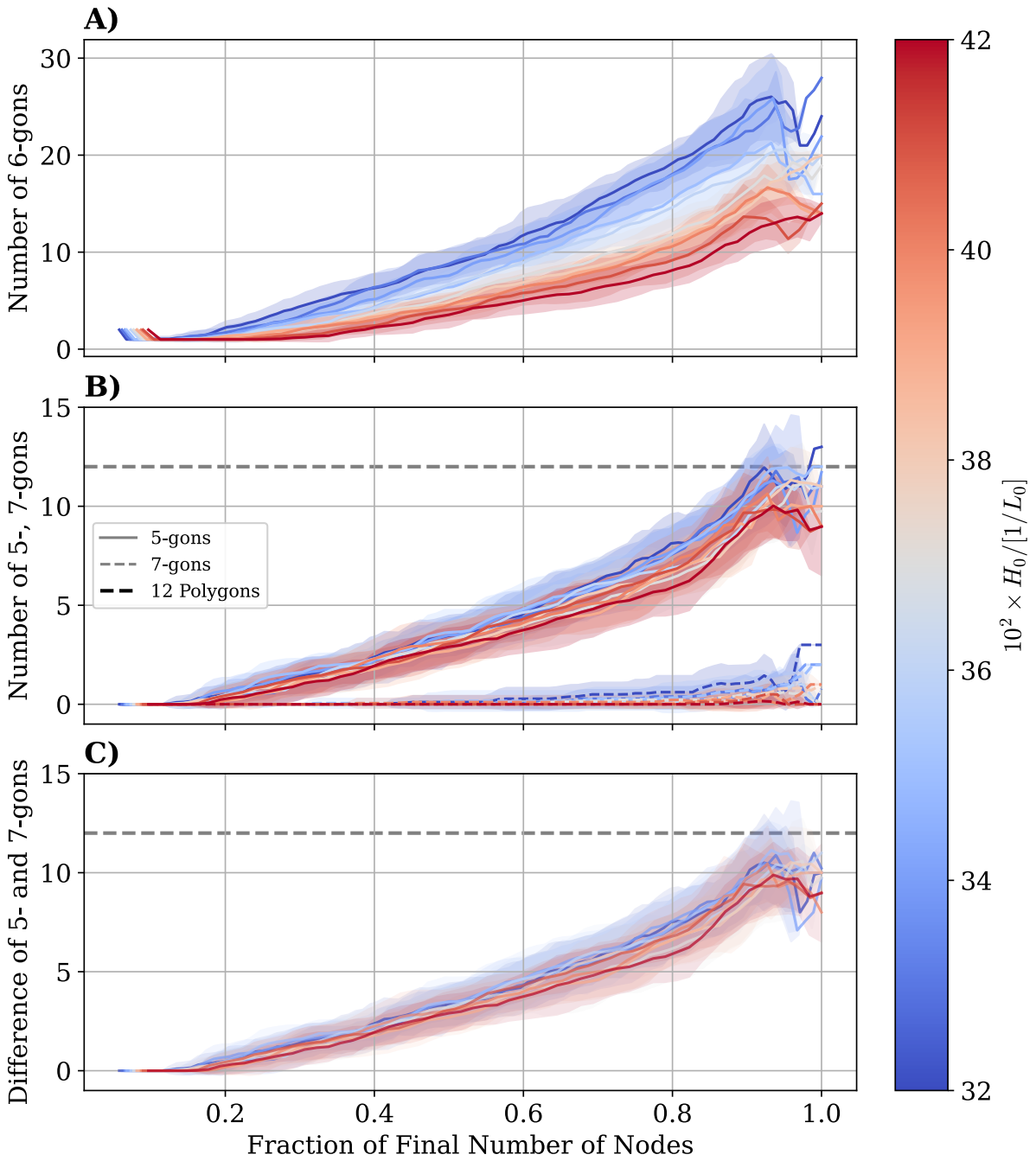

Figure 4.13: Polygon inclusion versus fraction of coat completeness. Shown are the number of polygons, with A) showing the number of hexagons, B) showing the number of pentagons (solid line) and heptagons (dashed line) and C) showing the difference in pentagons and heptagons. In B) and C) the dashed grey line indicates $N = 12$ polygons. Curvature is indicated by colour.

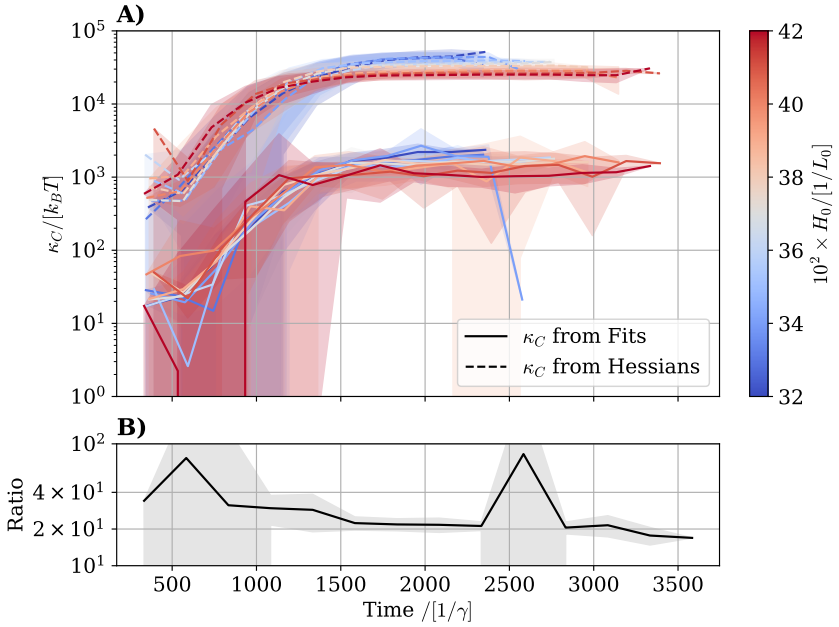

Figure 4.14: Effective bending rigidities over evolution time. A) Shown are the effective bending rigidities κ_C, either found by fitting to the gradient (solid line) or by calculating the Hessian (dashed line). Curvature is indicated by colour. B) The two different methods depicted in A) show an offset by a factor $\kappa_C^{\text{Hessian}}/\kappa_C^{\text{Fit}}$. The average of this factor is shown.

ratio of $\kappa_C^{\text{Hessian}}$ to κ_C^{Fit} is shown in Figure 4.14 B), and while the values of κ_C (for both fitting methods) span multiple orders of magnitude, their ratio seems to be stable around roughly 20. At the current time, we do not yet know why there is this difference in the two methods of calculating κ_C. We can, however, utilize both methods combined. While the fit to the gradient yields the correct numerical values at a low stability of the fit, the Hessian overestimates the slope of the gradient, it is, however, much more stable. Therefore, the combination of both methods can yield a stable shape of the trajectory, while the fit to the gradient yields the correct numerical values.

We can again leverage the fact that the node number can serve as a surrogate for evolution time and look at the effective bending rigidities against the fractional node number, shown in Figure 4.15. A) shows the evolution of κ_C against the total node number, which again shows the increase. If we correct for the final lattice size, i.e., normalize the node number by the final node number, we see in B) that the increase of κ_C seems to be **independent of the curvature** at which the lattice is grown and only **depends on the fractional node number**. This suggests that the contribution of the individual clathrins plays a minor role in the increase of the bending rigidity, but rather the **spherical geometry is the major driver of the rigidity increase**. Taking the fitted values to represent the true value of κ_C, we can estimate that the initial bending rigidity is given by

$$\kappa_C^{\text{init.}} \sim 20\,\text{k}_\text{B}\text{T} \tag{4.12}$$

with the final bending rigidity given by

$$\kappa_C^{\text{fin.}} \sim 1 - 2 \times 10^3\,\text{k}_\text{B}\text{T} \tag{4.13}$$

What determines the size of these effective parameters? For the final bending rigidity, the only parameter with units of energy in that order of magnitude is the leg stretching

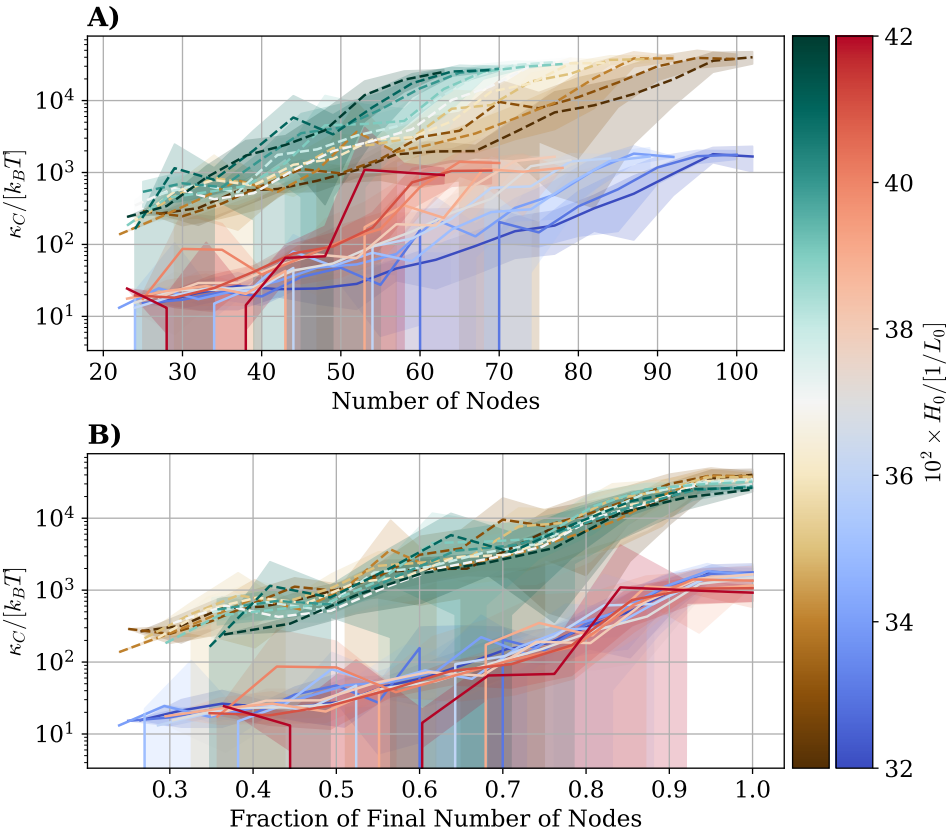

Figure 4.15: Effective bending rigidities against node number. Shown are the effective bending rigidities κ_C calculated from the Hessian (dashed lines) and the fit of the gradient (solid lines). In A), they are shown against the total node number. B) shows κ_C against the fraction of the final node number. For better visibility, both fitting methods are also differentiated by different colour schemes, indicating the curvatures.

stiffness k_L, which is around

$$k_L \sim 800\,\mathrm{k_B T} \tag{4.14}$$

For the initial bending rigidity, the analogous value would be the dipping angle stiffness κ_χ with values of

$$k_\chi \sim 20\,\mathrm{k_B T} \tag{4.15}$$

as we expected from Section 2.1.2. This would mean that initially, bending of the clathrin coat is controlled by bending of the clathrin legs, and therefore the energy scale is given by k_χ. If the coat grows to further completion, the bending is more and more controlled by stretching of the clathrin legs. Finally, if the coat is (almost) closed, a curvature change is only possible through rescaling of the entire coat, and therefore the energy scale is given by k_L.

While we are looking at the effective bending rigidities, we can examine the energy minimising curvature of clathrin H_{min} which minimises the Helfrich energy.[3] Their evolution against time is shown in Figure 4.16 A). We have to remember that all grids have the underlying preferred curvature of the individual clathrin of $H_0 = 0.4/L_0$. Up to some artifacts due to the low sampling size at the beginning of the growth process, we can see that all grids start out at $H_{\mathrm{min}} \approx H_0$. However, over their evolution, all grids eventually

[3] Remember: This is all without a membrane energy to minimise!

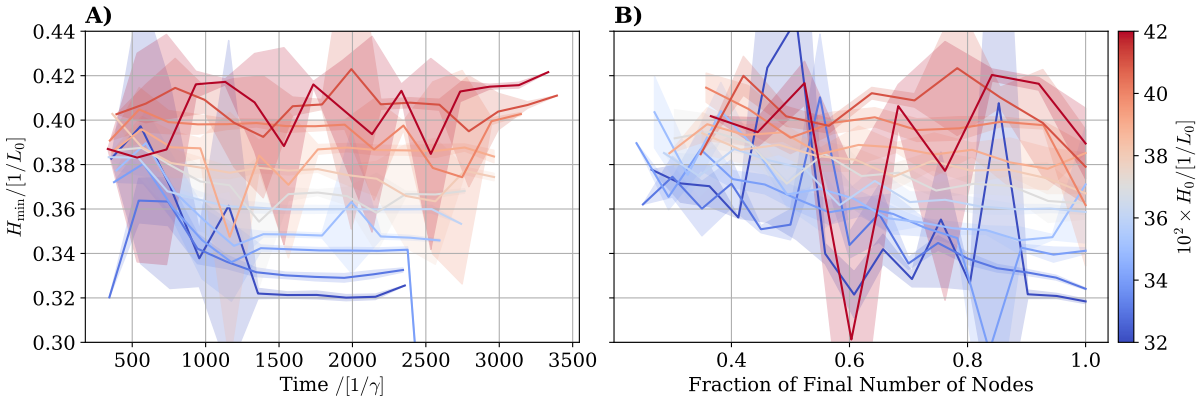

Figure 4.16: Evolution of the energy minimising curvature. Shown is the evolution of the energy minimising curvature $H_{\min}$ against time in A) and against the fraction of the final node number in B). For better visibility, the margins of errors have been halved. The curvature at which the grid was grown is indicated by colour.

approach their growth curvature H_{growth}. This means that the curvature at which a grid grows slowly becomes imprinted into the lattice. At some point the **grid has lost all memory of its microscopically preferred curvature**. This can be related to stalled or aborted clathrin pits. It could be possible that a flat clathrin patch grows, and at some point has "lost" the memory of its microscopically preferred curvature, which is why the clathrin plaque never starts to bend after a certain point. As mentioned by Sochacki et al. (2021), many curved clathrin patches can be found adjacent to larger flat clathrin structures with visible scissions, as shown in Figure 1.6. They also proposed the idea of ruptures in the clathrin coat releasing stored energy of geometric frustration like a Brownian ratchet. Another way to look at this could be that a tightly connected flat structure loses all memory of its microscopic spontaneous curvature. However, a few ruptures could suddenly shift the energy minimising curvature $H_{\min}$ towards larger values closer to H_0, thereby forcing the now separated structure to curve.

In Figure 4.16 B) we can see the evolution of $H_{\min}$ against the fraction of the final node number. Here, the artifacts are larger, probably due to how the data points are binned. However, we will take this to formulate a first phenomenological description of the data.

4.4.2 A Phenomenological Description of the Parameter Evolution

We propose a preliminary model for the parameter evolution at constant curvature. Inspired by this, we will try to transfer our ansatz to the case of variable curvature.

First, we can look at Figure 4.15. As already mentioned, if we define

$$n := \frac{N_{\text{Nodes}}}{N_{\text{final}}} \in [0, 1] \tag{4.16}$$

as the fractional node number, then the evolution of κ_C as a function of n becomes **independent** of the curvature H at which its corresponding lattice is grown. The fractional node number n serves as a measure of how complete a coat has grown. We will explain further below how this fractional node number can be linked to time, drawing a connection to the CoopCM from Mund et al. (2023). Looking at Figure 4.15 B), a first ansatz

to model the evolution of $\kappa_C(n)$ would be a linear dependency in the logarithm such as

$$\ln \kappa_C(n) = \ln \kappa_C^{\text{init.}} + n \left(\ln \kappa_C^{\text{fin.}} - \ln \kappa_C^{\text{init.}} \right) \Rightarrow \boxed{\kappa_C(n) = \kappa_C^{\text{init.}} \left(\frac{\kappa_C^{\text{fin.}}}{\kappa_C^{\text{init.}}} \right)^n} \tag{4.17}$$

giving rise to an exponential growth of the rigidity. As this law is only governed by the initial and final bending rigidities, only variations of these two parameters change the shape of the graph. As explained above, we suspect that $\kappa_C^{\text{init.}} \sim k_\chi$ and $\kappa_C^{\text{fin.}} \sim k_L$, which then define the macroscopic course of the effective bending rigidity from microscopic parameters.

The observation that the behaviour of the parameters becomes independent of the curvature if it is analysed as a function of n seems to be a more universal property. Looking at Figure 4.16 B), while the data is very jagged, a simple phenomenological ansatz to describe the evolution of the energy minimising curvature H_{min} could be

$$\boxed{H_{\text{min}}(n) = H_0 + n \left(H_{\text{growth}} - H_0 \right)} \tag{4.18}$$

We can now try to transfer both equations to a model with a changeable curvature. While Equation (4.17) does not reference curvature explicitly, Equation (4.18) does. The idea that the wrong curvature imprints itself onto the system suggests that the energy minimising curvature H_{min} of clathrin possesses some form of "memory". Therefore, a natural extension of (4.18) would be the introduction of a memory integral over the fractional node number n:

$$H_{\text{min}}(n) = H_0 + \int_0^n F(H_{\text{growth}}(\tau), H_0)\, K(n-\tau) \mathrm{d}\tau \tag{4.19}$$

where $K(n-\tau)$ is the memory kernel. To match the case of constant growth curvature, we can set $H_{\text{growth}}(\tau) = \text{const.}$ and move F outside of the integral. To get a linear dependence in n, we need

$$\int_0^n K(n-\tau)\mathrm{d}\tau \propto n \;\Rightarrow\; K(\tau) = \text{const.}$$

which can be found through differentiation. Then $F(H_{\text{growth}}, H_0) \overset{!}{=} H_{\text{growth}} - H_0$, which gives us

$$\boxed{H_{\text{min}}(n) = (1-n)H_0 + \int_0^n H_{\text{growth}}(\tau)\mathrm{d}\tau} \tag{4.20}$$

To connect to a changeable curvature, we have to look at how the growth curvature H_{growth} depends on the energy minimising curvature H_{min}. From Section 2.1.3, we propose that

$$\boxed{H_{\text{growth}}(n) = \alpha(n) H_{\text{min}}(n)} \tag{4.21}$$

where $\alpha(n)$ is the ratio of the coat bending rigidity κ_C to the total bending rigidity $\kappa_C + \kappa_M$, including the membrane. All relevant formulae are summarised in Table 4.1.

If we allow for a variable growth curvature $H_{\text{growth}}(n)$, i.e., we introduce Equation (4.21), then Equation (4.20) gives us a differential equation for $H_{\text{min}}(n)$:

$$H_{\text{min}}(n) = (1-n)H_0 + \int_0^n \alpha(\tau) H_{\text{min}}(\tau)\mathrm{d}\tau \tag{4.22}$$

$$\Leftrightarrow \frac{\mathrm{d}H_{\text{min}}}{\mathrm{d}n} = -H_0 + \alpha(n) H_{\text{min}}(n) \tag{4.23}$$

(1.)	$\kappa_C(n)$	$=$	$\kappa_C^{\text{init.}} \left(\frac{\kappa_C^{\text{fin.}}}{\kappa_C^{\text{init.}}}\right)^n$
(2.)	$H_{\text{min}}(n)$	$=$	$(1-n)H_0 + \int_0^n H_{\text{growth}}(\tau)\mathrm{d}\tau$
(3.)	$\alpha(n)$	$=$	$\frac{\kappa_C(n)}{\kappa_C(n)+\kappa_M}$
(4.)	$H_{\text{growth}}(n)$	$=$	$\alpha(n)H_{\text{min}}(n)$

Table 4.1: A First Descriptive Model. Summarised are the four equations we propose to describe our observed data.

which can be numerically solved. Together with the evolution of $\alpha(n)$, one can find the overall energy minimising curvature, i.e., the growth curvature $H_{\text{growth}}(n) = \alpha(n)H_{\text{min}}(n)$. The evolution of the four quantities of Table 4.1 is shown in Figure 4.17 for a representative set of system parameters. A more detailed analysis for variations of the parameters used in Figure 4.17 is found in Figure B.4.

We see that the description predicts a curvature increase in the range of the observed values for clathrin-coated vesicles, marked by the shaded area. As a reminder, they are listed in Table 1.1, and a CCV's curvature ranges from $H_{\text{CCV}} \sim 0.05 - 0.3\,\text{L}_0^{-1}$. While the increase of about two orders of magnitude for the clathrin's bending rigidity is essential for driving this process, its precise final value is not that impactful. It is only important that $\kappa_C^{\text{fin.}} \gg \kappa_M$ in order to drive $\alpha \to 1$. The evolution of H_{growth} hints at a flat-to-curved transition.

Interestingly, we see a decrease in growth curvature H_{growth} after n crosses roughly the 60% threshold, because at this point the increase in α cannot compensate for the decrease of the clathrin coat's energy minimising curvature. Basically, if the clathrin's spontaneous curvature is too large, then initially it will drive growth curvature evolution to larger values, which are, however, still beneath H_0. Because of that, the clathrin's energy minimising curvature will further decrease, until it eventually reaches $H_{\text{min}} \approx H_{\text{growth}}$. At this point, α has usually gone over into the saturated growth and cannot compensate the decrease in H_{min} any more, which leads to the decrease of H_{growth}. This decrease in curvature is of course not observed in experiment. Whether this decrease in curvature can, however, be incorporated into the coat is questionable, as at this point the coat has already crossed the equator and has started to close again. We expect it to be much harder at this stage to change the curvature.

We can also draw a direct connection to the cooperative curvature model. Looking at Figure 4.10, we see that during the initial growth of the clathrin lattices, the node number N is approximately proportional to time, up until the final stages of the growth process, where the saturation kicks in. However, the constant of proportionality changes depending on the curvature, i.e.

$$N(t) = \Xi(H)t \tag{4.24}$$

and therefore $N \propto t$. If we assume that for a constant curvature H, the node number N is proportional to the area A, for which we know that $A(\theta) = 2\pi R^2(1-\cos(\theta))$, then the fractional node number can be related to the invagination angle θ by

$$n(\theta) = \frac{A(\theta)}{A^{\text{fin.}}} = \frac{2\pi R^2(1-\cos(\theta))}{4\pi R^2} \;\Rightarrow\; \theta(n) = \arccos(1-2n) \tag{4.25}$$

This behaviour is shown in Figure 4.18 A) and is strikingly similar to the behaviour of θ against the pseudotime in Mund et al. (2023), shown in Figure 4.18 B). It replicates the $\propto \sqrt{t}$ initial growth phase of θ and the sharp increase towards final pseudotime.

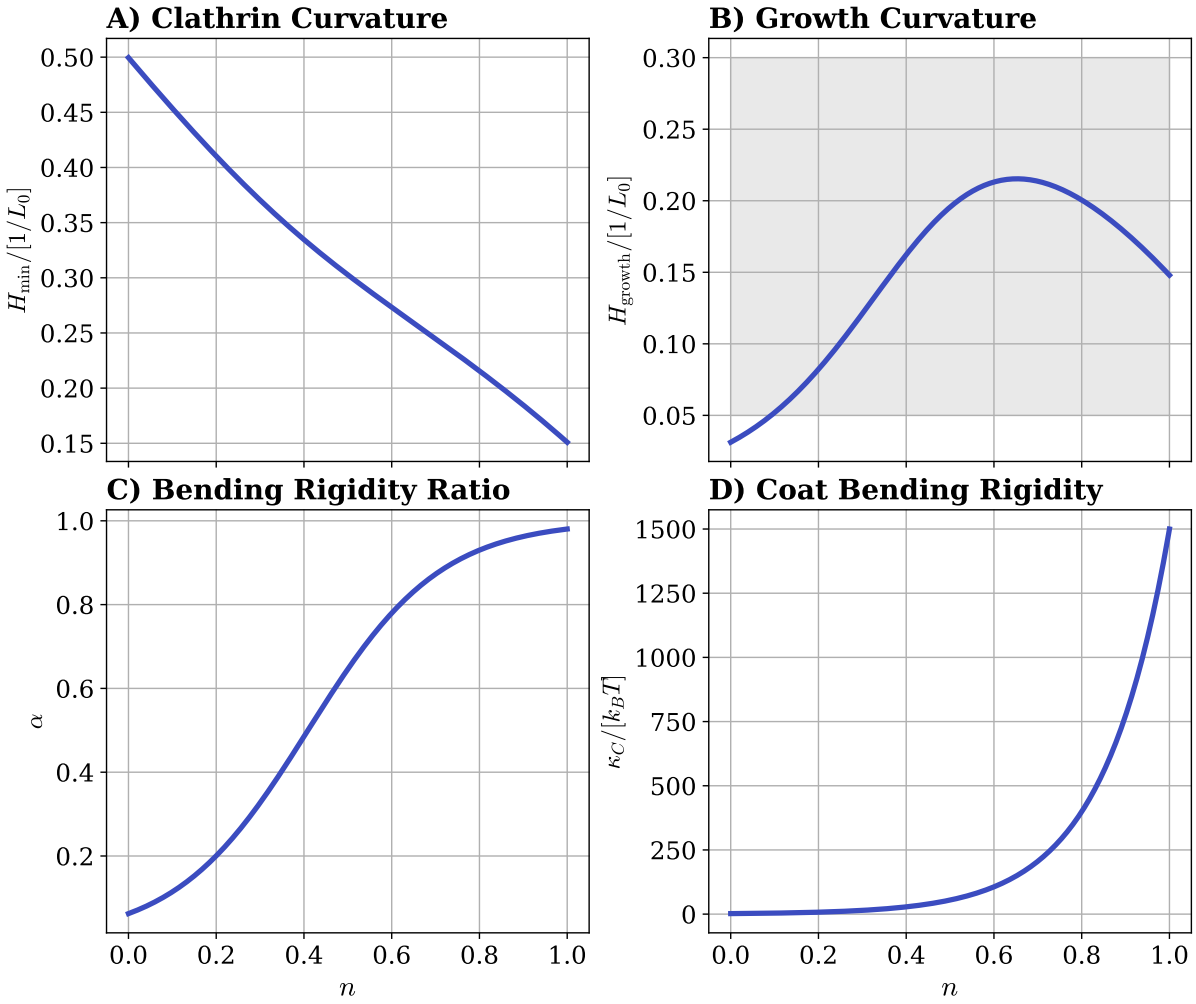

Figure 4.17: Evolution of parameters for variable curvature. The evolution of the four quantities described in Table 4.1 against the fractional node number n for $\kappa_M = 30\,\mathrm{k_BT}$, $\kappa_C^{\mathrm{init.}} = 2\,\mathrm{k_BT}$, $\kappa_C^{\mathrm{fin.}} = 1500\,\mathrm{k_BT}$ and $H_0 = 0.5\,\mathrm{L_0^{-1}}$. A) The evolution of the energy minimising curvature of the clathrin coat. B) The growth curvature of the combined system of membrane and clathrin coat. It is the product of the clathrin coat's energy minimising curvature and the bending rigidity ratio. The grey shaded area marks the found curvatures of clathrin-coated vesicles. We see an increase in curvature, from flat to curved, with a decrease at about 60% of the maturation process. C) The bending rigidity ratio approaches 1 because of the high final coat bending rigidity. The initial value of κ_C determines the starting ratio. D) The evolution of the clathrin coat bending rigidity follows an exponential law towards the final coat bending rigidity.

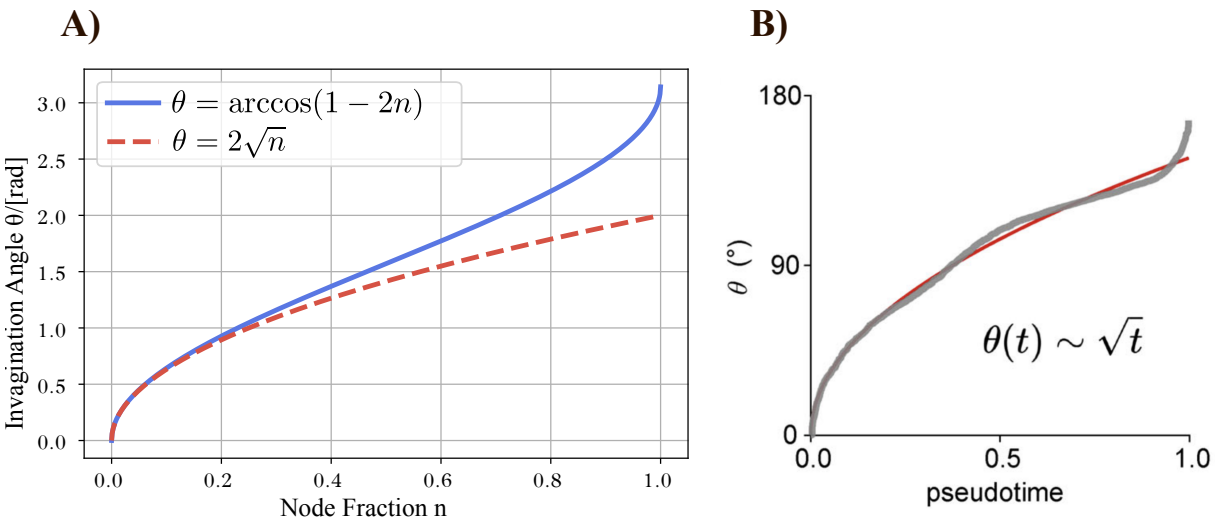

Figure 4.18: Closing angle against fractional node number and pseudotime. A) The evolution of the invagination angle θ follows an arccos dependency on the fractional node number n. Close to small n, this is approximated by a $\sqrt{n}$ plot. B) The invagination angle θ as a function of pseudotime for experimental data (grey) and from the CoopCM (red). Figure taken from Mund et al. (2023).

4.5 Grids at Variable Curvature

Finally, we present a preliminary outlook on grid growth at variable curvature. While these results require further statistical validation, they provide valuable insights into future research directions. To achieve this, we include the membrane energy into the system's total energy and allow for curvature changes as described in Sections 3.4 and 3.7. As mentioned in the latter one, an important parameter for the implementation of curvature change is the maximal curvature change $\Delta H_{\max}$ we allow at every Monte Carlo step. Empirically, we observed that for $\Delta H_{\max} \sim 1 \times 10^{-4}\,\mathrm{L}_0^{-1}$ the curvature evolution seemed stable. For too small values, the curvature simply does not change by relevant amounts. For too large values, the curvature performs a random walk which does not seem to be influenced by a changing energy-minimising curvature. However, further analysis is required to confirm that the precise value of $\Delta H_{\max}$ does not have an influence on the average evolution of the curvature.

For 10 systems with identical initial conditions, the evolution of the total energy, the node number, the growth curvature and the number of penta-, hexa- and heptagons over time is shown in Figure 4.19. The precise evolution parameters can be found in Table C.2.

The most striking observation to note in Figure 4.19 C) is that **curvature generation is observed.** More specifically, we seem to observe a **flat-to-curved** transition of the curvature, starting out at $H^{\text{init.}} \approx 0$, staying around that region up until the $T = 0.5\,\gamma^{-1}$ mark, and then increasing to a value of $H^{\text{fin.}} \approx 0.22\,\mathrm{L}_0^{-1}$. The initial and final curvature observed seems to agree with the minimal and maximal growth curvature predicted in Section 4.4.2, although the evolution does not necessarily match.

Looking at Figure 4.19 A) and B), one can observe that the grid growth is not yet completed. The resulting grids contain many more nodes (from 170 to 300) than our previous grids (between 60 and 100), and therefore the necessary number of evolution steps required to observe the growth of a completed cage increases from $N_{\text{evol}} \sim 6 \times 10^3$

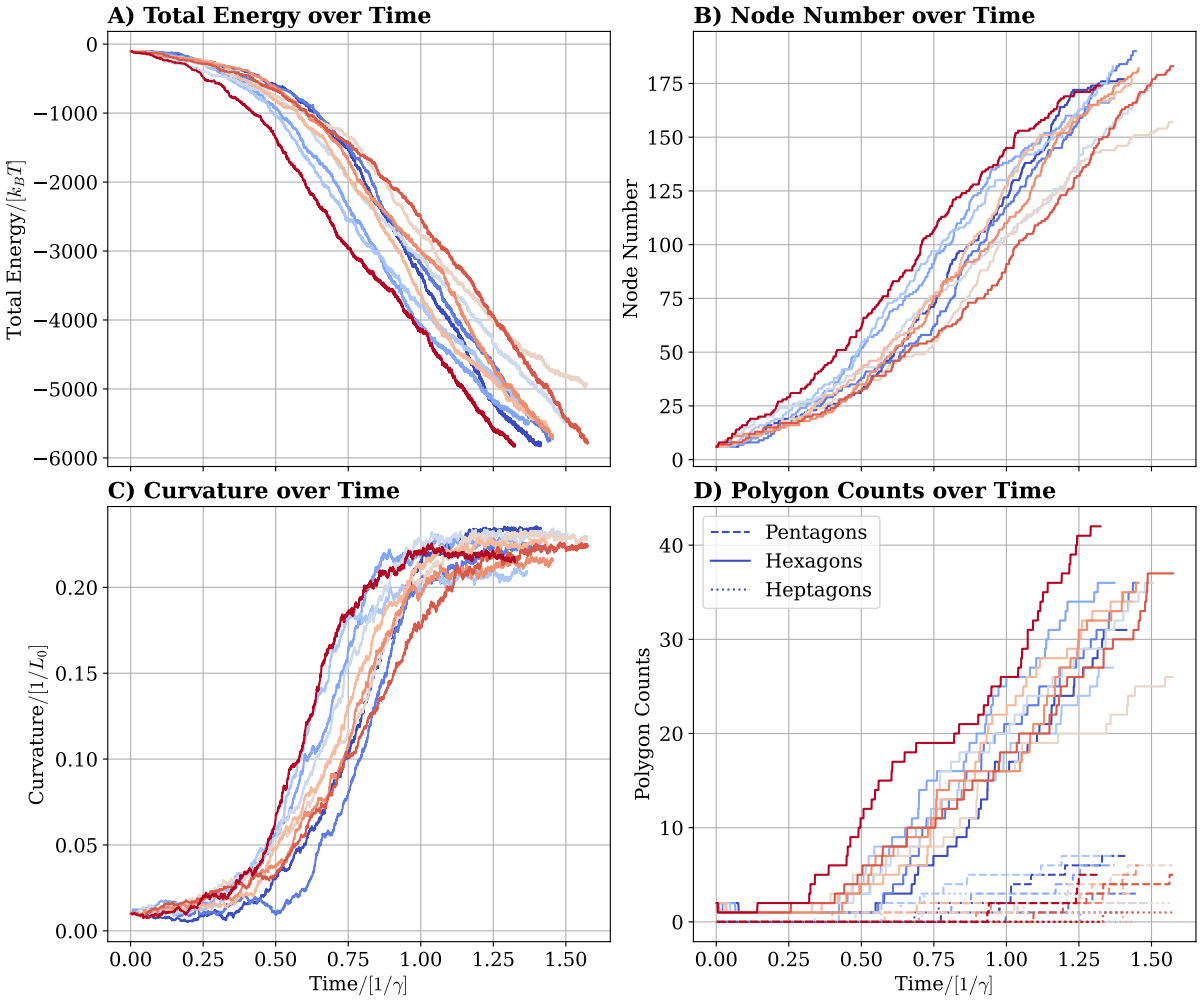

Figure 4.19: Parameter evolution for grids at variable curvature. A) The total energy of the clathrin coat and the underlying membrane. B) The number of clathrins incorporated in the clathrin coat. C) The growth curvature over time. D) The number of polygons, with pentagons (dashed), hexagons (solid) and heptagons (dotted). Different colours label different realisations of the system trajectory.

to $N_{\text{evol}} \sim 2 - 3 \times 10^4$. Looking at the final curvature of $H^{\text{fin.}} \approx 0.22\,\text{L}_0^{-1}$, we would expect the sphere to fit $N_{\text{id}} \approx 200$ nodes, see Appendix A.3. At the time when the curvature generation stops, around $T = 1.0\,\gamma^{-1}$, the grids have only incorporated around $N = 100 - 150$ nodes, so between 50% and 75% of their final expected node number. This indicates that **curvature generation neither starts immediately after the initial growth phase nor does it end only when the cage closes!** There rather seems to be an **intermediate regime** where the coat can **generate enough force** to bend the coat-membrane system and is **still flexible** enough to be bent.

Looking at Figure 4.19 D), one can observe that the onset of curvature generation coincides with the onset of pentagon inclusion, with the pentagon generation being possibly a bit delayed. A higher temporal resolution could clarify the causal relationship: do pentagons "sense" curvature (appearing as a reaction), or do they drive its generation?

Finally, we analyse three systems that, due to delayed curvature onset, reached the imposed node limit. While initially regarded as computational failures, the resulting node deletion events potentially served as unintended scission experiments. The data generated from these three anomalous trajectories shows flat growth of clathrin lattices for identical initial conditions as the previous simulations. Also, it might suggest a key mechanism in which the release of accumulated lattice frustration energy drives curvature generation.

The evolution of the grid parameters is shown in Figure 4.21. Looking at C), we see that all three of the grids have a very long period of *no curvature generation*. Because of that, the rate of node inclusion was much higher for these grids[4] resulting in them hitting the upper node limit of $N_{\text{node}}^{\text{max}} = 220$ for this batch of simulations.[5] Hitting this limit means that the 221st node that is placed is identical to a node that is already in the grid, essentially removing this previously existing node from the grid. Therefore, although not intended, these three failed simulations might be examples of the effect of sudden scissions in the clathrin coat. Looking at the light blue line, we see no curvature generation at all until the $N_{\text{node}} = 220$ mark is hit. As soon as a node is removed from the grid, the curvature suddenly increases. For the dark blue and orange line, while there eventually is curvature generation, it can still be observed that as soon as the first node is removed from the grid, the curvature suddenly starts to increase again.

These observations can be linked to physical phenomena found in CME. We have observed that with identical initial conditions it is possible that some grid configurations do not start to generate curvature. From our previous investigation at fixed curvature, see Section 4.4, we can speculate that after some time the flat curvature has imprinted itself onto the clathrin coat, making curvature generation increasingly difficult the later the onset is. This can be linked to the observation of clathrin plaques, large flat regions of clathrin coat, often found adjacent to maturing CCVs (Heuser 1980; Heuser 1989; Larkin et al. 1986). Some authors speculate (Tagiltsev et al. 2021; Sochacki et al. 2021) that ruptures in flat clathrin lattices release energy, formerly stored as geometric frustration, that drives curvature generation. Our observations might indicate that indeed a few ruptures (here in the form of node removals) lead to subsequent curvature generation.

[4] This is because on a quasi flat surface, there is simply more space for new nodes to bind to the grid.

[5] This upper limit exists to speed up the simulation. `JAX` requires fixed shapes for optimisation, therefore the node list has to have a fixed size. All nodes are created in the beginning, and then subsequently activated during coat growth.

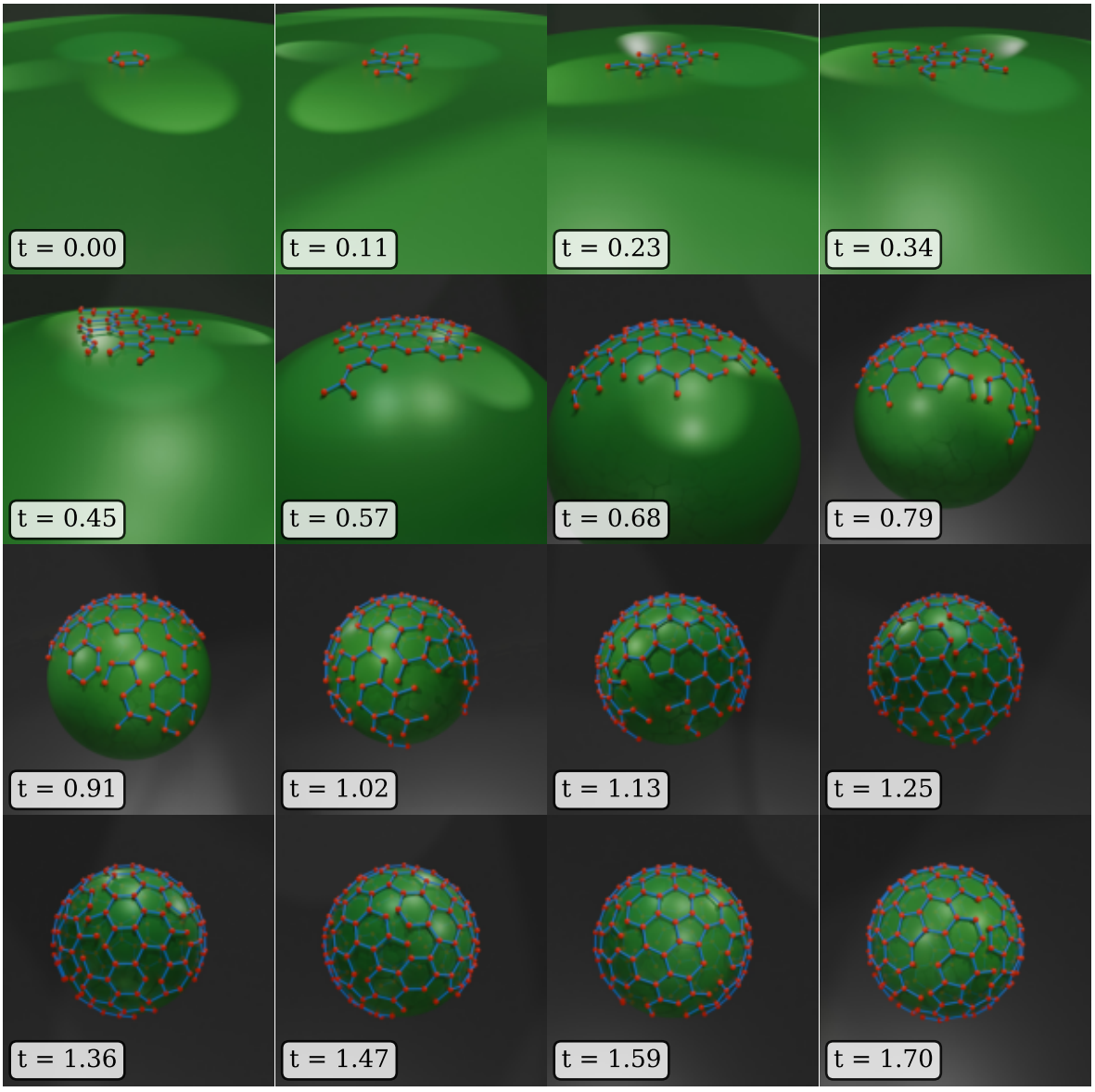

Figure 4.20: The 3D evolution of a curved grid. Shown is the 3D visualisation of a grid showing the flat-to-curved transition from the data set of Figure 4.19. The evaluation time is shown in units of $1/\gamma$.

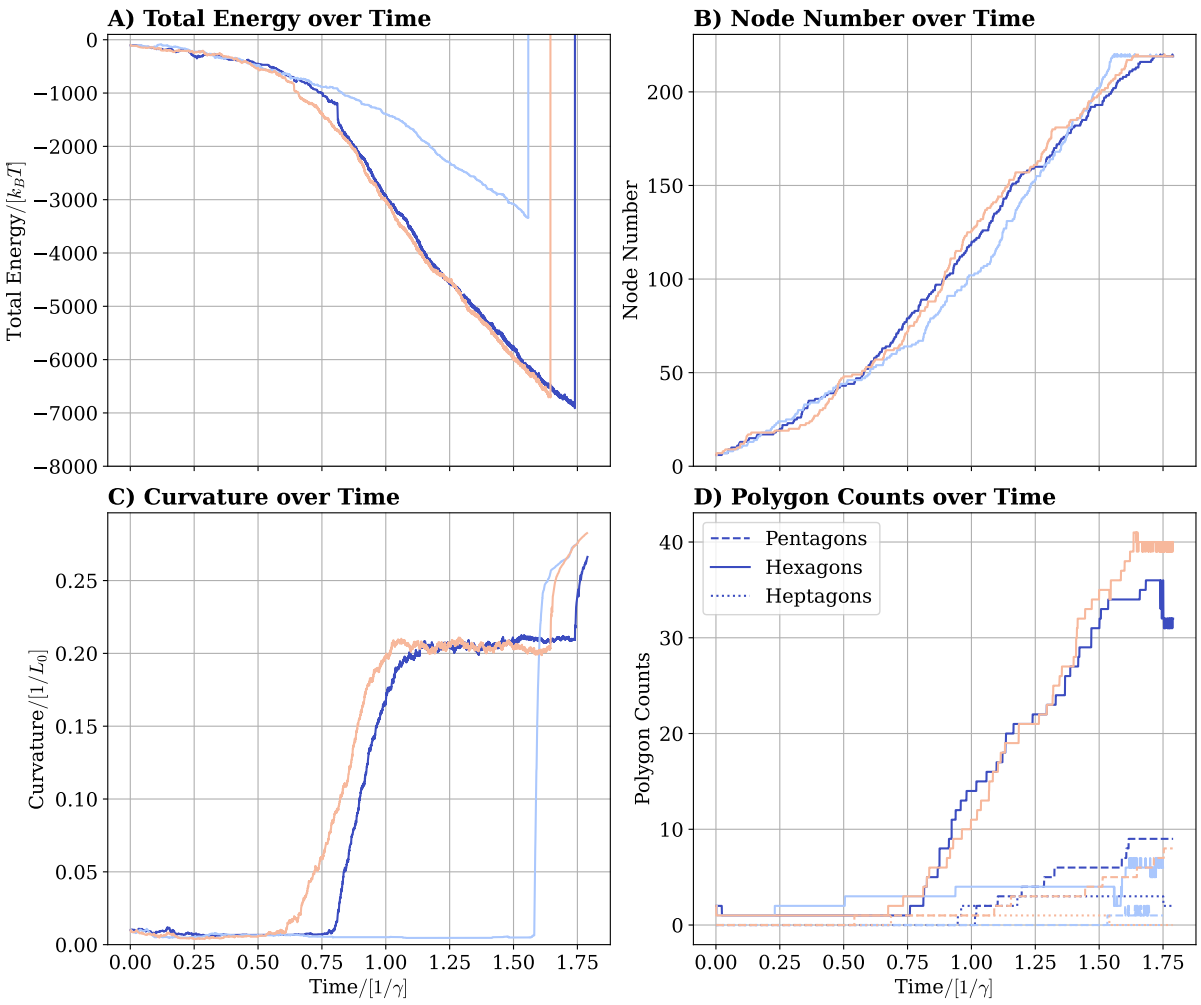

Figure 4.21: Parameter evolution for stalled curvature onset. Similar to Figure 4.19, the parameter evolution for three grids is shown with the total energy A), the node number B), the growth curvature C) and the number of polygons in D). These three simulations show delayed or absent curvature generation, eventually leading to the deletion of nodes once the maximum number of nodes in the system is reached.

Chapter 5

Conclusion and Outlook

5.1 Summary of Findings

In this thesis, we developed a model to simulate the dynamic assembly of clathrin on a deformable membrane, aimed at capturing the essential dynamics of clathrin-mediated endocytosis. The model consists of abstract clathrin units capable of binding to a spherical surface and forming bonds between one another. The dynamics include lateral diffusion of clathrin, the formation and dissociation of bonds and the evolution of membrane curvature. A fundamental distinction is established between state transitions driven by specific physical rates—such as clathrin binding and bond formation—and those arising from equilibrium thermal fluctuations. This kinetic perspective is consistent with established continuum models, where time evolution is introduced by relating the rate of area growth to the coat's circumference ε:

$$\frac{\mathrm{d}A}{\mathrm{d}t} \propto \varepsilon$$

This formulation captures the physical intuition that new clathrin is incorporated exclusively at the boundary. Our approach represents the discrete, microscopic realization of this continuum principle: rather than imposing a global growth rate, the time evolution in our model emerges naturally from the kinetic process of individual node additions at the lattice rim.

The dynamics is purely energy driven. To capture emergent behaviour, we formulated the Hamiltonian governing clathrin locally, only considering the clathrin's legs and its immediate environment. We showed that some emergent macroscopic properties, like a preferred curvature, can directly be derived from the microscopic theory. However, other macroscopic quantities are not yet derivable. One of our goals was to investigate whether such properties, and their time evolution, can be extracted from the clathrin coat coupled to the membrane.

A first simulation demonstrated that, under appropriate geometric conditions, the dynamic assembly of nodes converges towards highly regular polyhedra, such as the truncated icosahedron. In these generated lattices, local deviations from the ideal polyhedral structure were identified as high-energy defects, confirming that the formation of these regular structures is strongly energetically driven. This is a first indicator that the highly ordered global structure is regulated through the local microscopic interactions.

While on average, global properties can be established, in their detail they depend on the microscopic arrangements of the coat, i.e., its microstate. To remedy this, we defined

an energy functional dependent only on the curvature rather than the full microstate. This effective energy not only exhibits the characteristics of a Helfrich-like Hamiltonian in terms of the curvature H but also proved sufficiently smooth to be differentiable. By combining the gradient and Hessian information, we were able to calculate a stable evolution of the effective bending rigidity κ_C. We saw that for grids grown at a fixed curvature, the bending rigidity increased by two orders of magnitude during the growth process. While others have speculated on such a mechanism driving curvature generation (Mund et al. 2023), our observation confirms that such a stiffening can occur purely through the clathrin assembly. We hypothesise that the two scales for the rigidity evolution are set by the angular stiffness k_χ and the stretching stiffness k_L. In the beginning, the clathrin coat is very flexible, and curvature changes result mainly in changes of the leg angles. Therefore, the macroscopic bending energy is controlled through the leg angle energy, and therefore through k_χ. In the later stages of the evolution, curvature changes cannot be accommodated without stretching larger parts of the lattice. At this point, the energy of bending is dominated by the stretching stiffness k_L of the legs.

We determined that this lattice stiffening scales with the degree of surface coverage, independent of the growth curvature—a finding that supports our previous hypothesis. This stiffening was accompanied by a shift in the energetically preferred curvature: the lattice drifted away from the intrinsic preferred curvature of individual triskelia and towards the growth curvature. Effectively, the growth geometry was mechanically imprinted onto the system, causing the clathrin coat to "lose memory" of its microscopically preferred curvature. We propose that this loss of curvature memory, combined with the progressive stiffening of the lattice, constitutes a mechanism underlying stalled, flat growth events. Crucially, since our model reproduces both productive curvature generation and stalled states within the same physical framework, we conclude that the necessary ingredients for these distinct dynamical regimes are inherent to the assembly process itself as we model it.

Leveraging the data gathered at constant growth curvature, we developed a phenomenological description of the evolution of the coat's bending rigidity and preferred curvature. When extended to the dynamic growth regime, this model successfully predicted a curvature increase consistent with biologically measured values. Notably, parts of our formulation align with existing phenomenological descriptions derived from experimental observations. This demonstrates that the idealized case of static growth curvature can serve as a valuable theoretical tool for isolating and extracting fundamental physical behaviours governing assembly and stiffening.

In the fully dynamic regime, our model predicted two distinct outcomes: a spontaneous flat-to-curved transition and stalled, flat growth. The successful simulation of the flat-to-curved transition implies that intrinsic clathrin mechanics provide a sufficient mechanism for curvature generation. We propose that the specific evolution of the growth curvature is determined by a competition between the bending rigidities of the membrane and the maturing clathrin coat. As the latter evolves, the adopted curvature shifts towards higher values. Furthermore, the emergence of stalled, flat lattices—also observed **in vivo**—strengthens our conclusions. The fact that our model naturally reproduces both productive curvature generation and flat patch formation suggests that clathrin mechanics alone are sufficient to explain the diverse structural outcomes found in nature.

Let us contextualize our findings within the existing literature. Mund et al. (2023) presented the cooperative curvature model (CoopCM), for which Frey and Schwarz (2024) subsequently provided a more fundamental derivation. An essential component of this

derivation was the stiffening of the clathrin coat. In that work, the bending rigidity, $\kappa_C(H)$, was defined via an effective dependency on the curvature, evolving as:

$$\kappa_C(H) = \kappa_C^{\text{fin.}} \left(\frac{H}{H_0} \right)^3 \tag{5.1}$$

where $\kappa_C^{\text{fin.}} \sim 300\,\text{k}_\text{B}\text{T}$. Similarly, in Section 4.4, we observed an increase in bending rigidity. However, our analysis utilized the fractional node number, n, as the evolution parameter, yielding:

$$\kappa_C(n) = \kappa_C^{\text{init.}} \left(\frac{\kappa_C^{\text{fin.}}}{\kappa_C^{\text{init.}}} \right)^n \tag{5.2}$$

with $\kappa_C^{\text{init.}} \sim 20\,\text{k}_\text{B}\text{T}$ and $\kappa_C^{\text{fin.}} \sim 1 \times 10^3\,\text{k}_\text{B}\text{T}$ (compare Table 3.2). Both frameworks describe an increase in bending rigidity. While the final values differ by a factor of 3, the precise magnitude of $\kappa_C^{\text{fin.}}$ is less critical than the condition $\kappa_C^{\text{fin.}} \gg \kappa_M$, which is satisfied in both cases. Our description is valid for static curvature; however, the applicability of this model to dynamic curvature remains speculative. It may be possible to reconcile the equations from Table 4.1 with the curvature evolution described in Equation (5.1), though we have not yet achieved this derivation.

The question of whether CME proceeds via the constant area model (CAM), the constant curvature model (CCM), or a hybrid of both is a subject of intense debate in the literature. The curvature evolution we observed in Figure 4.19 resembles a flat-to-curved transition. The coat initially grows flat, subsequently generates curvature parallel to its growth, and finally ceases curvature generation to complete the final $\approx 25\%$ of growth at a constant curvature. While the initial phase resembles the constant area model (no initial curvature) and the final phase resembles the constant curvature model, the majority of the growth process—which is most critical for determining the final curvature—is not adequately described by either model. Here, an interpolation such as the cooperative curvature model appears most appropriate. It predicts an onset of curvature only after the coat has reached a size:

$$A_{\text{onset}} = \frac{\pi}{\gamma^2} \tag{5.3}$$

with $\gamma = 0.147 - 0.239\,\text{L}_0^{-1}$, resulting in $A_{\text{onset}} = 54 - 145\,\text{L}_0^2$. Given an average area per clathrin of $A_{\text{Cl}} \approx 1.3\,\text{L}_0^2$, this corresponds to an onset of curvature generation at $N_{\text{Cl}} = 42 - 112$ nodes. This range is consistent with, albeit slightly higher than, the node numbers we observed for the onset of curvature generation.

The role of fractures within the clathrin coat is another contested topic. Some authors, such as Sochacki et al. (2021) and Den Otter et al. (2011), argue that clathrin coats assemble loosely to facilitate bond fractures, thereby releasing geometric frustration energy, acting like a Brownian ratchet. Our observation that identical initial conditions can lead to either a flat-to-curved transition or stalled, flat patches suggests that the latter may harbour significant frustration energy. The observed curvature increase following node deletion reinforces the notion that scissions can act as a driver for curvature generation if the initial activation threshold is missed. This suggests two parallel mechanisms: either clathrin patches generate curvature following a brief initial growth phase at flat curvature, or, if this window is passed, the patch grows as a flat plaque until a scission event releases sufficient energy to drive curvature generation.

5.2 Critical Assessment

In critically assessing our results, several caveats and limitations must be addressed. To accelerate computation, we employed an oversampling technique. This approach allows certain kinetic processes to be accepted with a probability $P > 1$, causing them to be effectively underrepresented in the final simulated dynamics. Although we endeavoured to keep the fraction of such events low, further analysis—specifically, comparing these results against longer simulations with minimised underrepresentation—is required to verify the validity of the observed dynamics.

When utilizing our zero-temperature energy to determine the bending rigidity of the clathrin coat, we observed that the calculated Hessian overestimates the gradient slope by a factor of approximately 20. We speculate that this discrepancy is caused by the relaxation of the grid, compensating large parts of the bending energy through reshuffling. However, this hypothesis needs to be further verified. We proceeded to use the Hessian to determine the shape of the parameter trajectories, noting that this scaling factor appeared relatively constant. However, identifying the physical or numerical source of this factor and confirming that the Hessian accurately reflects the true parameter evolution remains a necessary step.

Regarding the curvature-changing algorithm, the influence of the maximum curvature change, $\Delta H_{\max}$, is not yet fully resolved. While we empirically identified values of $\Delta H_{\max}$ that yield plausible curvature evolution, it is crucial to ensure that the specific choice of this parameter introduces no artifacts into the curvature trajectory or the time evolution of the system.

Furthermore, the sample size for grid growth under dynamic curvature changes was limited by computational constraints. Consequently, the current results should be interpreted as indicative rather than definitive. To draw robust statistical conclusions, larger sample sizes spanning a wider range of parameters are required.

Additionally, our model assumes zero thickness for both the clathrin and membrane layers. Physically, however, both possess finite thicknesses of approximately $0.25\,\mathrm{L}_0$. Moreover, a connecting layer resides between them with a thickness of roughly L_0. These geometric offsets imply that the membrane curvature is higher than the clathrin curvature. Future iterations of the model would benefit from incorporating these finite-thickness effects into the membrane energy formulation.

Finally, our parameter estimates—such as the leg stretching stiffness—rely on the assumption that clathrin legs can be modelled as elastic, unbendable rods, except at the joints. Physiologically, however, individual clathrin legs appear to exhibit greater flexibility. Consequently, the values utilized in Table 3.2 may require re-evaluation or exploration over a wider range. In particular, the stiffness of the single clathrin leg pucker angle, k_χ, could be significantly lower than estimated, around $k_\chi \sim 1\,\mathrm{k_B T}$, as discussed in Appendix 2.2.

5.3 Outlook

A primary avenue for future research lies in elucidating the precise physical determinants of the clathrin coat's effective bending rigidity. While we hypothesised that the order of magnitude is governed by the leg stretching stiffness, k_L, the quantitative mismatch suggests more complex interactions. Furthermore, we observed that the initial value of κ_C exerts a measurable impact on the initial growth curvature, shifting it toward $H = 0$

for lower values of $\kappa_C^{\text{init.}}$. It remains to be determined whether κ_C originates at a finite value or vanishes at zero curvature, as proposed by Mund et al. (2023). To clarify this, the analysis of the evolution of κ_C, which we performed for fixed curvatures, should be extended to the regime of grid growth at variable curvatures. Additionally, while our current geometric description relies on the dipping angle χ and the projection angle ϕ, a more natural description might involve the total intra-leg angle θ. Investigating whether a rigorous mapping between these frameworks can be established—and whether they yield consistent results—would be a valuable theoretical step.

Beyond static properties, the dynamic interplay between topological defects and curvature generation warrants deeper investigation. While pentagon inclusion scales with the fractional node number for static curvature, the dynamics under variable curvature remain unclear. Key open questions include whether defect inclusion precedes or follows curvature generation, what governs the specific inclusion of heptagons, and what defines the onset of curvature generation—specifically, whether it is triggered by a critical node number or a threshold of frustration energy. Connecting these dynamics to our observations in Section 4.2.2, where deviations from regular polyhedra were visible in node energies, suggests that targeted removal of high-energy nodes could model the curvature transition. This hypothesis could be tested by combining targeted deletion with general scission experiments, building upon the findings in Section 4.5, to determine if node removal acts as a driver for the flat-to-curved transition.

Finally, from a methodological and broader theoretical perspective, several avenues for optimization and comparison exist. The current implementation of curvature changes—specifically regarding the timing relative to the Monte Carlo steps in node-diffusion—remains an open area for refinement. Implementing the alternative curvature update scheme discussed in Section 3.7, or the alternative membrane bending energy mentioned in Section 3.5.4, could yield more stable or spherical clathrin patch growths. Furthermore, it is essential to reconcile our findings with the typical growth models, like the Eden growth model, used to describe clathrin assembly. We must also verify if the curvature memory effect observed for static curvatures persists in dynamic scenarios, and whether the preliminary model from Section 4.4.2 can adequately describe these dynamics.

Appendix A

Supplementary Calculations and Additional Theory

A.1 Sampling Random Numbers

For the proposal of new node positions, we need to sample these out of a probability distribution that is defined by an energy landscape. We will therefore go through some basic techniques on how we find these random numbers.

A.1.1 The Inversion Method

One of the two main methods we employ to sample a random number $x \in X$ out of a probability distribution $p_X(x)$ is the so called *inversion method*. It works in the following.

1. Calculate the (if not yet) normalised cumulative probability density function (CDF) as
$$\mathrm{CDF}(t) := \frac{\int_{-\infty}^{t} p(x)\mathrm{d}x}{\int_{-\infty}^{\infty} p(x)\mathrm{d}x} \in [0,1] \tag{A.1}$$
2. Sample a random number $u \sim \mathrm{Uniform}(0,1)$ uniformly between 0 and 1.
3. Solve $\mathrm{CDF}(v) = u$ for v. The so chosen v is sampled according to $p_X(v)$.

The method effectively works by looking at the area under the probability density function and choosing a point in that area at random.

A.1.2 The Rejection Sampling Method

For many CDFs, the aforementioned inversion is not easily performed. For such distributions, in 1951 John von Neumann introduced the method of *rejection sampling* in a talk, of which nowadays a copy is hard to find (von Neumann 1951). It allows to sample values from a complicated probability density function $f_X(x)$ by sampling numbers from a better known PDF $g_X(x)$ at the cost of rejecting some of the sampled numbers.

To generate a random number v according to $f_X(x)$:

1. Define an *acceptance rate* k with the property that
$$k \cdot g_X(x) \geq f_X(x) \; \forall \; x \; \in \Omega \tag{A.2}$$

2. Sample a random number $v \sim g_X(x)$ according to $g_X(x)$.

3. Accept the so chosen number with probability

$$P_{\text{accept}}(v) = \frac{f(v)}{k \cdot g(v)} \tag{A.3}$$

   If rejected, return to step 2.

The idea being again that we sample uniformly the area under the curve, but now of $g_X(x)$. If chosen a point that way, we simply ask by how much $k \cdot g_X(x)$ overestimates $f_X(x)$ and accept with the therefore given probability.

A.1.3 Sampling a Random Node Position

We are tasked with the challenge of sampling the position of a node according to the PDF

$$p(L, \vartheta) = \frac{1}{Z} \exp\left(-\beta \left[k_\vartheta(\vartheta - \vartheta_0)^2 + k_L(L - L_0)^2\right]\right) \tag{A.4}$$

where the lengths are in units of L_0. The coordinate transform from cartesian to polar coordinates introduces another factor of L, as $\mathrm{d}x\,\mathrm{d}y = L\mathrm{d}L\,\mathrm{d}\vartheta$. The two marginals we can therefore look at are defined as

$$\begin{aligned} p_\vartheta(\vartheta) =& \frac{1}{Z_\vartheta} \exp\left(-\beta k_\vartheta(\vartheta - \vartheta_0)^2\right) \\ p_L(L) =& \frac{1}{Z_L} L \exp\left(-\beta k_L(L - L_0)^2\right) \end{aligned} \tag{A.5}$$

Sampling the Angle

We want to sample an angle at which a new node is positioned. For this, it is easiest to sample the angle deviation $\Delta\vartheta$ from the ideal angle, which is 120° or $\frac{2}{3}\pi$ radians. The probability is distributed according to

$$p(\Delta\vartheta) = \frac{1}{Z_\vartheta} e^{-\beta k_\vartheta (\Delta\vartheta)^2} \quad \Delta\vartheta \in [-\pi, \pi] \tag{A.6}$$

The normalisation Z_ϑ is defined to be

$$Z_\vartheta := \int_{-\pi}^{\pi} \mathrm{d}(\Delta\vartheta)\, e^{-\beta k_\vartheta (\Delta\vartheta)^2} = \sqrt{\frac{\pi}{\beta k_\vartheta}}\, \mathrm{erf}(\sqrt{\beta k_\vartheta}\pi) \tag{A.7}$$

where we employed the definition of the error function

$$\mathrm{erf}(z) := \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} \mathrm{d}t \tag{A.8}$$

We sample $\Delta\vartheta$ by first uniformly sampling $\rho \in [0, 1]$ and then equating this with the CDF of $\Delta\vartheta$, which is defined to be

$$\mathrm{CDF}(\Delta\vartheta) := \frac{1}{Z_\vartheta} \int_{-\pi}^{\Delta\vartheta} e^{-\beta k_\vartheta u^2} \mathrm{d}u = \frac{1}{2Z_\vartheta} \sqrt{\frac{\pi}{\beta k_\vartheta}} \left\{\mathrm{erf}(\pi\sqrt{\beta k_\vartheta}) + \mathrm{erf}(\sqrt{\beta k_\vartheta}\Delta\vartheta)\right\} \tag{A.9}$$

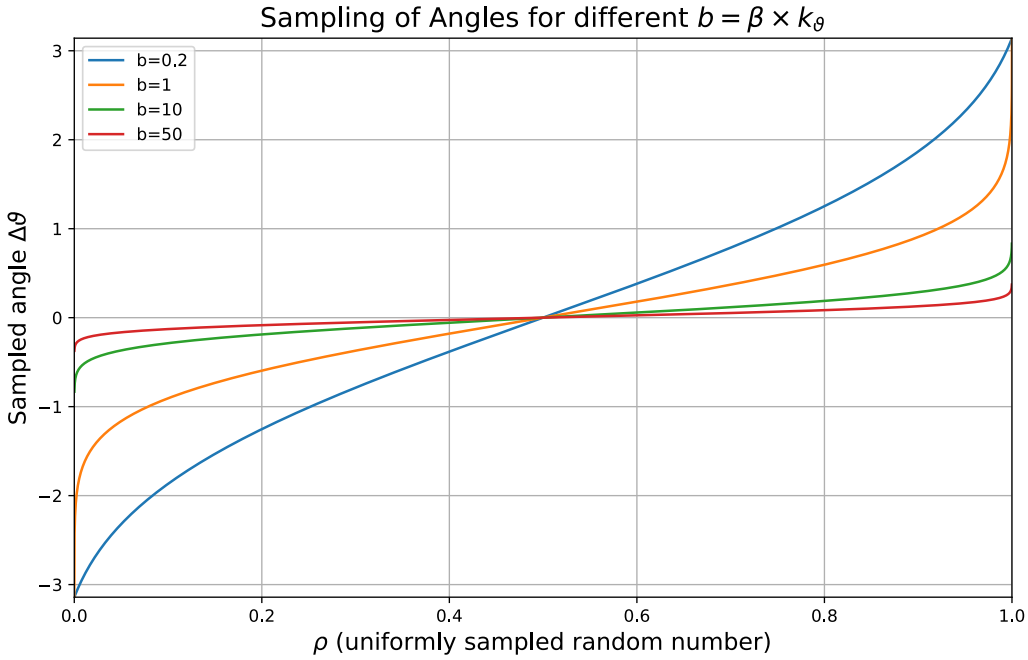

Figure A.1: Angular Sampling. The sampled angle $\Delta\vartheta$ as a function of ρ (the uniformly sampled random variable) and the parameter $b = \beta \times k_\vartheta$.

Therefore, the sampled $\Delta\vartheta$ is set to be

$$\begin{aligned}\Delta\vartheta =& \frac{1}{\sqrt{\beta k_\vartheta}} \operatorname{erf}^{-1}\left\{2Z_\vartheta \times \rho \times \sqrt{\frac{\beta k_\vartheta}{\pi}} - \operatorname{erf}(\pi\sqrt{\beta k_\vartheta})\right\} \\ =& \frac{1}{\sqrt{\beta k_\vartheta}} \operatorname{erf}^{-1}\left\{(2\rho - 1)\operatorname{erf}(\pi\sqrt{\beta k_\vartheta})\right\}\end{aligned} \tag{A.10}$$

which seems plausible, as we symmetrically sample a number above or below π, with π being the maximum. The dependency of $\Delta\vartheta$ on ρ and $\beta \times k_\vartheta$ can be seen in Figure A.1.

Sampling the Leg Length

Because of the linear factor of L, the inversion method does not work well with $p_L(L)$, which is why we will employ the rejection sampling method. We will however make one simplification. The system we are looking at is finite, and therefore the sampled length will effectively be finite. We will therefore restrict L to be in $L \in (0, L_{\max}]$. It will turn out that the choice $L_{\max} = 2L_0$ will simplify many expressions in the following. It seems plausible that a leg will never stretch to double its length if we stick to the biological regime.

The normalisation constant is then defined to be (with $b := \beta k_L$)

$$\begin{aligned}Z_L :=& \int_0^2 \mathrm{d}x\, x e^{-b(x-1)^2} = \int_{-1}^1 e^{-bu^2}\mathrm{d}u + \int_{-1}^1 u e^{-bu^2}\mathrm{d}u \\ =& 2\int_0^{\sqrt{b}} e^{-v^2}\frac{\mathrm{d}v}{\sqrt{b}} + \left[\frac{-1}{2b}e^{-bv^2}\right]_{-1}^1 = \sqrt{\frac{\pi}{b}}\operatorname{erf}(\sqrt{b})\end{aligned}$$

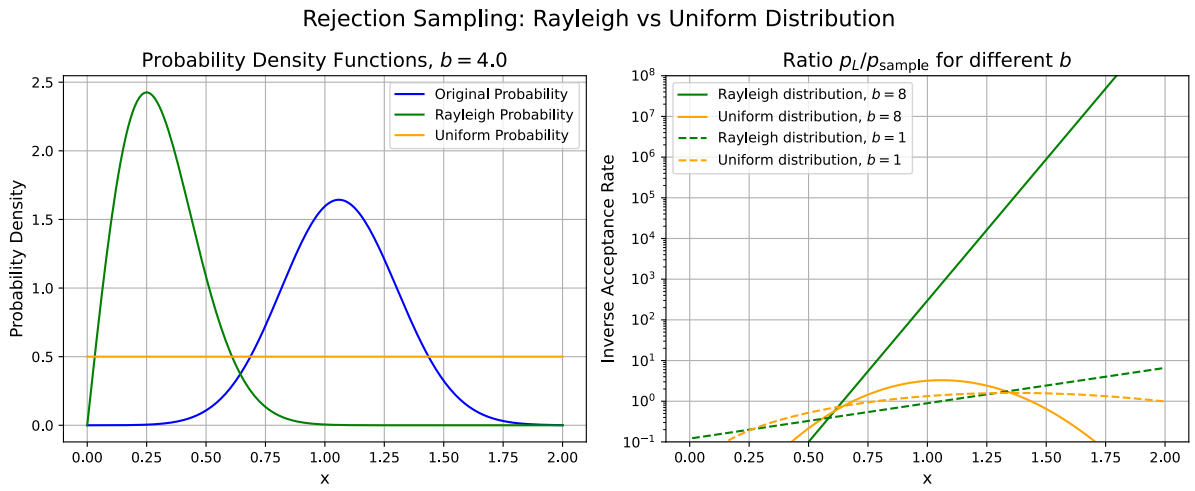

Figure A.2: Uniform rejection sampling distribution. A uniform sampling probability distribution seems to result in a much higher acceptance probability.

It seems that a uniform sampling probability distribution yields much higher acceptance rates than a Rayleigh distribution, as can be seen in Figure A.2. Around where we expect most lengths to lie within, the acceptance rate is around 1/3.

On $x \in [0, 2]$, the uniform probability distribution is defined to be

$$p_U(x) = \frac{1}{2} \tag{A.11}$$

To find the acceptance rate, we have to find the maximum of $p_L(x)$. This is given by the root of the derivative, it it lies within the definition interval, or by the border value of the definition interval:

$$\frac{\mathrm{d}p_L(x)}{\mathrm{d}x} = \frac{e^{-b(x-x_0)^2}}{Z_L}\left[1 - 2b(x - x_0)\right] \stackrel{!}{=} 0 \Rightarrow x_{\max} = x_0 + \frac{1}{2b}$$

Using $x_0 = 1$, we find that

$$x_{\max} = \begin{cases} 1 + \frac{1}{2b} & | \; b \geq \frac{1}{2} \\ 2 & | \; b < \frac{1}{2} \end{cases} \tag{A.12}$$

and therefore

$$p_L(x_{\max}; b) = \begin{cases} \frac{1+\frac{1}{2b}}{Z_L} e^{-1/4b} & | \; b \geq \frac{1}{2} \\ \frac{2}{Z_L} e^{-b} & | \; b < \frac{1}{2} \end{cases} \tag{A.13}$$

We therefore find the acceptance rate for a uniform sampling distribution to be

$$\boxed{k_{\text{uniform}}(b) = \begin{cases} \frac{2+1/b}{\sqrt{\frac{\pi}{b}}\,\mathrm{erf}(\sqrt{b})} e^{-1/4b} & | \; b \geq \frac{1}{2} \\ \frac{4}{\sqrt{\frac{\pi}{b}}\,\mathrm{erf}(\sqrt{b})} e^{-b} & | \; b < \frac{1}{2} \end{cases}} \tag{A.14}$$

The acceptance rate $k(b)$ as a function of b for a uniform sample distribution can be seen in Figure A.3. We see that even over a large range of b-values, the acceptance rate stays within a limited range and is of rather small magnitude.

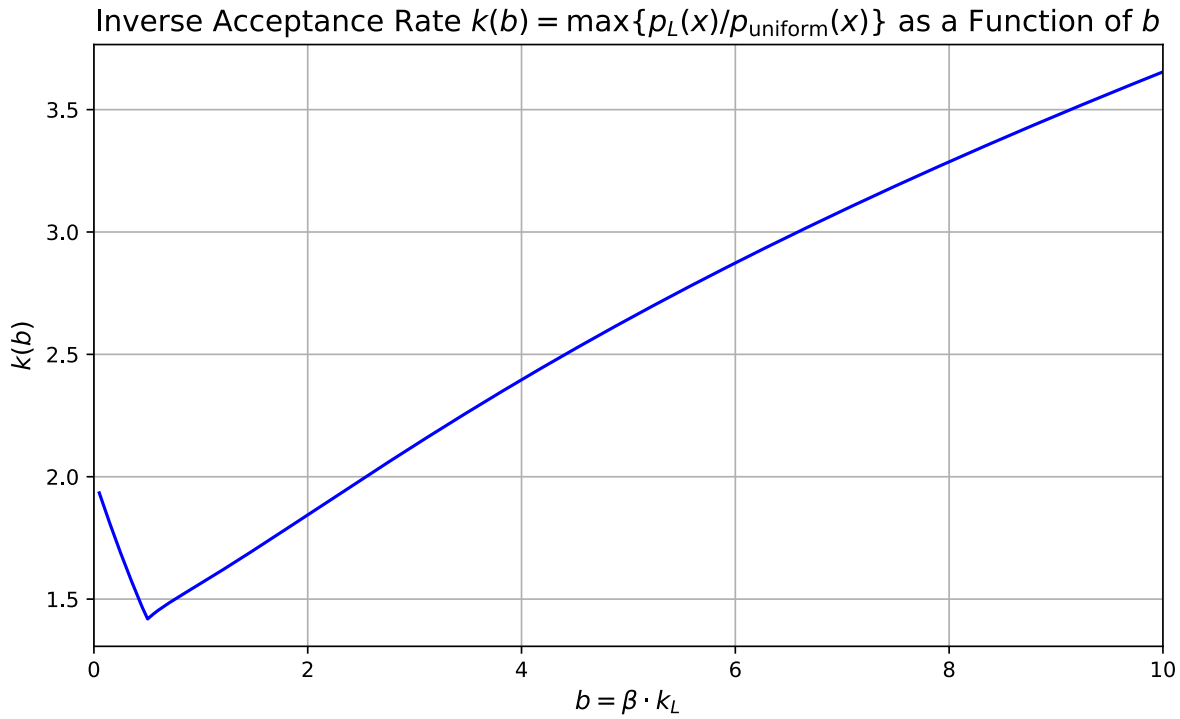

Figure A.3: Inverse acceptance rate for rejection sampling. The acceptance rate for a uniform sampling probability distribution as a function of b.

A.2 Stereographic Projection

The stereographic projection is a standard way to project a sphere onto a plane. The simplest way to think about is is to position the projection plane at the equator of the sphere. One then chooses one of the poles, called the *projection point* P. For any other point on the sphere K, the line drawn between P and K intersects the plane at exactly one point P', which is the projection of P onto the plane. This projection has only one singular point, which is the projection point itself, which gets mapped to infinity.

In the following, we denote the 2d projected coordinates by x, y and the 3d spherical coordinates by X, Y, Z. For a sphere with radius R, we have $R^2 = X^2 + Y^2 + Z^2$. Since we have two poles, we have two sets of formulae: If we choose the north pole as the projection point, we get the so called *southern projection* and vice versa. The naming comes from the fact that e.g. in the southern projection, the southern hemisphere is singularity free and the distortion is limited.

The relations between the two sets of coordinates can be found in Table A.1. For some visualisation methods it is important to know that the infinitessimal area piece $\mathrm{d}A_{\text{Sphere}}$ of the sphere relates to the infinitessimal area piece on the plane via

$$\mathrm{d}A_{\text{sphere}} = \frac{4}{(1 + (x/R)^2 + (y/R)^2)^2} \mathrm{d}x\mathrm{d}y \tag{A.15}$$

A.3 Ideal Node Number

Every sphere of radius R has a number of clathrins it can fit if every clathrin occupies space according to itx relaxed leg length. We call this number the *ideal node number*.

	Northern Projection	Southern Projection
$x =$	$RX/(R+Z)$	$RX/(R-Z)$
$y =$	$RY/(R+Z)$	$RY/(R-Z)$
$X =$	$2R^2x/(R^2+x^2+y^2)$	$2R^2x/(R^2+x^2+y^2)$
$Y =$	$2R^2/(R^2+x^2+y^2)$	$2R^2/(R^2+x^2+y^2)$
$Z =$	$R(R^2-x^2-y^2)/(R^2+x^2+y^2)$	$R(-R^2+x^2+y^2)/(R^2+x^2+y^2)$

Table A.1: Relations in different stereographic projections. The relations for the 3d coordinates X, Y, Z and the projected coordinates x, y change slightly between the two different projections.

Imagine a sphere of curvature H, i.e. with radius $R = 1/H$. This sphere has a surface of $A = 4\pi R^2$. We can approximate the number of clathrin hubs able to fit on the sphere in a hexagonal grid by assuming the sphere is large enough as to us being able to tile the sphere-surface with the flat hexagons. The area of a hexagon with single-side-length a is $A_{\text{hex}} \approx 2.598 \times a^2$, and every clathrin touches three hexagons, of which it can "claim" a sixth of its area. Therefore, the single clathrin area is

$$A_{\text{cl}} = \frac{3}{6} A_{\text{hex}} = \frac{2.598}{2} \times L_0^2 \tag{A.16}$$

if L_0 is the equilibrium leg-length of a single clathrin leg. Therefore, the *ideal node number* for a coat of curvature $H = 1/R$ is

$$N_{\text{Cl,ideal}} \approx \frac{4\pi R^2}{L_0^2 \times 1.3} = \frac{4\pi}{H^2 L_0^2 \times 1.3} \tag{A.17}$$

which can be seen in Figure A.4.

A.4 Incorporating a Middle Layer

Clathrin and the underlying membrane are not directly coupled but are connected through an approximately 20 nm thick layer. If we say the layer has a thickness d, then the bending radius of the underlying membrane is given by

$$R_M = R_C - d \tag{A.18}$$

and therefore the curvature of the underlying membrane is given by

$$H_M = \frac{1}{R_M} = \frac{1}{R_C - d} = \frac{1}{\frac{1}{H_C} - d} = \frac{H_C}{1 - dH_C} \tag{A.19}$$

Only for smaller curvatures of the clathrin coat, a modification of the membrane's curvature is to be expected.

A.5 Relaxation Times

For our Metropolis-Hastings Monte Carlo algorithm, we have to ensure that our final drawn distribution is indeed from an equilibrium distribution, i.e. that all correlations from the initial state have more or less died out. To gauge the order of magnitude for

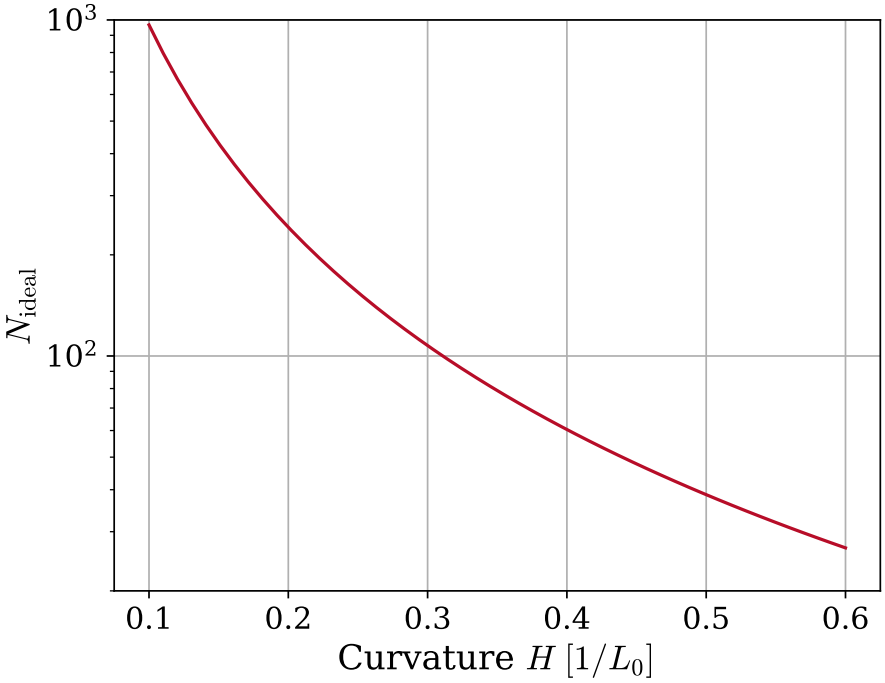

Figure A.4: The ideal node number vs. curvature. The ideal number of clathrin nodes against curvature for a full spherical coverage. Small curvature changes can already lead to large changes in N_{ideal}.

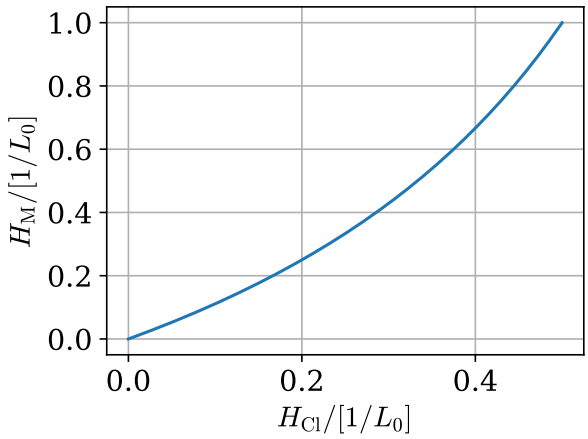

Figure A.5: Membrane curvature against clathrin coat curvature. For a connecting layer of thickness $d = 1L_0$, shown is the corresponding curvature of the underlying membrane versus the curvature of the clathrin coat.

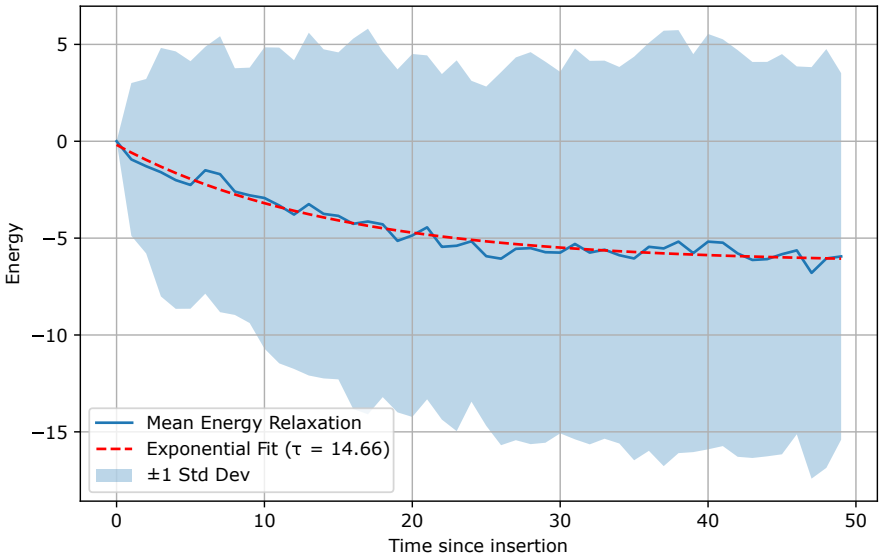

Figure A.6: Decay of energy after node addition. To gauge the number of relaxation steps t necessary to equilibrate the system, one can look at the energy after a node addition and its evolution during a MH-MC equilibration. While the fluctuations are large, we see that on average the energy decays with a decay time of $\tau \approx 15$ steps.

such a process, we can look at a small system of nodes and track the energy $E(t)$ of the system after a node addition even as a function of number of Monte Carlo steps t.

This is shown in Figure A.6. There, a decay-time of $\tau = 15$ steps is recorded on average. This number of steps will in general depend on the system. To ensure we will usually sample from a distribution unbiased by initial conditions, we usually choose $\tau = 50$. This number can of course be increased at the cost of runtime to make sure all correlations have died out.

A.6 Preferred Hub-Angle

The alternative to the pucker angle χ (which uses the projection-plane tangential to the horizontal one) is to calculate the angle θ_{ij} between the vectors representing the physical bonds. For references to the geometry, see section 3.2.

For this, we again want to work in the approximation that the area covered by clathrin is much smaller than the area of the would-be sphere on which the clathrin-coat grows. Assume that we have node on top of the sphere at $(0, 0, R)$ and assume that the next "ring" of nodes is at a lower altitudal angle $\theta = \pi - \Delta\theta$. With this, the area covered by clathrin is

$$A_{Cl} = \pi(R \times \Delta\theta)^2 \tag{A.20}$$

and two of the neighboring points P_1, P_2 are

$$P_1 = (R\Delta\theta, 0, R(1 - \Delta\theta/2)^2) \ , \quad P_2 = (R\Delta\theta\cos(2\pi/3), R\Delta\theta\sin(2\pi/3), R(1 - \Delta\theta/2)^2) \tag{A.21}$$

if we assume that the legs point out at a projected angle of exactly $2\pi/3$, up to order

$\mathcal{O}(\Delta\theta^3)$. The vectors from our central node to the neighboring points P_1, P_2 are then

$$V_1 = (R\Delta\theta, 0, R(\Delta\theta/2)^2)\ , \quad V_2 = (R\Delta\theta\cos(2\pi/3), R\Delta\theta\sin(2\pi/3), R(\Delta\theta/2)^2) \tag{A.22}$$

both with norm

$$\|V_{1,2}\| = R\Delta\theta \times \sqrt{1+\Delta\theta^2/4} \tag{A.23}$$

The ideal angle between these two legs, θ_{id}, is then calculated by

$$\begin{aligned}\frac{V_1 \cdot V_2}{\|V_1\| \times \|V_2\|} =& \frac{(R\Delta\theta)^2\cos(2\pi/3) + \frac{R^2\Delta\theta^4}{4}}{R^2\Delta\theta^2\,(1+\Delta\theta^2/4)} = \frac{\cos(2\pi/3)+\Delta\theta^2/4}{1+\Delta\theta^2/4} \\ &\approx \left(\cos(2\pi/3)+\Delta\theta^2/4\right) \times \left(1-\Delta\theta^2/4\right) \\ &= \cos(2\pi/3) + (1-\cos(2\pi/3))\,\Delta\theta^2/4\end{aligned}$$

We can now use eq. A.20 to replace $\Delta\theta$ to finally find

$$\boxed{\cos\theta_{id} = \cos(2\pi/3) + \frac{A_{Cl}}{4\pi R^2}\,(1-\cos(2\pi/3))} \tag{A.24}$$

where we could replace $4\pi R^2 = A_{Sp}$ again.

Is the approximation sensible? The area covered by a single clathrin in a clathrin coat is between $A_{Cl} \approx (105-218)\text{nm}^2$ (Zeno et al. 2021; Tagiltsev et al. 2021). A typical clathrin cage has a radius of $R_{cage} \approx (32-50)\text{nm}^2$ (Frey and Schwarz 2024), CCVs are usually a bit larger. This leads to preferred angles of

$$\theta_{id} = 118.4^\circ - 119.8^\circ$$

which are in agreement to the angles presented in Table 1.1.

A.7 Radius sampling

We want to sample curvature changes, since this is the quantity appearing in the energy. But for our code, we need to convert a curvature change ΔH into a radius change ΔR. For this we notice that

$$\Delta H = H' - H = \frac{1}{R'} - \frac{1}{R} \tag{A.25}$$

and therefore

$$R' = \frac{1}{\Delta H + \frac{1}{R}} \Rightarrow \Delta R = R' - R = \frac{R}{1+R\Delta H} - R \tag{A.26}$$

A.8 Simpler Form of the Helfrich Hamiltonian

We want to show that in the special case of a spherical-like membrane, i.e. with equal principal curvatures $\kappa_1 = \kappa_2$ and therefore $K = H^2$, the Helfrich-Hamiltonian of eq. 2.1 assumes a simpler form. Let $\kappa' := \kappa + \kappa_G$ and $\overline{H}' := \frac{\kappa}{\kappa'}\overline{H}$. Then we see

$$\begin{aligned}E &= \int \mathrm{d}A\,\kappa\left(H-\overline{H}\right)^2 + \kappa_G K = \int \mathrm{d}A\,(\kappa+\kappa_G)\,H^2 - 2H\kappa\overline{H} + \kappa\overline{H}^2 \\ &= \int \mathrm{d}A\,\kappa'\left\{H^2 - 2H\frac{\kappa}{\kappa'}\overline{H} + \frac{\kappa}{\kappa'}\overline{H^2}\right\} = \int \mathrm{d}A\,\kappa'\left\{\left(H-\overline{H}'\right)^2 + \overline{H}'^2 + \frac{\kappa'}{\kappa}\overline{H}'^2\right\} \\ &= \int \mathrm{d}A\,\kappa'\left(H-\overline{H}'\right)^2 + \int \mathrm{d}A\,\kappa'\left(1+\frac{\kappa'}{\kappa}\right)\overline{H}'^2\end{aligned}$$

With our new spontaneous curvature $\overline{H}'$ and the new bending modulus κ', the curvature-dependant part of the Helfrich-Hamiltonian has the of one with $\kappa_G \equiv 0$, but now with a surface-area dependant part (in addition to any other surface tension). Since on the sphere the curvature does not depend on the position, we get for a spherical-like membrane with surface area A and radius R that the bending energy has the form

$$E_{\mathcal{M}}(R, A) = \kappa' \left(H - \overline{H}'\right)^2 \times A + \kappa' \left(1 + \frac{\kappa'}{\kappa}\right) \overline{H}'^2 \times A \tag{A.27}$$

We can further define the pseudo surface tension σ_P and finally get

$$\boxed{E(R, A) = \kappa' \left(H - \overline{H}'\right)^2 \times A + \sigma_P \times A} \tag{A.28}$$

where the new constants relate to the old ones like

$$\kappa' := \kappa + \kappa_G \quad , \quad \overline{H}' := \frac{\kappa}{\kappa'} \overline{H} \quad , \quad \sigma_P := \kappa' \left(1 + \frac{\kappa'}{\kappa}\right) \overline{H}'^2$$

For vanishing spontaneous curvature, $\overline{H} = 0$, σ_P vanishes and $\overline{H}' = 0$, leading to only a modification of the bending rigidity κ'.

A.9 Calculating the Base Rate

For the calculation of the base rate we employ the definition of the **error function**

$$\operatorname{erf}(z) := \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} \mathrm{d}t \tag{A.29}$$

Then we can write for the angular integral

$$\begin{aligned}
\int_0^{2\pi} \mathrm{d}\vartheta \, \exp\left(-\beta k_\vartheta (\vartheta - \vartheta_0)^2\right) &= \int_{-\vartheta_0}^{2\pi - \vartheta_0} \mathrm{d}u \, \exp\left(-\beta k_\vartheta u^2\right) \\
&= \frac{1}{\sqrt{\beta k_\vartheta}} \int_0^{\sqrt{\beta k_\vartheta} 2\pi - \vartheta_0} e^{-v^2} \mathrm{d}v + \frac{1}{\sqrt{\beta k_\vartheta}} \int_{-\sqrt{\beta k_\vartheta}\vartheta_0}^0 e^{-v^2} \mathrm{d}v \\
&= \frac{\sqrt{\pi / \beta k_\vartheta}}{2} \left[\operatorname{erf}\left(\sqrt{\beta k_\vartheta}(2\pi - \vartheta_0\right) + \operatorname{erf}\left(\sqrt{\beta k_\vartheta}\vartheta_0\right)\right]
\end{aligned}$$

Separated from this, we can evaluate the radial integral:

$$\begin{aligned}
\int_0^\infty \mathrm{d}R \, R \exp\left(-\beta k_R (R - R_0)^2\right) &= \int_{-R_0}^\infty \mathrm{d}u (u + R_0) e^{-\beta k_R u^2} \\
&= \int_{-R_0}^\infty \mathrm{d}u \, u e^{-\beta k_R u^2} + \int_{-R_0}^\infty \mathrm{d}u \, R_0 e^{-\beta k_R u^2} \\
&= \left[\frac{-1}{2\beta k_R} e^{-\beta k_R u^2}\right]_{-R_0}^\infty + \frac{R_0}{\sqrt{\beta k_R}} \int_{-\sqrt{\beta k_R} R_0}^\infty \mathrm{d}v \, e^{-v^2} \\
&= \frac{\exp\left(-\beta k_R R_0^2\right)}{2\beta k_R} + \frac{R_0}{\sqrt{\beta k_R}} \frac{\sqrt{\pi}}{2} \left[\operatorname{erf}(\infty) + \operatorname{erf}\left(\sqrt{\beta k_R} R_0\right)\right] \\
&= \frac{\exp\left(-\beta k_R R_0^2\right)}{2\beta k_R} + \frac{\sqrt{R_0^2 \pi / \beta k_R}}{2} \left[1 + \operatorname{erf}\left(\sqrt{\beta k_R} R_0\right)\right]
\end{aligned}$$

We can then combine everything together to find the total base rate to be

$$\begin{aligned}k^B(L_0,\vartheta_0;k_\vartheta,k_L) :=& \gamma\int \mathrm{d}A\,\exp\left(-\beta\left[-\Delta E_{\mathrm{pol,hub}}-\Delta E_{\mathrm{pol,leg}}+k_\vartheta(\vartheta-\vartheta_0)^2+k_L(L-L_0)^2\right]\right)\\ =&\gamma e^{\beta(\Delta E_{\mathrm{pol,hub}}+\Delta E_{\mathrm{pol,leg}})}\left\{\frac{\sqrt{\pi/\beta k_\vartheta}}{2}\left[\mathrm{erf}\left(\sqrt{\beta k_\vartheta}(2\pi-\vartheta_0)\right)+\mathrm{erf}\left(\sqrt{\beta k_\vartheta}\vartheta_0\right)\right]\right\}\\ &\times\left\{\frac{\exp\left(-\beta k_R R_0^2\right)}{2\beta k_R}+\frac{\sqrt{R_0^2\pi/\beta k_R}}{2}\left[1+\mathrm{erf}\left(\sqrt{\beta k_R}R_0\right)\right]\right\}\end{aligned}$$

A.10 Equivalence of BKL-KMC and rKMC

It is quit insightful and gives some intuition to follow the proof of Serebrinsky (Serebrinsky 2011) and understand why the rejection-KMC method works as it does.

Lets set the groundwork: We look at a physical process that describes the time-evolution of a probability distribution $P_{ij}(t)$, that is the probability that the state X of our system at time t is found in state j if it started at time $t=0$ at state i: $P_{ij}(t) = \Pr\{X(t)=j|X(0)=i\}$. This is connected to the master equation, which simply describes the time-evolution of said probability distribution:

$$\frac{\mathrm{d}P_{ij}(t)}{\mathrm{d}t}=\sum_k\left[P_{ik}(t)q_{kj}-P_{ij}(t)q_{jk}\right]=\sum_{k\neq j}\left[P_{ik}(t)q_{kj}-P_{ij}(t)q_{jk}\right]$$

where q_{ij} is the rate with which the state i transitions into state j. This distribution $P(t)$ defines the Markow process, since the probabilites at every time-step only depend on the probabilites at the previous time-step. Equivalent to P, one can construct the s.c. Q-matrix with $Q=(q_{ij})=P'(0):=\frac{\mathrm{d}P_{ij}(0)}{\mathrm{d}t}$, since $P_{ij}(t=0)=\delta_{ij}$. Per conservation of probability, this matrix satisfies

$$\begin{aligned}&q_{ij}\geq 0 \quad \text{for } i\neq j && (1)\\ &q_{ii}\leq 0 && (2)\end{aligned}$$

with

$$-q_{ii}:=r_i=\sum_{j\neq i}q_{ij} \tag{A.30}$$

For every so defined Q-matrix, there exist a unique probability transition function $P(t)$, and therefore the specification of $P(t)$ is equivalent to the specification of the Q-matrix.

A third definition of the same Markow-process, and for our purposes very useful, are the following two properties (Sadiq 1984):

1. The time a state i persists, called the *holding time* T_i, is a random variable. It is exponentially distributed[1] with

$$P_{T_i}(t)=\Pr\{T_i>t|X(0)=i\}=\exp\left(-r_i t\right) \tag{A.31}$$

where the exit rate r_i controlls the distribution. The *mean residence time* is $\tau_i = 1/r_i$.

[1]The exponential distribution is simply so because at every timestep Δt, the probability the state remains is $1-\Delta t\times r_i$. Therefore, for finite time $t=n\times\Delta t$, we have $P=(1-r_i t/n)^n=\exp(-r_i t)$.

2. If the process exits state i, it enters state j with probability $q_{ij}/\sum_{j\neq i} q_{ij} = q_{ij}/r_i$:

$$\Pr\{X(T_i) = j | X(0) = i\} = \frac{q_{ij}}{r_i} \quad (j \neq i) \tag{A.32}$$

A Markow-chain S with rate-vector $\boldsymbol{r}$ that satisfies these conditions is[2]

$$S_{ij} = \begin{cases} 0 & \text{if} \quad j = i \\ q_{ij}/r_i & \text{if} \quad j \neq i \end{cases} \tag{A.33}$$

which is a direct implementation of rejection-free KMC:

1. Initially, the system is in state i at time t.
2. A new state $j \neq i$ is drawn with probability S_{ij}.
3. The time is advanced by Δt drawn from an exponential distribution with parameter r_i.

We can now show that there exist other pairs $(V, \boldsymbol{w})$ of Markow-chains V and rates $\boldsymbol{w}$ that describe the same process. They form a continuous set, in which a limiting case is the rejection free Markow-chain S.

Let us therefore define a new process with **fictitious exit rates** $w_i \geq r_i$. The name *exit*-rate might be misleading: It is the rate at which a state-change is *attempted*. The exit from said state is however not guaranteed. Intuition for this comes from the idea of *oversampling*: The rejection-free process looks at the system at exactly the minimum number of times necessary to see every change. By using higher fictitious exit rates w_i, we will essentially look more often at the system, but accept transition with a lower probability, namely r_i/w_i, and therefore reject state-changes with **rejection probability** $\gamma_i := 1 - r_i/w_i$. Therefore lets define the alternative Markow-chain V as

$$V_{ij}[\boldsymbol{\gamma}(\boldsymbol{w})] = \begin{cases} \gamma_i & \text{if} \quad j = i \\ (1-\gamma_i)S_{ij} = \frac{r_i}{w_i} S_{ij} & \text{if} \quad j \neq i \end{cases} \tag{A.34}$$

Now the system is allow to remain in state i with probability γ_i. It leaves state i with probability $1-\gamma_i = \omega_i/r_i$ at every step[3], and therefore remains at said state with parameter $r_i \times w_i/r_i = w_i$ for the exponential distribution of the holding time. If it leaves, it chooses then state j with the combined probability of leaving and choosing this exact j, $(1-\gamma_i)S_{ij}$.

This is the situation a rejection-method describes:

1. Start at time t in state i
2. Choose a state $j \neq i$ and accept it by probability $V_{ij} = \frac{r_i}{w_i} S_{ij} \leq S_{ij}$.
3. Increase the time, whether or not the state change was accepted, by Δt drawn from an exponential distribution with parameter $N_s w_i$, the rate at which state-changes are attempted times the number of states out of which j is chosen.

[2]Where we assume no *absorbing states* with exit rate $r_i = 0$ exist. For such states, a rejection-free KMC is not feasible.

[3]Again here the idea is that if $w_i = r_i$, i.e. we sample at the minimu possible rate, then at every step the system leaves its state.

The important distinction distinction lies in the second step: In rfKMC, the state j is drawn using a probability distribution that requires knowledge of **all** S_{ij}, since each state is chosen with probability q_{ij}/r_i, and the computation of r_i requires knowledge of all q_{ij}, whereas in rKMC, a j is chosen uniformly between all possible js and is accepted using only the knowledge of said transition $V_{ij} = q_{ij}/w_i$!

It now remains to show that the pair $(V, \boldsymbol{w})$ describes the same process as $(S, \boldsymbol{r})$, and for this we need to prove that the two properties (A.31) and (A.32) are equal for both processes.

Lets start with the second property: For $(S, \boldsymbol{r})$, it is obvious that if the process exits state i, it enters state j by probability q_{ij}/r_i.

For $(V, \boldsymbol{w})$, this also follows almost immediately. The probability of choosing state j,if the system exits state i, is given by

$$\frac{V_{ij}}{\sum_{j\neq i} V_{ij}} = \frac{(1-\gamma_i)S_{ij}}{(1-\gamma_i)\sum_{j\neq i} S_{ij}} = S_{ij} = \frac{q_{ij}}{r_i} \checkmark$$

where we have to divide by the sum of the V_{ij} since they are not yet normalized to 1. Then we use the property that $\sum_{j\neq i} S_{ij} = 1$.

For the first property, we have to prove that the holding time T_i, i.e. the time that the process remains in state i, is distributed by an exponential distribution with parameter r_i. For $(S, \boldsymbol{r})$, this follows by definition. For $(V, \boldsymbol{w})$, we have to work a tiny bit. For notation, one denotes with (i, j, m) the event that the m-th jump is from state i to state j, and with $N(t)$ that at time t, N state changes have been attempted. Then the probability-distribution for the holding time can be written as

$$\begin{aligned} P_{T_i}(t) :=& \Pr\{T_i > t | X(0) = i\} \\ =& \Pr\{N(t) = 0\} + \Pr\{N(t) = 1, (i,i,1)\} + \Pr\{N(t) = 2, (i,i,1), (i,i,2)\} + \dots \end{aligned}$$

The property that our process is a Markow-process means that successive jumps are *independant*, i.e. one may write

$$\Pr\{N(t) = n, (i,i,1), \dots\} = \text{Poiss}\{n, \lambda = t \cdot \omega_i\} \times V_{ii}^n$$

where $\text{Poiss}\{n, \lambda = t \cdot \omega_i\}$ is the probability distribution of a poisson-process, i.e. the probability that n jumps happen, if the expected number of jumps is $t \cdot w_i$, multiplied with the probability that every jump ends up in state i again. Then we can write

$$\begin{aligned} P_{T_i}(t) &= \sum_{n=0}^{\infty} \text{Poiss}\{n, \lambda = t \cdot \omega_i\} \times V_{ii}^n \\ &= \sum_{n=0}^{\infty} \frac{(w_i \cdot t)^n}{n!} e^{-w_i \cdot t} \times \gamma_i^n \\ &= \exp(-w_i \cdot t) \exp(\gamma_i \cdot w_i \cdot t) \\ &= \exp((1-\gamma_i) w_i \cdot t) = \exp(-r_i \cdot t) \checkmark \end{aligned}$$

which proves the second property.

A.11 Rescaling the Radius is Equivalent to Rescaling the Equilibrium Leg-Length

A mapping as shown in section 3.4 does not change any bonds of the grid. It also does not change the **angles** between bonds of nodes, since any such relation has to be dimensionless,

and therefore the radius cannot appear. The only thing that can change is the **distance** between nodes. In the limit that the coat is adiabatically relaxing to some state, i.e. we do not need to be concerned with "traveling" nodes or node-pairs, the only effect of the distance between nodes is in the node energy $E(L)$. For dimensional reasons, the energy needs to be a function of the ratio L/L_0. For any two points on the sphere, their distance d has to depend linearly on the spherical radius R, since this is the only dimensionful unit in spherical coordinates. Since $L \sim R$, we know that a rescaling of $R \to R'$ leads to a rescaling of distances, $d \to \frac{R'}{R}d$ and therefore to a rescaling of leg-lengths $L \to L' = \frac{R'}{R}L$. Since the energy depends on the ratio L/L_0, a rescaling of $R \to R'$ is therefore equivalent to a rescaling of $L_0 \to L_0' = \frac{R}{R'}L_0$.

Appendix B

Supplementary Figures

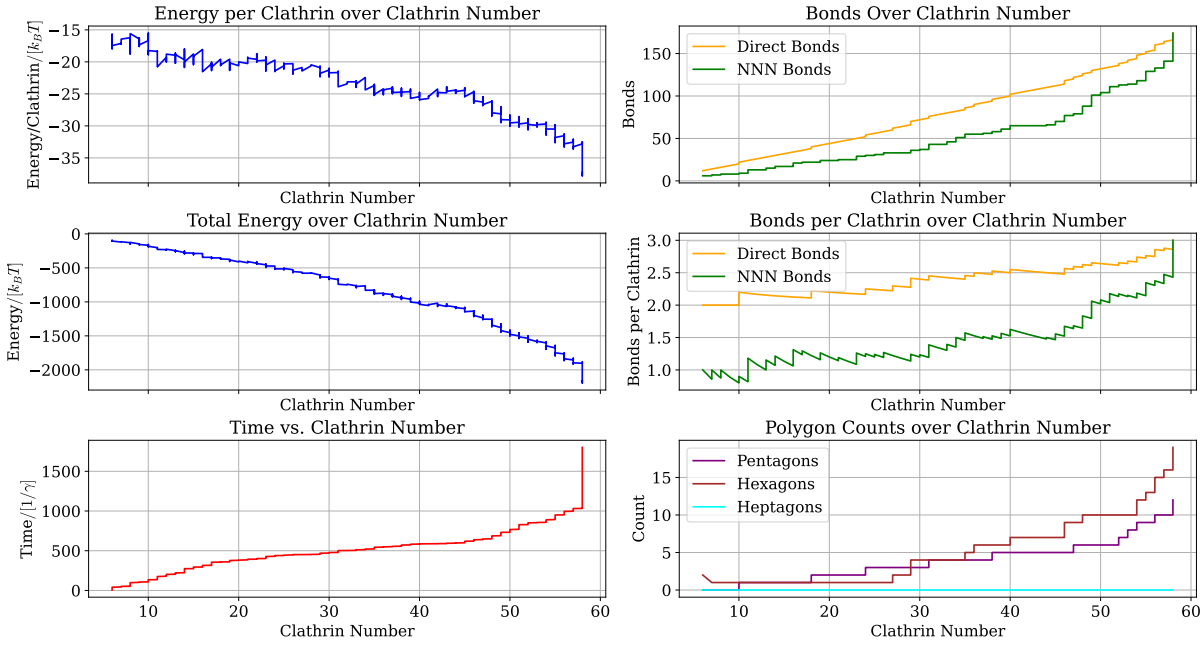

Figure B.1: Data vs node number for grid approaching a TI. The evolutionary data for the exemplary grid which approaches a truncated icosahedron. We see the curvature being slightly too large for a perfect TI. The node number does not fully reach 60, but stalls at 58. Therefore, we reach the necessary 12 pentagons, but stall at 19 hexagons.

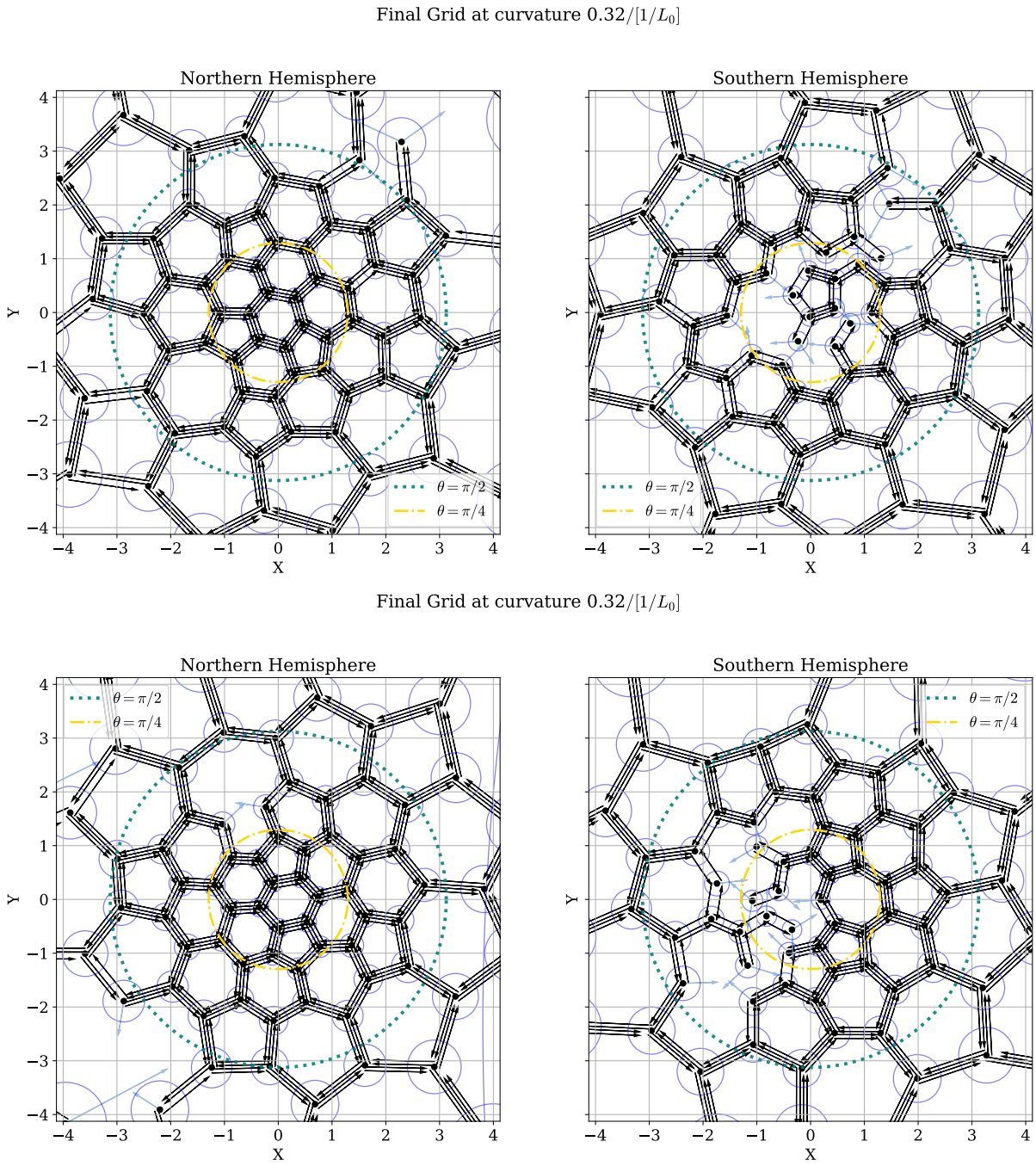

Figure B.2: Exemplary grids grown away from preferred curvature. These two grids were grown at $H = 0.32/L_0$ with a preferred curvature of $H = 0.42/L_0$, with the parameters specified in C.1. We see that while the initial coat growth is regular, the closing will likely not be happen in a regular fashion.

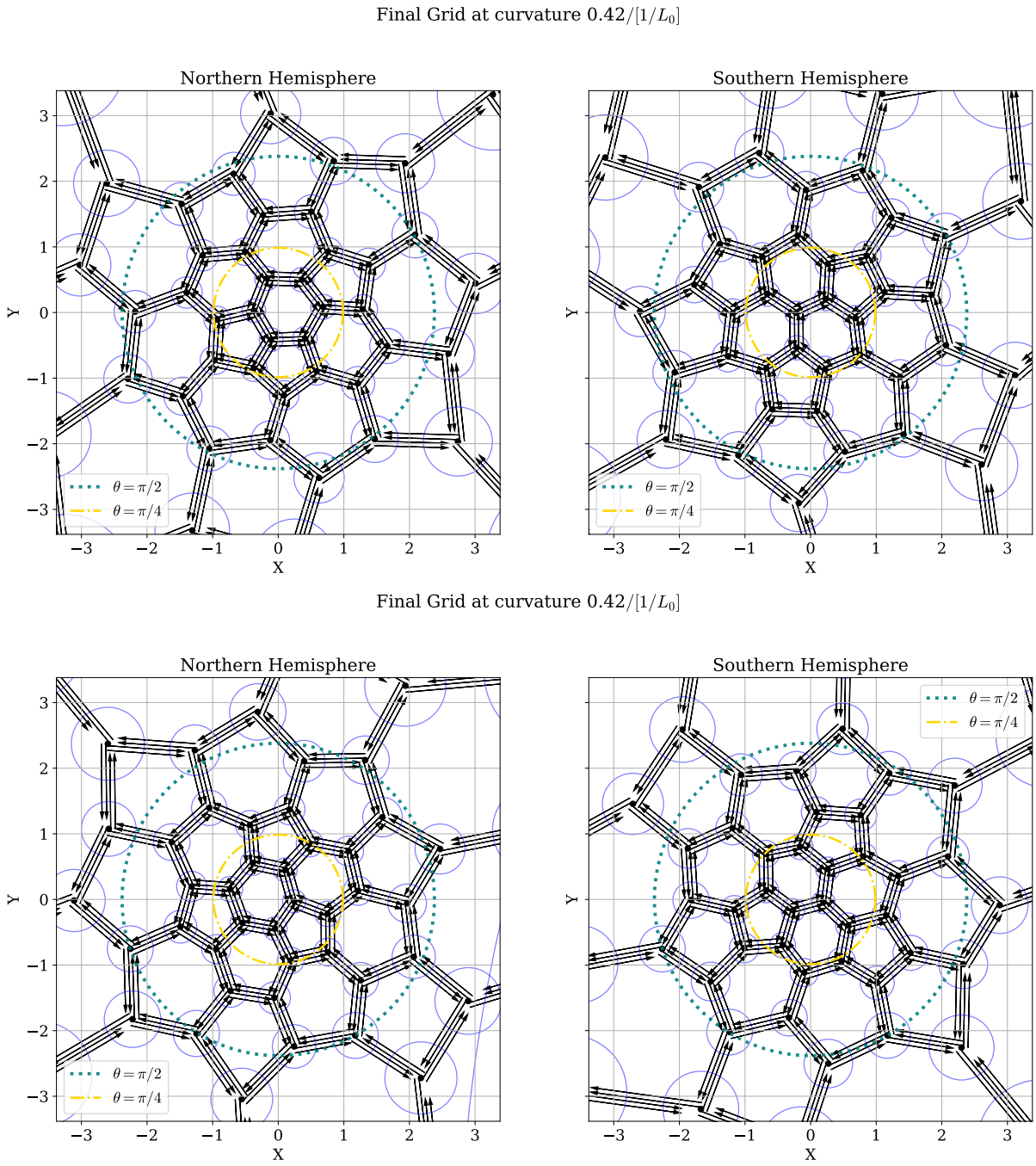

Figure B.3: Exemplary grids grown at preferred curvature. These two grids were grown at their preferred curvature of $H = 0.42/L_0$, the curvature allowing the formation of the truncated icosahedron. We see that both grids are fully closed. Both cages are very close to a perfect TI.

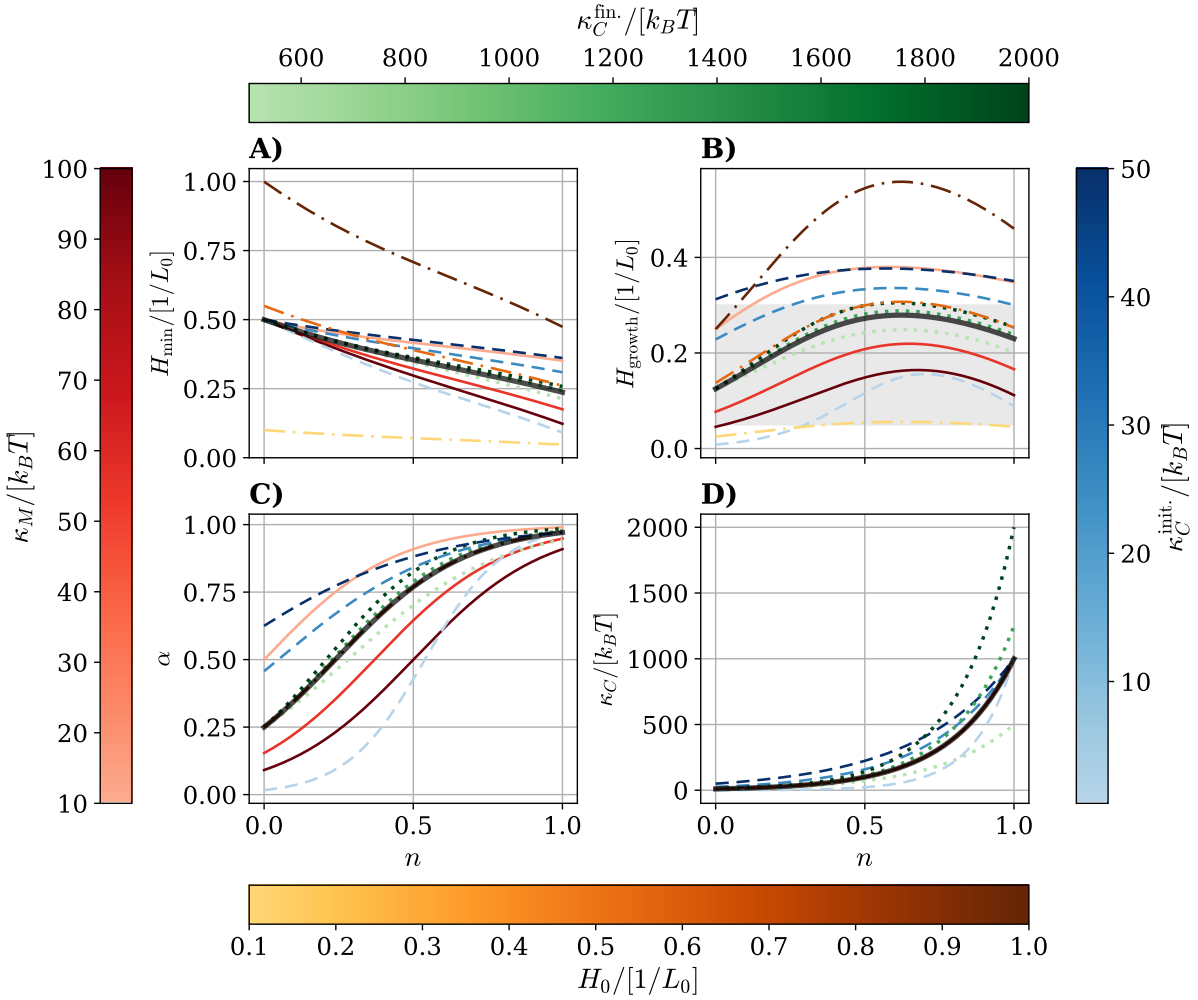

Figure B.4: Parameter evolution for variable curvature. Shown are the parameter evolutions with the final node fraction n for the model described in Table 4.1. Varied are the membranes bending rigidity κ_M (red, solid), the initial (blue, dashed) and final (green, dotted) coat bending rigidities $\kappa_C^{\text{init, fin.}}$ and the spontaneous curvature of clathrin, H_0 (yellow-orange, dash-dotted). All variations are indicated by colour. Standard values were $\kappa_M = 30\,\text{k}_\text{B}\text{T}$, $\kappa_C^{\text{init.}} = 10\,\text{k}_\text{B}\text{T}$, $\kappa_C^{\text{fin.}} = 1 \times 10^3\,\text{k}_\text{B}\text{T}$ and $H_0 = 0.5\,1/\text{L}_0$, shown by the solid black line. A) The evolution of the energy minimising curvature of the clathrin coat, following the evolution law of Equation (4.23). B) The evolution of the growth curvature, being the product of the bending rigidity ratio α and the energy minimising curvature of clathrin. Marked in grey is the range of physical final curvatures of CCVs, see Table 1.1. C) The evolution of the bending rigidity ratio $\alpha = \kappa_C/(\kappa_C + \kappa_M)$. D) The evolution of the effective coat bending rigidity $\kappa_C(n)$ according to (4.17).

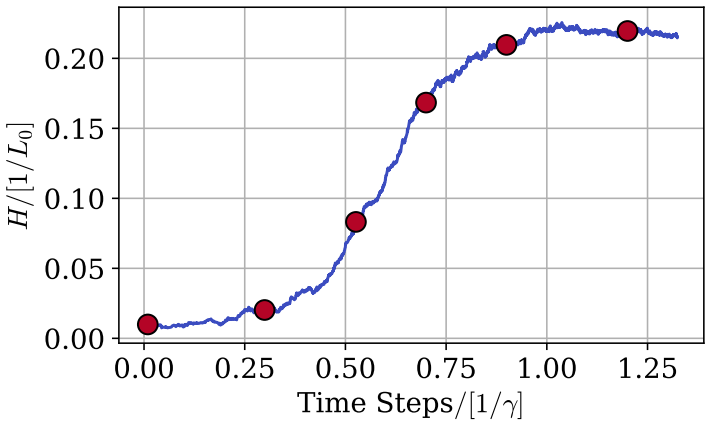

Figure B.5: Evaluation points for 3D visualisation. The evaluation points for the 3D visualisation of Figure 4.20.

Appendix C

Additional Data

Parameter	Value
total node number	140
oversampling ratio	$1e-05$
adiabatic relax steps	50
maximal curvature change	0.005
evolution steps	3000
beta	0.5
leg polymerization energy	7.0
hub polymerization energy	1.0
ideal sphere radius	2.4
ideal curvature	0.4166666666666667
initial sphere radius	3.125
initial curvature	0.32
k phi	32.828063500117445
k chi	32.828063500117445
k L	224.99999999999994
epsilon	$2e-06$
A cl membrane stiffness	0.0
sigma	0.7

Table C.1: Parameters for the growth of grids at fixed curvature. The parameters, used in section 4.4, are to be understood in their appropriate units as used in the code.

Parameter	Value
rng_seed	8086082
total_node_number	220
oversampling_ratio	1e-06
adiabatic_relax_steps	50
max_curvature_change	0.0001
evolution steps	6000
beta	1.0
leg_polymerization_energy	7.0
hub_polymerization_energy	1.0
ideal_sphere_radius	2.0
ideal_curvature	0.5
initial_sphere_radius	100.0
initial_curvature	0.01
k_phi	19.183674799192076
k_chi	19.183674799192076
k_L	799.9999999999999
epsilon	2e-07
kappa_membrane	20.0
sigma	0.7

Table C.2: Simulation parameters for grids with variable curvature. Shown are the simulation parameters used to simulate grids with variable curvature, as discussed in Section 4.5.

List of Figures

1.1 The structure of the lipid bilayer . . . 2
1.2 Vesicle transport between different compartments . . . 2
1.3 Where do different vesicular transport pathways take place . . . 3
1.4 The structure of clathrin and clathrin coated vesicles . . . 4
1.5 BAR-domain proteins . . . 6
1.6 Domed clathrin patch adjacent to flat clathrin structure . . . 7
1.7 Different stages of the cellular membrane during CME . . . 8
1.8 The constant area vs constant curvature model . . . 9
1.9 Evolution of geometric parameters against closing angle . . . 9
1.10 Pseudotime-resolution of clathrin coat evolution . . . 11

2.1 Pucker angle and its relation to the leg length. . . . 16
2.2 Combined bending energy of two membranes . . . 17
2.3 Geometry of a spherical cap . . . 19
2.4 An expression graph for the log-likelihood function . . . 30

3.1 Diagrammatic structure inside a clathrin grid . . . 34
3.2 Depiction of bonds between nodes . . . 35
3.3 Explanation of the single node geometry . . . 36
3.4 Wrapping method explained . . . 39
3.5 Azimuthal distance increase versus altitudinal position on the sphere . . . 40
3.6 Energy during radius changes . . . 43
3.7 Shift of equilibrium leg length L_0 . . . 44
3.8 Effective bending rigidity α against stiffness ratio . . . 46
3.9 Spring constants against polymerisation energy . . . 61
3.10 Increased polymerisation energies lead to decreased energies . . . 62
3.11 The dipping angle energy vs. curvature and bending radius . . . 63

4.1 Two-Dimensional Grid Visualisation . . . 66
4.2 3-Dimensional Grid Visualisation . . . 67
4.3 Parameter evolution during grid growth . . . 67
4.4 Example process contribution over time . . . 68
4.5 Example distribution histogram of acceptance probabilities . . . 69
4.6 MSD of clathrin node pairs showing Brownian motion . . . 70
4.7 The dodecahedron and truncated icosahedron . . . 71
4.8 A grown grid approaches a truncated icosahedron . . . 72
4.9 Representative examples of bending energy with its derivatives . . . 75
4.10 Number of nodes during growth at fixed curvature . . . 78

4.11 Average total energy during growth at fixed curvature 78
4.12 The average bond structure during growth at fixed curvature . . . 79
4.13 Polygon inclusion versus fraction of coat completeness 81
4.14 Effective bending rigidities over evolution time 82
4.15 Effective bending rigidities against node number 83
4.16 Evolution of the energy minimising curvature 84
4.17 Evolution of parameters for variable curvature 87
4.18 Closing angle against fractional node number and pseudotime . . 88
4.19 Parameter evolution for grids at variable curvature 89
4.20 The 3D evolution of a curved grid 91
4.21 Parameter evolution for stalled curvature onset 92

A.1 Angular Sampling 100
A.2 Uniform rejection sampling distribution 101
A.3 Inverse acceptance rate for rejection sampling 102
A.4 The ideal node number vs. curvature 104
A.5 Membrane curvature against clathrin coat curvature 104
A.6 Decay of energy after node addition 105

B.1 Data vs node number for grid approaching a TI 112
B.2 Exemplary grids grown away from preferred curvature 113
B.3 Exemplary grids grown at preferred curvature 114
B.4 Parameter evolution for variable curvature 115
B.5 Evaluation points for 3D visualisation 116

List of Tables

1.1 **Reference numbers for clathrin and related structures** 13

3.1 **Rate estimates for different processes** 51
3.2 **The free parameters for our simulation** 57

4.1 **A First Descriptive Model** 86

A.1 **Relations in different stereographic projections** 103

C.1 **Parameters for the growth of grids at fixed curvature** 117
C.2 **Simulation parameters for grids with variable curvature** 118

Bibliography

Aguet, François et al. (Aug. 2013). "Advances in Analysis of Low Signal-to-Noise Images Link Dynamin and AP2 to the Functions of an Endocytic Checkpoint". In: *Developmental Cell* 26.3, pp. 279–291. ISSN: 15345807. DOI: 10.1016/j.devcel.2013.06.019 (cit. on p. 5).

Alberts, Bruce et al. (2022). *Molecular Biology of the Cell*. Seventh edition, international student edition. New York, NY London: W. W. Norton & Company. 1404 pp. ISBN: 978-0-393-88482-1 978-0-393-88485-2 (cit. on pp. 2–4, 6).

Anderson, Richard G. W. et al. (Dec. 1977). "A Mutation That Impairs the Ability of Lipoprotein Receptors to Localise in Coated Pits on the Cell Surface of Human Fibroblasts". In: *Nature* 270.5639, pp. 695–699. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/270695a0 (cit. on p. 4).

Anderson, Richard G.W. et al. (Mar. 1977). "Role of the Coated Endocytic Vesicle in the Uptake of Receptor-Bound Low Density Lipoprotein in Human Fibroblasts". In: *Cell* 10.3, pp. 351–364. ISSN: 00928674. DOI: 10.1016/0092-8674(77)90022-8 (cit. on p. 4).

Arnold, Steven E. et al. (2005). "Neurodevelopment, Neuroplasticity, and New Genes for Schizophrenia". In: *Progress in Brain Research*. Vol. 147. Elsevier, pp. 319–345. ISBN: 978-0-444-51663-3. DOI: 10.1016/S0079-6123(04)47023-X (cit. on p. 3).

Avinoam, Ori et al. (June 19, 2015). "Endocytic Sites Mature by Continuous Bending and Remodeling of the Clathrin Coat". In: *Science* 348.6241, pp. 1369–1372. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aaa9555 (cit. on p. 8).

Bache, Kristi G et al. (July 21, 2004). "Defective Downregulation of Receptor Tyrosine Kinases in Cancer". In: *The EMBO Journal* 23.14, pp. 2707–2712. ISSN: 0261-4189, 1460-2075. DOI: 10.1038/sj.emboj.7600292 (cit. on p. 3).

Banerjee, Anand et al. (June 2012). "Stochastic Model of Clathrin-Coated Pit Assembly". In: *Biophysical Journal* 102.12, pp. 2725–2730. ISSN: 00063495. DOI: 10.1016/j.bpj.2012.05.010 (cit. on p. 6).

Baschieri, Francesco et al. (Sept. 20, 2018). "Frustrated Endocytosis Controls Contractility-Independent Mechanotransduction at Clathrin-Coated Structures". In: *Nature Communications* 9.1, p. 3825. ISSN: 2041-1723. DOI: 10.1038/s41467-018-06367-y (cit. on p. 5).

Binder, Kurt et al. (2010). *Monte Carlo Simulation in Statistical Physics: An Introduction*. Vol. 0. Graduate Texts in Physics. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-03162-5 978-3-642-03163-2. DOI: 10.1007/978-3-642-03163-2 (cit. on p. 22).

Bortz, A. B. et al. (Jan. 1, 1975). "A New Algorithm for Monte Carlo Simulation of Ising Spin Systems". In: *Journal of Computational Physics* 17.1, pp. 10–18. ISSN: 0021-9991. DOI: 10.1016/0021-9991(75)90060-1 (cit. on p. 26).

Brett, Tom J et al. (Aug. 2006). "Molecular Structures of Coat and Coat-Associated Proteins: Function Follows Form". In: *Current Opinion in Cell Biology* 18.4, pp. 395–406. ISSN: 09550674. DOI: 10.1016/j.ceb.2006.06.014 (cit. on p. 3).

Bucher, Delia et al. (Mar. 16, 2018). "Clathrin-Adaptor Ratio and Membrane Tension Regulate the Flat-to-Curved Transition of the Clathrin Coat during Endocytosis". In: *Nature Communications* 9.1, p. 1109. ISSN: 2041-1723. DOI: 10.1038/s41467-018-03533-0 (cit. on p. 10).

Campelo, Felix et al. (Apr. 10, 2014). "Sensing Membrane Stresses by Protein Insertions". In: *PLoS Computational Biology* 10.4. Ed. by David Van Der Spoel, e1003556. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003556 (cit. on p. 10).

Canham, P.B. (Jan. 1970). "The Minimum Energy of Bending as a Possible Explanation of the Biconcave Shape of the Human Red Blood Cell". In: *Journal of Theoretical Biology* 26.1, pp. 61–81. ISSN: 00225193. DOI: 10.1016/S0022-5193(70)80032-7 (cit. on p. 14).

Chen, Hong et al. (Aug. 1998). "Epsin Is an EH-domain-binding Protein Implicated in Clathrin-Mediated Endocytosis". In: *Nature* 394.6695, pp. 793–797. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/29555 (cit. on p. 10).

Cheng, Yifan et al. (Jan. 2007). "Cryo-Electron Tomography of Clathrin-coated Vesicles: Structural Implications for Coat Assembly". In: *Journal of Molecular Biology* 365.3, pp. 892–899. ISSN: 00222836. DOI: 10.1016/j.jmb.2006.10.036 (cit. on p. 4).

Cocucci, Emanuele, François Aguet, et al. (Aug. 2012). "The First Five Seconds in the Life of a Clathrin-Coated Pit". In: *Cell* 150.3, pp. 495–507. ISSN: 00928674. DOI: 10.1016/j.cell.2012.05.047 (cit. on p. 8).

Cocucci, Emanuele, Raphaël Gaudin, et al. (Nov. 5, 2014). "Dynamin Recruitment and Membrane Scission at the Neck of a Clathrin-Coated Pit". In: *Molecular Biology of the Cell* 25.22. Ed. by Jennifer Lippincott-Schwartz et al., pp. 3595–3609. ISSN: 1059-1524, 1939-4586. DOI: 10.1091/mbc.e14-07-1240 (cit. on p. 5).

Crowther, R A et al. (Dec. 1, 1981). "Assembly and Packing of Clathrin into Coats." In: *The Journal of cell biology* 91.3, pp. 790–797. ISSN: 0021-9525, 1540-8140. DOI: 10.1083/jcb.91.3.790 (cit. on pp. 13, 18).

Dawson, John C. et al. (Oct. 2006). "Bar Domain Proteins: A Role in Tubulation, Scission and Actin Assembly in Clathrin-Mediated Endocytosis". In: *Trends in Cell Biology* 16.10, pp. 493–498. ISSN: 09628924. DOI: 10.1016/j.tcb.2006.08.004 (cit. on p. 5).

Den Otter, Wouter K. et al. (Oct. 2011). "The Generation of Curved Clathrin Coats from Flat Plaques". In: *Traffic* 12.10, pp. 1407–1416. ISSN: 1398-9219, 1600-0854. DOI: 10.1111/j.1600-0854.2011.01241.x (cit. on pp. 13, 35, 36, 95).

Douc, Randal et al. (2018). *Markov Chains*. Springer Series in Operations Research and Financial Engineering. Cham: Springer International Publishing. ISBN: 978-3-319-97703-4 978-3-319-97704-1. DOI: 10.1007/978-3-319-97704-1 (cit. on p. 23).

Edeling, Melissa A. et al. (Jan. 1, 2006). "Life of a Clathrin Coat: Insights from Clathrin and AP Structures". In: *Nature Reviews Molecular Cell Biology* 7.1, pp. 32–44. ISSN: 1471-0072, 1471-0080. DOI: 10.1038/nrm1786 (cit. on p. 3).

Einstein, A. (Jan. 1905). "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen". In: *Annalen der Physik* 322.8, pp. 549–560. ISSN: 0003-3804, 1521-3889. DOI: 10.1002/andp.19053220806 (cit. on p. 70).

Eto, Danelle S. et al. (Dec. 2008). "Clathrin, AP-2, and the NPXY-binding Subset of Alternate Endocytic Adaptors Facilitate FimH-mediated Bacterial Invasion of Host

Cells". In: *Cellular Microbiology* 10.12, pp. 2553–2567. ISSN: 14625814, 14625822. DOI: 10.1111/j.1462-5822.2008.01229.x (cit. on p. 3).

Evans, E. et al. (Apr. 23, 1990). "Entropy-Driven Tension and Bending Elasticity in Condensed-Fluid Membranes". In: *Physical Review Letters* 64.17, pp. 2094–2097. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.64.2094 (cit. on p. 13).

Ford, Marijn G. J. et al. (Sept. 2002). "Curvature of Clathrin-Coated Pits Driven by Epsin". In: *Nature* 419.6905, pp. 361–366. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature01020 (cit. on p. 10).

Fotin, Alexander et al. (Dec. 2004). "Molecular Model for a Complete Clathrin Lattice from Electron Cryomicroscopy". In: *Nature* 432.7017, pp. 573–579. ISSN: 1476-4687. DOI: 10.1038/nature03079 (cit. on pp. 4, 7, 71).

Frey, Felix, Delia Bucher, et al. (July 1, 2020). "Eden Growth Models for Flat Clathrin Lattices with Vacancies". In: *New Journal of Physics* 22.7, p. 073043. ISSN: 1367-2630. DOI: 10.1088/1367-2630/ab99e1 (cit. on pp. 7, 35).

Frey, Felix and Ulrich S. Schwarz (2020). "Competing Pathways for the Invagination of Clathrin-Coated Membranes". In: *Soft Matter* 16.47, pp. 10723–10733. ISSN: 1744-683X, 1744-6848. DOI: 10.1039/D0SM01375G (cit. on p. 10).

— (Dec. 10, 2024). "Coat Stiffening Can Explain Invagination of Clathrin-Coated Membranes". In: *Physical Review E* 110.6, p. 064403. ISSN: 2470-0045, 2470-0053. DOI: 10.1103/PhysRevE.110.064403 (cit. on pp. 10, 20, 94, 106).

Frey, Felix, Falko Ziebert, et al. (Nov. 14, 2019). "Dynamics of Particle Uptake at Cell Membranes". In: *Physical Review E* 100.5, p. 052403. ISSN: 2470-0045, 2470-0053. DOI: 10.1103/PhysRevE.100.052403 (cit. on p. 5).

Gillespie, Daniel T. (Dec. 1, 1977). "Exact Stochastic Simulation of Coupled Chemical Reactions". In: *The Journal of Physical Chemistry* 81.25, pp. 2340–2361. ISSN: 0022-3654. DOI: 10.1021/j100540a008 (cit. on p. 26).

Gittes, F et al. (Feb. 15, 1993). "Flexural Rigidity of Microtubules and Actin Filaments Measured from Thermal Fluctuations in Shape." In: *The Journal of cell biology* 120.4, pp. 923–934. ISSN: 0021-9525, 1540-8140. DOI: 10.1083/jcb.120.4.923 (cit. on p. 18).

Goldstein, Joseph L. et al. (June 1979). "Coated Pits, Coated Vesicles, and Receptor-Mediated Endocytosis". In: *Nature* 279.5715, pp. 679–685. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/279679a0 (cit. on p. 4).

Hanada, Masanori et al. (2022). *MCMC from Scratch: A Practical Introduction to Markov Chain Monte Carlo*. Singapore: Springer Nature Singapore. ISBN: 978-981-19-2714-0 978-981-19-2715-7. DOI: 10.1007/978-981-19-2715-7 (cit. on p. 22).

Haucke, Volker et al. (Sept. 1, 2018). "Membrane Remodeling in Clathrin-Mediated Endocytosis". In: *Journal of Cell Science* 131.17, jcs216812. ISSN: 1477-9137, 0021-9533. DOI: 10.1242/jcs.216812 (cit. on pp. 3, 6, 8).

Helfrich, W (Dec. 1, 1973). "Elastic Properties of Lipid Bilayers: Theory and Possible Experiments". In: *Zeitschrift für Naturforschung C* 28.11–12, pp. 693–703. ISSN: 1865-7125, 0939-5075. DOI: 10.1515/znc-1973-11-1209 (cit. on pp. 6, 14).

Henkelman, Graeme et al. (Dec. 1, 2001). "Long Time Scale Kinetic Monte Carlo Simulations without Lattice Approximation and Predefined Event Table". In: *The Journal of Chemical Physics* 115.21, pp. 9657–9666. ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.1415500 (cit. on p. 27).

Heuser, J (Mar. 1, 1980). "Three-Dimensional Visualization of Coated Vesicle Formation in Fibroblasts." In: *The Journal of cell biology* 84.3, pp. 560–583. ISSN: 0021-9525, 1540-8140. DOI: 10.1083/jcb.84.3.560 (cit. on pp. 5, 8, 22, 90).

— (Feb. 1, 1989). "Effects of Cytoplasmic Acidification on Clathrin Lattice Morphology." In: *The Journal of cell biology* 108.2, pp. 401–411. ISSN: 0021-9525, 1540-8140. DOI: 10.1083/jcb.108.2.401 (cit. on pp. 8, 90).

Heuser, J. et al. (July 1985). "Deep-Etch Views of Clathrin Assemblies". In: *Journal of Ultrastructure Research* 92.1–2, pp. 1–27. ISSN: 00225320. DOI: 10.1016/0889-1605(85)90123-5 (cit. on pp. 8, 13).

Jansen, A.P.J. (2012). *An Introduction to Kinetic Monte Carlo Simulations of Surface Reactions*. Vol. 856. Lecture Notes in Physics. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-29487-7 978-3-642-29488-4. DOI: 10.1007/978-3-642-29488-4 (cit. on p. 22).

Jin, Albert J et al. (Oct. 1993). "Topological Mechanisms Involved in the Formation of Clathrin-Coated Vesicles." In: *Biophysical Journal* 65.4, pp. 1523–1537. ISSN: 0006-3495. PMID: 8274646 (cit. on pp. 7, 8, 22).

Jin, Albert J. and Ralph Nossal (Mar. 2000). "Rigidity of Triskelion Arms and Clathrin Nets". In: *Biophysical Journal* 78.3, pp. 1183–1194. ISSN: 00063495. DOI: 10.1016/S0006-3495(00)76676-8 (cit. on pp. 12, 13, 18, 60).

Jin, Albert J., Kondury Prasad, et al. (May 2006). "Measuring the Elasticity of Clathrin-Coated Vesicles via Atomic Force Microscopy". In: *Biophysical Journal* 90.9, pp. 3333–3344. ISSN: 00063495. DOI: 10.1529/biophysj.105.068742 (cit. on pp. 8, 12, 13, 59).

Kaksonen, Marko et al. (May 2018). "Mechanisms of Clathrin-Mediated Endocytosis". In: *Nature Reviews Molecular Cell Biology* 19.5, pp. 313–326. ISSN: 1471-0072, 1471-0080. DOI: 10.1038/nrm.2017.132 (cit. on p. 3).

Kingma, Diederik P. et al. (Jan. 30, 2017). *Adam: A Method for Stochastic Optimization*. DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980 [cs]. URL: http://arxiv.org/abs/1412.6980 (visited on 10/01/2025). Pre-published (cit. on p. 31).

Kirchhausen, T. et al. (May 1, 2014). "Molecular Structure, Function, and Dynamics of Clathrin-Mediated Membrane Traffic". In: *Cold Spring Harbor Perspectives in Biology* 6.5, a016725–a016725. ISSN: 1943-0264. DOI: 10.1101/cshperspect.a016725 (cit. on p. 3).

Kirchhausen, Tom (Nov. 2009). "Imaging Endocytic Clathrin Structures in Living Cells". In: *Trends in Cell Biology* 19.11, pp. 596–605. ISSN: 09628924. DOI: 10.1016/j.tcb.2009.09.002 (cit. on pp. 6, 8, 13).

Kroto, H. W. et al. (Nov. 1985). "C60: Buckminsterfullerene". In: *Nature* 318.6042, pp. 162–163. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/318162a0 (cit. on p. 71).

Lampe, Marko et al. (Oct. 2016). "Clathrin Coated Pits, Plaques and Adhesion". In: *Journal of Structural Biology* 196.1, pp. 48–56. ISSN: 10478477. DOI: 10.1016/j.jsb.2016.07.009 (cit. on p. 5).

Landau, David P. et al. (2015). *A Guide to Monte Carlo Simulations in Statistical Physics*. Fourth edition. Cambridge: Cambridge University Press. 1 p. ISBN: 978-1-107-07402-6 978-1-139-69646-3. DOI: 10.1017/CBO9781139696463 (cit. on pp. 22, 47).

Larkin, J M et al. (Dec. 1, 1986). "Potassium-Dependent Assembly of Coated Pits: New Coated Pits Form as Planar Clathrin Lattices." In: *The Journal of cell biology* 103.6, pp. 2619–2627. ISSN: 0021-9525, 1540-8140. DOI: 10.1083/jcb.103.6.2619 (cit. on pp. 8, 90).

Lifšic, Evgenij M. et al. (1986). *Theory of Elasticity*. 3., Engl. ed., rev. and enl. Course of Theoretical Physics / L. D. Landau and E. M. Lifshitz 7. Oxford New York Beijing Frankfurt [Main i.e.] Kronberg São Paulo Sydney Tokyo Toronto: Pergamon Press. 187 pp. ISBN: 978-0-08-033917-7 978-0-08-033916-0 (cit. on p. 18).

Linggi, B et al. (Dec. 2006). "ErbB Receptors: New Insights on Mechanisms and Biology". In: *Trends in Cell Biology* 16.12, pp. 649–656. ISSN: 09628924. DOI: 10.1016/j.tcb.2006.10.008 (cit. on p. 3).

Loerke, Dinah et al. (Mar. 17, 2009). "Cargo and Dynamin Regulate Clathrin-Coated Pit Maturation". In: *PLoS Biology* 7.3. Ed. by Fred Hughson, e1000057. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1000057 (cit. on pp. 6, 13).

Macia, Eric et al. (June 2006). "Dynasore, a Cell-Permeable Inhibitor of Dynamin". In: *Developmental Cell* 10.6, pp. 839–850. ISSN: 15345807. DOI: 10.1016/j.devcel.2006.04.002 (cit. on p. 5).

Maib, Hannes et al. (Apr. 2017). "Forty Years on: Clathrin-Coated Pits Continue to Fascinate". In: *Molecular Biology of the Cell* 28.7. Ed. by David G. Drubin, pp. 843–847. ISSN: 1059-1524, 1939-4586. DOI: 10.1091/mbc.e16-04-0213 (cit. on p. 3).

Margossian, Charles C. (July 2019). "A Review of Automatic Differentiation and Its Efficient Implementation". In: *WIREs Data Mining and Knowledge Discovery* 9.4, e1305. ISSN: 1942-4787, 1942-4795. DOI: 10.1002/widm.1305 (cit. on pp. 28–30).

Maritzen, Tanja et al. (Oct. 2012). "Turning CALM into Excitement: AP180 and CALM in Endocytosis and Disease". In: *Biology of the Cell* 104.10, pp. 588–602. ISSN: 0248-4900, 1768-322X. DOI: 10.1111/boc.201200008 (cit. on p. 10).

McMahon, Harvey T. and Emmanuel Boucrot (Aug. 2011). "Molecular Mechanism and Physiological Functions of Clathrin-Mediated Endocytosis". In: *Nature Reviews Molecular Cell Biology* 12.8, pp. 517–533. ISSN: 1471-0072, 1471-0080. DOI: 10.1038/nrm3151 (cit. on p. 3).

McMahon, Harvey T. and Jennifer L. Gallop (Dec. 2005). "Membrane Curvature and Mechanisms of Dynamic Cell Membrane Remodelling". In: *Nature* 438.7068, pp. 590–596. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature04396 (cit. on p. 10).

Merrifield, Christien J. et al. (Sept. 2002). "Imaging Actin and Dynamin Recruitment during Invagination of Single Clathrin-Coated Pits". In: *Nature Cell Biology* 4.9, pp. 691–698. ISSN: 1465-7392, 1476-4679. DOI: 10.1038/ncb837 (cit. on p. 5).

Metropolis, Nicholas et al. (June 1, 1953). "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6, pp. 1087–1092. ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.1699114 (cit. on p. 25).

Mund, Markus et al. (Mar. 6, 2023). "Clathrin Coats Partially Preassemble and Subsequently Bend during Endocytosis". In: *Journal of Cell Biology* 222.3, e202206038. ISSN: 0021-9525, 1540-8140. DOI: 10.1083/jcb.202206038 (cit. on pp. 9–11, 19, 84, 86, 88, 94, 97).

Nossal, Ralph (Jan. 2001). "Energetics of Clathrin Basket Assembly". In: *Traffic* 2.2, pp. 138–147. ISSN: 1398-9219, 1600-0854. DOI: 10.1034/j.1600-0854.2001.020208.x (cit. on p. 13).

Pearse, Barbara M.F. (Sept. 1975). "Coated Vesicles from Pig Brain: Purification and Biochemical Characterization". In: *Journal of Molecular Biology* 97.1, pp. 93–98. ISSN: 00222836. DOI: 10.1016/S0022-2836(75)80024-6 (cit. on p. 3).

Polo, Simona et al. (Apr. 2004). "Endocytosis and Cancer". In: *Current Opinion in Cell Biology* 16.2, pp. 156–161. ISSN: 09550674. DOI: 10.1016/j.ceb.2004.02.003 (cit. on p. 3).

Ramanan, Vyas et al. (2011). "Systems Biology and Physical Biology of Clathrin-Mediated Endocytosis". In: *Integrative Biology* 3.8, p. 803. ISSN: 1757-9694, 1757-9708. DOI: 10.1039/c1ib00036e (cit. on p. 3).

Rawicz, W. et al. (July 2000). "Effect of Chain Length and Unsaturation on Elasticity of Lipid Bilayers". In: *Biophysical Journal* 79.1, pp. 328–339. ISSN: 00063495. DOI: 10.1016/S0006-3495(00)76295-3 (cit. on p. 13).

Roth, Thomas F. et al. (Feb. 1, 1964). "Yolk Protein Uptake in the Oocyte of the Mosquito *AEDES AEGYPTI* . L". In: *The Journal of Cell Biology* 20.2, pp. 313–332. ISSN: 1540-8140, 0021-9525. DOI: 10.1083/jcb.20.2.313 (cit. on p. 3).

Ruzayqat, Hamza M. et al. (Jan. 9, 2018). "A Rejection Scheme for Off-Lattice Kinetic Monte Carlo Simulation". In: *Journal of Chemical Theory and Computation* 14.1, pp. 48–54. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/acs.jctc.7b00974 (cit. on p. 47).

Sadiq, Abdullah (Sept. 1984). "A New Algorithm for the Monte Carlo Simulation of Spin-Exchange Kinetics of Ising Systems". In: *Journal of Computational Physics* 55.3, pp. 387–396. ISSN: 00219991. DOI: 10.1016/0021-9991(84)90028-7 (cit. on p. 108).

Saffarian, Saveez et al. (Sept. 8, 2009). "Distinct Dynamics of Endocytic Clathrin-Coated Pits and Coated Plaques". In: *PLoS Biology* 7.9. Ed. by Fred Hughson, e1000191. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1000191 (cit. on p. 8).

Saleem, Mohammed et al. (Feb. 19, 2015). "A Balance between Membrane Elasticity and Polymerization Energy Sets the Shape of Spherical Clathrin Coats". In: *Nature Communications* 6.1, p. 6249. ISSN: 2041-1723. DOI: 10.1038/ncomms7249 (cit. on pp. 12, 13).

Scott, Brandon L. et al. (Jan. 29, 2018). "Membrane Bending Occurs at All Stages of Clathrin-Coat Assembly and Defines Endocytic Dynamics". In: *Nature Communications* 9.1, p. 419. ISSN: 2041-1723. DOI: 10.1038/s41467-018-02818-8 (cit. on p. 10).

Serebrinsky, Santiago A. (Mar. 31, 2011). "Physical Time Scale in Kinetic Monte Carlo Simulations of Continuous-Time Markov Chains". In: *Physical Review E* 83.3, p. 037701. ISSN: 1539-3755, 1550-2376. DOI: 10.1103/PhysRevE.83.037701 (cit. on pp. 27, 108).

Sochacki, Kem A. et al. (Apr. 2021). "The Structure and Spontaneous Curvature of Clathrin Lattices at the Plasma Membrane". In: *Developmental Cell* 56.8, 1131–1146.e3. ISSN: 15345807. DOI: 10.1016/j.devcel.2021.03.017 (cit. on pp. 7, 13, 84, 90, 95).

Sorger, P K et al. (Oct. 1, 1986). "Clathrin Cubes: An Extreme Variant of the Normal Cage." In: *The Journal of cell biology* 103.4, pp. 1213–1219. ISSN: 0021-9525, 1540-8140. DOI: 10.1083/jcb.103.4.1213 (cit. on p. 8).

Sorkin, Alexander et al. (Aug. 2002). "Signal Transduction and Endocytosis: Close Encounters of Many Kinds". In: *Nature Reviews Molecular Cell Biology* 3.8, pp. 600–614. ISSN: 1471-0072, 1471-0080. DOI: 10.1038/nrm883 (cit. on p. 1).

Tagiltsev, Grigory et al. (Aug. 13, 2021). "Nanodissected Elastically Loaded Clathrin Lattices Relax to Increased Curvature". In: *Science Advances* 7.33, eabg9934. ISSN: 2375-2548. DOI: 10.1126/sciadv.abg9934 (cit. on pp. 4, 90, 106).

Taylor, Marcus J., Marko Lampe, et al. (Apr. 10, 2012). "A Feedback Loop between Dynamin and Actin Recruitment during Clathrin-Mediated Endocytosis". In: *PLoS Biology* 10.4. Ed. by Sandra L. Schmid, e1001302. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1001302 (cit. on p. 5).

Taylor, Marcus J., David Perrais, et al. (Mar. 22, 2011). "A High Precision Survey of the Molecular Dynamics of Mammalian Clathrin-Mediated Endocytosis". In: *PLoS Biology* 9.3. Ed. by Sandra L. Schmid, e1000604. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1000604 (cit. on p. 5).

Ungewickell, Ernst et al. (1981). "Assembly Units of cIathrin Coats". In: *Nature* 289 (cit. on p. 4).

Vigers, G.P. et al. (Mar. 1986). "Three-Dimensional Structure of Clathrin Cages in Ice." In: *The EMBO Journal* 5.3, pp. 529–534. ISSN: 02614189. DOI: 10.1002/j.1460-2075.1986.tb04242.x (cit. on pp. 13, 22).

von Kleist, Lisa et al. (Aug. 2011). "Role of the Clathrin Terminal Domain in Regulating Coated Pit Dynamics Revealed by Small Molecule Inhibition". In: *Cell* 146.3, pp. 471–484. ISSN: 00928674. DOI: 10.1016/j.cell.2011.06.025 (cit. on p. 3).

Von Neumann, John (1951). *Various Techniques Used in Connection With Random Digits, Notes by G. E. Forsythe*. National Bureau of Standards Applied Math Series, 12 (1951) pp 36-38 (cit. on p. 98).

Wilson, Robin J. (2009). *Introduction to Graph Theory*. 4. ed., [Nachdr.] Harlow Munich: Prentice Hall. 171 pp. ISBN: 978-0-582-24993-6 (cit. on p. 21).

Yoshida, Aiko et al. (May 3, 2018). "Morphological Changes of Plasma Membrane and Protein Assembly during Clathrin-Mediated Endocytosis". In: *PLOS Biology* 16.5. Ed. by Sandra Schmid, e2004786. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.2004786 (cit. on p. 10).

Zeno, Wade F. et al. (Mar. 2021). "Clathrin Senses Membrane Curvature". In: *Biophysical Journal* 120.5, pp. 818–828. ISSN: 00063495. DOI: 10.1016/j.bpj.2020.12.035 (cit. on p. 106).

Zhang, Sulin et al. (Sept. 22, 2015). "Physical Principles of Nanoparticle Cellular Endocytosis". In: *ACS Nano* 9.9, pp. 8655–8671. ISSN: 1936-0851, 1936-086X. DOI: 10.1021/acsnano.5b03184 (cit. on p. 3).

Zhu, Cuncheng et al. (Sept. 2022). "Mem3DG: Modeling Membrane Mechanochemical Dynamics in 3D Using Discrete Differential Geometry". In: *Biophysical Reports* 2.3, p. 100062. ISSN: 26670747. DOI: 10.1016/j.bpr.2022.100062 (cit. on p. 15).

Zimmerberg, Joshua et al. (Jan. 1, 2006). "How Proteins Produce Cellular Membrane Curvature". In: *Nature Reviews Molecular Cell Biology* 7.1, pp. 9–19. ISSN: 1471-0072, 1471-0080. DOI: 10.1038/nrm1784 (cit. on p. 10).

Declaration of AI-assisted Tools Usage:

This thesis was partially prepared with the assistance of AI-based tools. Language models such as ChatGPT and NotebookLM were used to generate preliminary summaries of texts, which were subsequently reviewed and verified. Code was completed, debugged, and improved with the support of Microsoft Copilot, but never in a way that treated the generated code as a "black box." No graphics or images were generated using AI. To enhance readability, the text was also reviewed and, where appropriate, refined using language models.

Erklärung:

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 30.11.2025