

Physics Area - PhD course in Physics and Chemistry of Biological
Systems

Coherent statistics of branched polymers
and double-folded rings
Exact results and algorithms

Candidate:
Pieter H.W. van der Hoek

Advisor:
Angelo Rosa
Co-advisor:
Ralf Everaers

Academic Year 2025-2026



Abstract

At a sufficiently coarse-grained level, many polymers adopt a branched, tree-like organization. In polymer physics, such structures were first studied in the context of chemically branched molecules, while more recently branched structures have emerged as useful coarse-grained descriptions of nonconcatenated ring polymers in melts. Related tree-like representations have also been used to describe supercoiled and looped DNA and the secondary structures of RNA.

Motivated by these observations, we study the statistical mechanics of the corresponding branched-polymer ensembles. We consider randomly branched polymers, whose configurations are specified by an underlying tree connectivity, and tightly double-folded rings, for which a sequentially labelled contour additionally wraps around a branched backbone.

We first study both ensembles in the ideal limit, where spatial interactions are neglected. In this case, their configurational multiplicities can be counted exactly through one-to-one correspondences with sequences of integers. Randomly branched polymers are related to graph-theoretical labelled trees through Prüfer codes, whereas tightly double-folded rings are represented by a new encoding which we call the wrapping code. These correspondences yield exact finite-size multiplicities and partition functions at controlled branching activity and allow us to establish a quantitative mapping between the tree and ring ensembles.

We then use these exact statistical weights to develop efficient numerical sampling strategies. Ideal trees can be generated independently in $\mathcal{O}(N_{\text{tree}})$ operations. When spatial interactions prevent direct sampling, we generalize Monte Carlo leaf-mover algorithms to remain efficient over the full range of branching activity. Finally, the mapping between trees and rings allows equilibrium statistics of interacting tightly double-folded rings to be obtained through tree-level sampling. When excluded volume interactions are applied consistently in both ensembles, this method provides a speedup by at least a factor of order N_{ring} for equilibrating double-folded ring melts.

These results establish a coherent statistical and computational framework connecting annealed tree connectivity and tightly double-folded rings. Explicitly accounting for the configurational entropy associated with changing representation provides both quantitative relations between these ensembles and efficient routes for sampling them in ideal ensembles as well as in ensembles with volume effects.

Contents

Abstract	i
1 Introduction	1
1.1 Configurational entropy and coarse-graining	1
1.2 Tree-like organization in polymer physics	2
1.2.1 Randomly branched polymers	2
1.2.2 Topologically constrained ring polymers	2
1.3 Tree-like organization in biopolymers	4
1.3.1 Bacterial DNA	5
1.3.2 Eukaryotic DNA during interphase	5
1.3.3 Secondary structures of RNA molecules	6
1.4 Aim and organization of the thesis	6
2 Statistical physics of polymers and ring melts	8
2.1 Ideal linear chains	9
2.1.1 The end-to-end distance	9
2.1.2 Gyration radius	11
2.1.3 Kramers' theorem	12
2.2 Ring polymers	13
2.2.1 Conformational multiplicity of a ring polymer	13
2.2.2 Ring closure as contact probability	14
2.2.3 The size of ideal rings	14
2.3 Interacting linear chains	15
2.3.1 Self-avoiding walks	15
2.3.2 Melts of linear chains	16
2.3.3 Coil-to-globule transition	16
2.3.4 Contact probabilities for interacting linear chains	16
2.4 Tree-like descriptions of ring melts	17
2.4.1 Double folding and branched primitive paths	18
2.4.2 Scope of the double-folded approximation	18
2.5 Chapter summary	19

3	Exact configurational statistics of randomly branched polymers	20
3.1	Tree ensemble, constraints, and statistical weights	20
3.1.1	Functionality constraints	21
3.1.2	Controlled activity	22
3.1.3	Ideal and interacting tree ensembles	22
3.2	Exact configurational entropy of randomly branched polymers	23
3.2.1	The total number of labelled trees	24
3.2.2	Counting trees with specified functionalities	25
3.2.3	The partition function for ideal trees	25
3.3	Labelled and unlabelled tree ensembles	26
3.4	Structural consequences of the exact tree ensemble	27
3.4.1	Radius of gyration at fixed branching	27
3.4.2	Low-branching activity and the Daoud–Joanny form	29
3.5	Branching activity over the full branching range	30
3.6	Chapter summary	31
4	Tightly double-folded rings and the tree–ring correspondence	33
4.1	Ring representation and notation	33
4.1.1	Ring size and stored length	33
4.1.2	Ring configurations and conformations	34
4.1.3	Branching activity and spatial interactions	34
4.2	Exact configurational entropy of tightly double-folded rings	35
4.2.1	Wrapped trees and a wrapping code	35
4.2.2	The total number of viable wrapping codes	36
4.2.3	The partition function for ideal tightly double-folded rings	37
4.2.4	Accounting for the reptons in the elastic lattice model	37
4.3	Exact correspondence between tree and ring ensembles	38
4.3.1	The multiplicity of labelled trees revisited	38
4.3.2	Wrapping labelled trees	39
4.3.3	Quantitative link between tree and ring ensembles	39
4.3.4	Including spatial interactions	41
4.3.5	Arbitrary node functionalities	41
4.4	Measures for elastic rings	42
4.4.1	Branching activity for rings	42
4.4.2	Repton distribution according to tree nodes’ functionality	43
4.5	Chapter summary	43
5	Sampling algorithms for branched polymers and double-folded rings	45
5.1	Direct Prüfer sampling of ideal trees	45
5.1.1	Unconstrained labelled trees	46
5.1.2	Functionality constraint $f \leq 3$	46

5.1.3	Arbitrary functionality constraints	46
5.1.4	Sampling other codes	47
5.2	Amoeba sampling at fixed tree size	47
5.2.1	Original Amoeba algorithm	48
5.2.2	Semi-kinetic Amoeba algorithm	49
5.2.3	Double-leaf Amoeba algorithm	50
5.2.4	Generalization to other Hamiltonians	51
5.3	Local Monte Carlo dynamics of tightly double-folded rings	51
5.3.1	Local trial moves on the ring level	51
5.3.2	Extensions of the local move set	52
5.4	Semi-grand-canonical tree sampling of fixed-length rings	53
5.4.1	Monte Carlo move classes	53
5.4.2	Proposal and acceptance probabilities	53
5.4.3	The single-node boundary	54
5.4.4	Complete semi-grand-canonical algorithm	55
5.4.5	Generalizations	56
5.5	Alternative tree-sampling algorithms and their ensembles	56
5.5.1	Non-local branch-moving algorithms	56
5.5.2	Growth and sequential-importance algorithms	57
5.5.3	Uniform spanning trees and Wilson’s algorithm	58
5.6	Chapter summary	58
6	Validation and ensemble equivalence	60
6.1	Validation of fixed-size Amoeba tree sampling	60
6.1.1	Historical benchmark	60
6.1.2	Consistency between Amoeba variants across branching activity . . .	61
6.1.3	Recovering the labelled-tree configurational entropy	62
6.1.4	Excluding unlabelled trees	62
6.2	Structural observables of the exact ideal-tree ensemble	65
6.2.1	Radius of gyration at fixed branching	65
6.2.2	Full crossover controlled by the branching activity	67
6.3	Validation of elastic double-folded-ring statistics	67
6.3.1	Joint distribution of tree size and branching	67
6.3.2	Tree size and functionality averages	70
6.3.3	Stored-length distribution	70
6.4	Equivalence of tree-level and ring-level sampling	71
6.5	Chapter summary	74
7	Performance and equilibration dynamics of the sampling algorithms	75
7.1	Equilibration criteria and reference time scales	76
7.1.1	Centre-of-mass diffusion as an internal equilibration diagnostic . . .	76

7.1.2	Naive reptation-like scaling for leaf movers	77
7.1.3	Monte Carlo steps, CPU time, and independent samples	77
7.2	Fixed-size Amoeba algorithms with controlled branching	77
7.2.1	Weakly branched trees: rejection of side-branching moves	79
7.2.2	Strongly branched trees: mobility of the $f=2$ defect	79
7.3	Effective trial moves and the effective-move time t_0	80
7.4	A universal tree regime in Amoeba dynamics	81
7.5	Direct Prüfer sampling bypasses equilibration	85
7.6	Equilibration of tightly double-folded rings	87
7.6.1	Systems, initial states, and observables	87
7.6.2	Relaxation from compact configurations	87
7.6.3	CPU-time scaling	90
7.7	Extensions, alternative move sets, and limitations	90
7.7.1	Alternative transition-state constructions	91
7.7.2	Higher functionality	91
7.7.3	Spatial interactions	91
7.8	Practical synthesis and chapter conclusion	92
8	Concluding Remarks	93
8.1	Main Findings	93
8.2	Discussion	94
8.2.1	Application to biological polymers	94
8.2.2	Application to a model of melts of rings	94
8.2.3	Labelled versus unlabelled trees	95
8.3	Outlook	96
	Acknowledgements	98
A	Algorithm visualisations and generalizations	103
A.1	Constructing a Prüfer code from a connectivity graph	104
A.2	Constructing a connectivity graph from a Prüfer code	105
A.3	Sampling Prüfer codes	106
A.4	Constructing a tree wrapping and its corresponding code	107
A.5	Decoding a wrapping code	108
A.6	Decoding invalid and valid wrapping codes	109
A.7	Using recursions for sampling trees with arbitrary functionality constraints .	109
B	Combinatorial result for wrapping codes	111
B.1	Bertrand's ballot theorem	111

C Asymptotic behaviour of tightly double-folded rings	114
C.1 Partition function for tightly double-folded rings	114
C.2 Approximate mean tree size with reptons	116
C.3 Tightly double-folded rings of arbitrary functionality	116
Bibliography	119

Chapter 1

Introduction

Many polymers in biology have a simple linear covalent organization at small scales, but an increasingly non-trivial structure at larger length scales. DNA and RNA are synthesized as continuous chains that remain linear or circular, yet supercoiling, folding, loop formation, and secondary-structure interactions can divide these chains into arms, junctions, and terminal domains. Once microscopic details are coarse-grained, such structures can often be represented as randomly branched, tree-like polymers.

This thesis studies the statistical mechanics of these effective tree-like states: How many configurations are available, how are different ensembles of branching structures related, and how can they be sampled efficiently.

1.1 Configurational entropy and coarse-graining

Statistical physics replaces a detailed microscopic description by a smaller set of variables that retain the information relevant to the question of interest. The corresponding loss of microscopic detail is quantified by Boltzmann’s relation

$$S(i) = k_B \ln(\Omega(i)), \quad (1.1)$$

where $\Omega(i)$ is the number of microstates compatible with a macrostate i , and $S(i)$ is its entropy. The entropy of a polymer can therefore be computed in several different ways, depending on how the system of interest is coarse-grained.

We are predominantly interested in the *configurational* degrees of freedom and their associated entropy. We use the term “configurational” to signify a polymer configuration before spatial embedding. A specific example of a representation of a polymer configuration is the connectivity matrix of individual labelled monomers for a specific branched polymer. We use the *conformational* entropy to quantify the number of ways a polymer configuration can then be embedded in space.

Under the right level of coarse-graining for Eq. (1.1), the configurational and conformational entropy can factorize cleanly. However, in practice, the categorization of microstates

into configurational macrostates independently of their conformational entropy is highly non-trivial in most physical situations. Two configurational microstates belonging to the same macrostate can have vastly different conformational entropies. In genome organization, spatial and topological [1] effects can strongly favour certain configurations while suppressing or excluding others. More generally, exact results are rare once spatial interactions such as excluded volume or confinement are considered.

We therefore often work in the ideal ensemble. We omit any form of spatial interaction, such that in the ideal models considered here the conformational contribution does not depend on the configuration of a fixed number of monomers N . At first, this seems like a highly unphysical assumption for studying the systems of interest. After all, spatial interactions fold polymers into their observed shapes, and excluding any form of spatial interaction removes the very physical effect that our models are based on. However, in the present simplified situations, exact calculations are possible across multiple ensembles, such that we can compare those ensembles to arbitrary precision. Therefore, we use the ideal ensemble as a necessary reference point for comparing randomly branched polymeric systems.

1.2 Tree-like organization in polymer physics

The tree-like configurations considered in this thesis arise in two closely related polymer-physics settings: randomly branched polymers and topologically constrained rings.

1.2.1 Randomly branched polymers

A randomly branched polymer (Fig. 1.1(b1)) is a polymer whose configuration can be represented by a connected graph without cycles, i.e. a tree. Its configuration is specified by the connectivity of the monomers, independently of how this connectivity is subsequently embedded in space.

Randomly branched polymers have a long history in polymer physics. The classical work of Flory, Zimm, and Stockmayer [2, 3, 4, 5, 6] considered chemically branched polymers, for which the connectivity is effectively *quenched* once the bonds are formed. The exact results of Zimm and Stockmayer will be recovered in Chapter 3.

Here, instead, we are primarily interested in branching with *annealed* connectivity [7, 8]. In this case branches can grow, shrink, or reorganize, so that the connectivity itself becomes a fluctuating configurational degree of freedom. The different tree connectivities therefore form a statistical ensemble whose multiplicity contributes directly to the configurational entropy introduced above.

1.2.2 Topologically constrained ring polymers

A second route to tree-like polymer organization arises in dense systems of ring polymers. The exact structure and dynamics of ring polymers in the melt state are less well understood

than those of linear chains [9]. If rings are unknotted and nonconcatenated, the fact that they cannot pass through one another gives rise to topological constraints that are absent for equivalent linear polymers [1]. As a result, sufficiently long rings cannot freely interpenetrate their neighbours and adopt compact, crumpled conformations [10].

A useful coarse-grained representation takes a crumpled ring contour (Fig. 1.1(a)) to be tightly double-folded into a branched configuration (Fig. 1.1(b2)). The underlying branched path traced out by the double-folded ring (Fig. 1.1(b1)) can itself be used as a representation of the ring. These branched representations connect the compact and territorial behaviour of rings in melts to that of randomly branched polymers [11]. The randomly branched polymer and the tightly double-folded ring can therefore share the same underlying tree geometry while carrying different configurational degrees of freedom. In this thesis, we count these configurational degrees of freedom explicitly to relate the two ensembles as in Fig. 1.1(c).

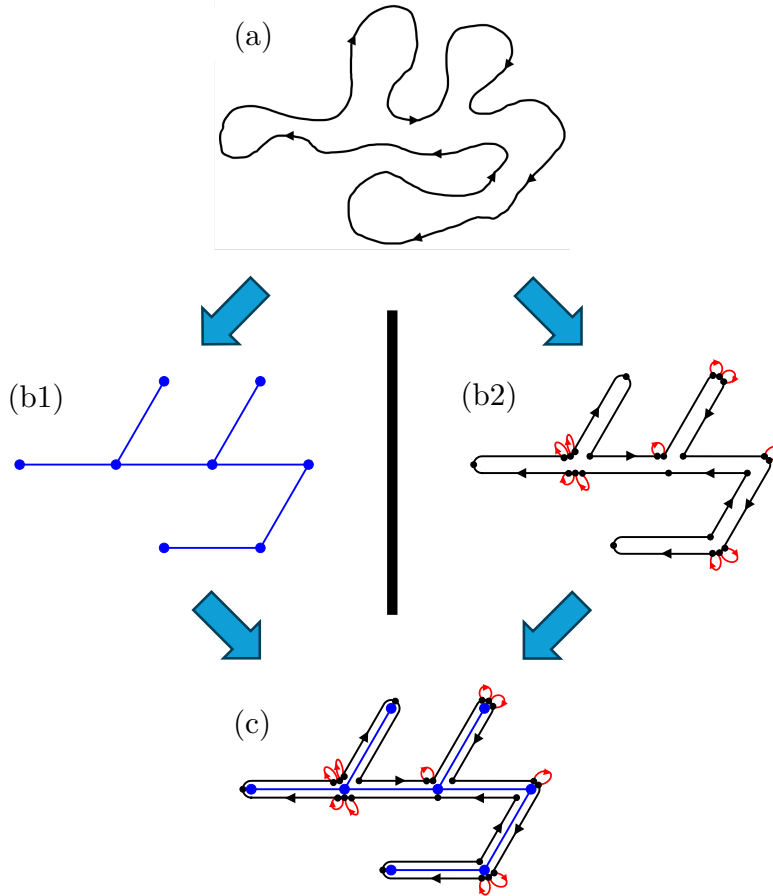


Figure 1.1: The central relation between the ensembles in this thesis. (a) A continuous ring conformation is coarse-grained to its branched geometry, which can either be studied as (b1) a randomly branched polymer or as (b2) a tightly double-folded ring contour with a tree-like geometry. In (c), the two descriptions are superimposed; zero-length ring bonds (red arcs) provide stored length.

1.3 Tree-like organization in biopolymers

Tree-like organization can arise in biological polymers through several different physical mechanisms. Below, we discuss three examples: bacterial DNA, eukaryotic chromatin, and RNA secondary structures.

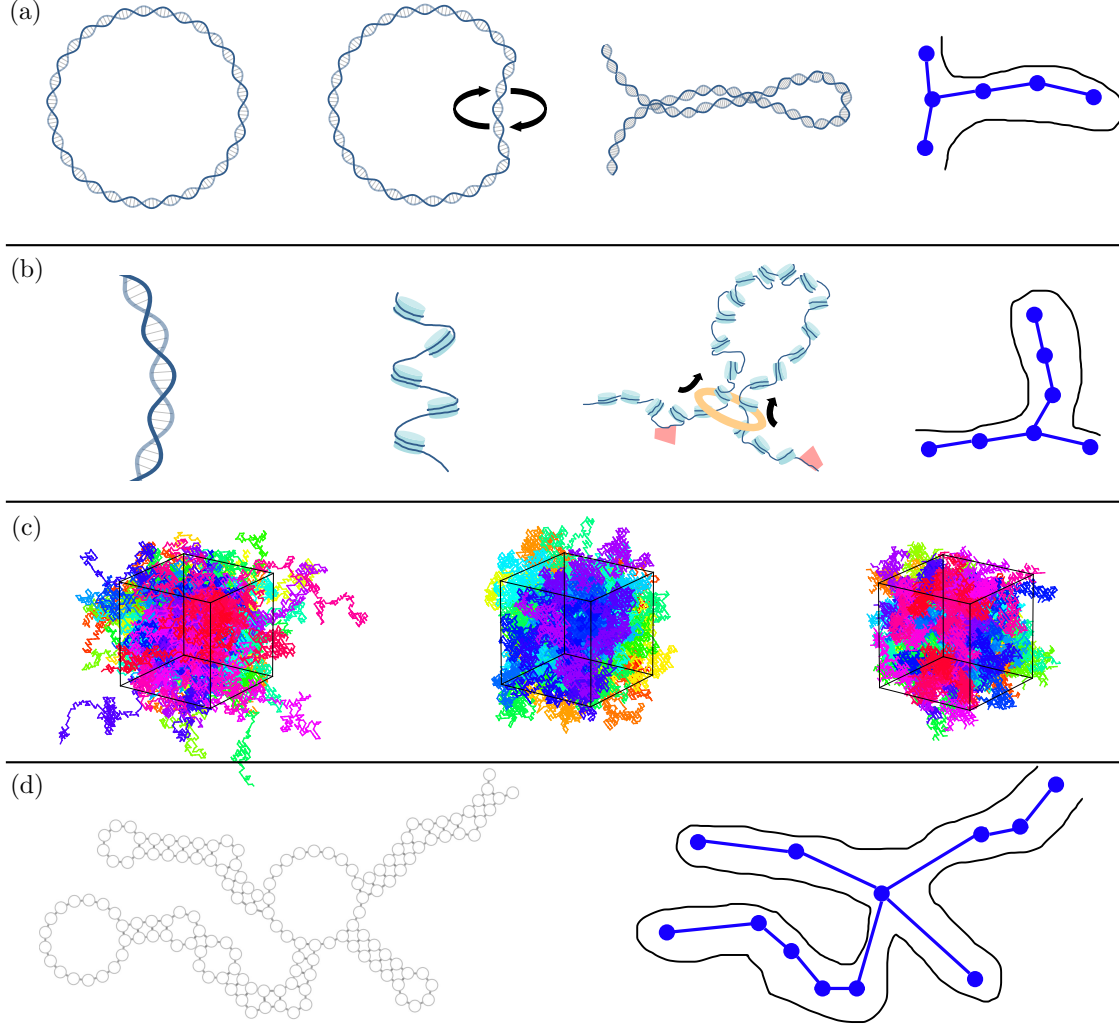


Figure 1.2: Examples motivating tree-like coarse-grained descriptions of polymers and biopolymers. Each panel is read from left to right. (a) Bacterial DNA is circular and, under torsional stress, can form plectonemic structures. These can be coarse-grained as a double-folded ring wrapped around a branched backbone. (b) Eukaryotic DNA undergoes successive levels of compaction: DNA is first wrapped around nucleosomes, while at larger scales SMC complexes (orange ring) extrude chromatin loops, whose growth can be constrained by CTCF sites (pink blocks). Such loops can likewise be coarse-grained as rings around a branched backbone. (c) Polymer melts of linear, ring, and tree polymers of comparable size. The periodic simulation box is shown for scale. Linear polymers strongly interpenetrate and explore a larger volume per polymer, whereas ring and tree polymers remain more spatially segregated and form territorial conformations. (d) A generic pseudoknot-free RNA secondary structure can be mapped onto a branched tree backbone.

1.3.1 Bacterial DNA

Many bacterial chromosomes and plasmids are circular DNA molecules, ranging in length from kilobase pairs for plasmids to several megabase pairs for chromosomes [12]. Because the double helix is closed on itself, additional winding cannot simply be released by rotating an endpoint, and local over- or underwinding therefore generates torsional stress [13].

Transcription, for example, can create overwinding ahead of a moving polymerase and underwinding behind it [14]. Bacterial cells can regulate this stress using topoisomerases [15], which cut DNA and thereby allow its topological state to change. Torsional stress can also be accommodated by supercoiling: the DNA folds back on itself at larger scales, forming *plectonemes* in which two parts of the contour wind around one another [16] (see Fig. 1.2(a)). For sufficiently long circular DNA, several plectonemes can coexist [16, 17], giving the molecule an effectively branched structure. Since these plectonemes can grow, shrink, diffuse, and reorganize along the contour, the corresponding connectivity is naturally annealed rather than quenched.

A coarse-grained computational model of this idea was first introduced in [18, 19]. More recently, a spatially embedded version was developed in [20, 21, 22]. In the latter model, both the circular contour of the DNA and the underlying tree geometry are retained by tightly wrapping a ring polymer around a lattice tree (see Fig. 1.1(c)). The resulting structures are dynamically equilibrated using local Monte Carlo moves that preserve the closed, branched geometry, as reviewed in Sec. 5.3.

The model has been shown to reproduce features of statistically averaged Hi-C contact maps of bacterial DNA [21]. However, simulations based only on the local ring dynamics become prohibitively slow at chromosome-scale lengths; the equilibration times estimated in Sec. 7.6.2 would correspond to several years of computation. This provides a concrete motivation for the more efficient sampling methods developed in this thesis.

1.3.2 Eukaryotic DNA during interphase

Eukaryotic chromosomes are generally much longer than bacterial chromosomes and have a linear rather than circular topology. In humans, approximately 2 m of DNA is packed into a nucleus only of order $10\,\mu\text{m}$ in diameter, making its spatial organization highly non-trivial [23].

DNA is first compacted by wrapping around nucleosomes to form chromatin. Beyond this scale, chromatin organization remains an active area of research [24, 25, 26, 27], with chromosome-contact measurements revealing structure over a broad range of genomic length scales [28, 29].

One mechanism that is particularly relevant here is loop extrusion. Structural maintenance of chromosomes (SMC) complexes [30] can bind to chromatin and progressively extrude loops, while other DNA-binding factors can stall this process at particular locations [31] (see Fig. 1.2(b)). As these loops form, grow, and reorganize, their large-scale geometry can be coarse-grained into an effectively branched structure.

A second connection with the models considered here appears on the scale of entire chromosomes. During interphase, chromosomes occupy distinct territories, whereas sufficiently long linear polymers at equilibrium in a dense melt strongly interpenetrate. The equilibration time of chromosome-sized polymers can, however, be much longer than biologically relevant time scales [32]. Chromosomes may therefore retain compact, weakly mixed conformations rather than reaching the equilibrium state of a linear-polymer melt. This observation has motivated the use of nonconcatenated ring polymers as an equilibrium reference for chromosome organization [32, 33], since sufficiently long rings in melts likewise form compact and territorial conformations [10] (see Fig. 1.2(c)).

These two pictures have different physical origins. Loop extrusion is an active, ATP-driven process, whereas the territorial organization of nonconcatenated rings follows from topological constraints in a dense environment. They should therefore not be identified with one another. Both can nevertheless lead, after sufficient coarse-graining, to an effectively branched polymer geometry.

1.3.3 Secondary structures of RNA molecules

RNA provides a different example of tree-like organization, arising from its secondary structure rather than from the large-scale folding mechanisms discussed above. In structures without pseudoknots, base-paired stems and single-stranded loops or junctions define a natural branched structure (see Fig. 1.2(d)).

In recent work [34], the scaling properties of such RNA secondary structures have been studied as randomly branched polymers. In this representation, a base-paired stem traces the two sides of a tree bond, while single-stranded loops and junctions form the tree nodes. Within this approximation, RNA secondary structures can be related to self-avoiding randomly branched polymers.

1.4 Aim and organization of the thesis

The central aim of this thesis is to derive and exploit configurational entropies for tree-like polymers. We use these entropies to establish quantitative connections between randomly branched polymers and tightly double-folded rings, and to develop efficient methods for sampling the corresponding statistical ensembles. The main results are:

- We develop an exact combinatorial description of ideal labelled trees with controlled branching activity, from which configurational entropies and structural observables can be calculated.
- We introduce a wrapping code for tightly double-folded rings, derive their configurational entropy, and establish a quantitative mapping between ring and tree ensembles. The same correspondence remains applicable to interacting systems when spatial interactions are treated consistently in the two representations.
- We use the tree representation to generate independent ideal-tree configurations in

linear time, providing exact benchmarks and access to system sizes that are difficult to equilibrate using conventional Monte Carlo sampling.

- We analyse and extend Amoeba Monte Carlo algorithms for annealed trees and develop sampling strategies for tightly double-folded rings, with particular attention to their equilibration and computational efficiency.

The remainder of the thesis is organized as follows. Chapter 2 introduces the polymer observables, scaling relations, and ring-melt physics needed in the following chapters. Chapters 3 and 4 form the analytical core of the thesis: the former derives the exact configurational statistics of randomly branched polymers, while the latter develops the corresponding description for tightly double-folded rings and establishes the tree-ring mapping. Chapter 5 develops the sampling algorithms used for these ensembles. Chapters 6 and 7 then test the theory and algorithms, first by establishing their statistical consistency and ensemble equivalence, and subsequently by analysing their equilibration and computational performance. Finally, Chapter 8 summarizes the main results and discusses their limitations and possible extensions.

The material presented in Chapters 3–7 closely follows these publications:

- [P.H.W. van der Hoek](#), A. Rosa, R. Everaers (2024). Amoeba Monte Carlo algorithms for random trees with controlled branching activity: Efficient trial move generation and universal dynamics. *Physical Review E*, 110 (4), 045312.
- [P.H.W. van der Hoek](#), A. Rosa, R. Everaers (2025). Configurational entropy of random trees. *Physical Review E*, 112 (6), 065405.
- [P.H.W. van der Hoek](#), A. Rosa, E. Ghobadpour, R. Everaers (2026). Configurational entropy of randomly double-folding ring polymers. *The Journal of Chemical Physics*, 164 (8), 081101.
- [P.H.W. van der Hoek](#), A. Rosa, E. Ghobadpour, R. Everaers (2026). Coherent modeling of double-folded ring polymers and their underlying random tree structure. *The Journal of Chemical Physics*, 165 (4), 044902.

During the course of this PhD, the candidate also contributed to:

- S.A. Harris, G. Lattanzi, A. Rosa, R. Everaers, et al. (including [P.H.W. van der Hoek](#)) (2026). Meso-soup: A community approach to building a computational description of the biological mesoscale. *Biophysical Journal*, 125(17), 4528–4540.

Chapter 2

Statistical physics of polymers and ring melts

Chapter 1 introduced the configurational degrees of freedom that are central to this thesis. Before studying their statistics in detail, we discuss the polymer-physics results needed to characterize their spatial conformations. We begin with ideal linear chains and rings, introduce the size and contact observables used later, and then discuss how spatial interactions modify these reference statistics. Finally, we return to ring melts to motivate the tree-like, double-folded representation introduced in Chapter 1.

Many polymer observables display power-law scaling and are therefore conveniently characterized by a small number of scaling exponents. Such scaling holds only between appropriate microscopic and macroscopic cut-offs. For the polymers considered in this thesis, we choose the Kuhn length l_K as the smallest length scale.

Equivalently, we describe polymers as freely jointed chains of effective segments of length l_K , so that local orientational correlations below this scale have already been coarse-grained out [35, 36, 37]. In the discrete models used below, this is naturally implemented by choosing a fixed bond length b such that

$$b = l_K. \tag{2.1}$$

We emphasize that, unless stated otherwise, the polymers considered here have fixed bond lengths between adjacent monomers. Elasticity is therefore not introduced through, for example, harmonic bond potentials. When polymers are embedded on a lattice, we also choose the lattice spacing a to be equal to the Kuhn length,

$$a = l_K. \tag{2.2}$$

2.1 Ideal linear chains

The simplest polymer is a linear chain. A linear chain of N monomers consists of $N - 1$ bonds of fixed length $b = l_K$. We embed the polymer on a regular lattice with lattice spacing $a = b$ and coordination number c . The monomers are labelled in increasing order along the contour, such that monomer $i = 1$ is at one end and monomer $i = N$ is at the other. When no additional interactions are present, the chain is an ideal random walk of $N - 1$ steps on the chosen lattice. Up to translations of the starting point, the number of conformations is therefore

$$\Omega_{\text{lin}}(N) = c^{N-1}. \quad (2.3)$$

The corresponding conformational entropy is

$$S_{\text{lin}} = k_B(N - 1) \ln c. \quad (2.4)$$

Equivalently, an ideal linear polymer is specified by the successive orientations of its bonds.

2.1.1 The end-to-end distance

We are usually interested in ensemble-averaged observables rather than individual conformations. One such observable is the squared end-to-end distance,

$$\langle R_{\text{ee}}^2(N) \rangle \equiv \langle (\vec{r}_N - \vec{r}_1)^2 \rangle. \quad (2.5)$$

The end-to-end distance can be computed exactly after specifying the underlying lattice. Consider, for example, a linear chain on the three-dimensional simple cubic lattice, for which $c = 2d = 6$. The $N - 1$ bond orientations can be divided into N_x steps along the x -axis, N_y steps along the y -axis, and N_z steps along the z -axis. For fixed N_x , N_y , and N_z , this can be done in

$$\frac{(N - 1)!}{N_x! N_y! N_z!} \quad (2.6)$$

ways. Each step along a given axis can be in the positive or negative direction. Thus $N_x = N_{x+} + N_{x-}$, and similarly for y and z . For fixed values of N_{x+} , N_{x-} , N_{y+} , N_{y-} , N_{z+} , and N_{z-} , the number of walks is

$$\begin{aligned} & \frac{(N - 1)!}{N_x! N_y! N_z!} \frac{N_x!}{N_{x+}! N_{x-}!} \frac{N_y!}{N_{y+}! N_{y-}!} \frac{N_z!}{N_{z+}! N_{z-}!} \\ &= \frac{(N - 1)!}{N_{x+}! N_{x-}! N_{y+}! N_{y-}! N_{z+}! N_{z-}!}. \end{aligned} \quad (2.7)$$

For a single conformation, the squared end-to-end distance is

$$R_{\text{ee}}^2 = b^2 [(N_{x+} - N_{x-})^2 + (N_{y+} - N_{y-})^2 + (N_{z+} - N_{z-})^2]. \quad (2.8)$$

The ensemble average is therefore

$$\langle R_{\text{ee}}^2(N) \rangle = \frac{(N-1)!}{6^{N-1}} \sum_{\substack{N_{x+}, N_{x-}, \\ N_{y+}, N_{y-}, \\ N_{z+}, N_{z-}}} \frac{R_{\text{ee}}^2}{N_{x+}! N_{x-}! N_{y+}! N_{y-}! N_{z+}! N_{z-}!}, \quad (2.9)$$

with the constraint

$$N_{x+} + N_{x-} + N_{y+} + N_{y-} + N_{z+} + N_{z-} = N - 1. \quad (2.10)$$

In Eq. (2.9) the six orientation counts are distributed multinomially, with probability $1/6$ for each possible orientation at each step. The corresponding means, variances, and covariances are thus

$$\langle N_\alpha \rangle = \frac{N-1}{6}, \quad (2.11)$$

$$\text{Var}(N_\alpha) = \frac{5(N-1)}{36}, \quad (2.12)$$

$$\text{Cov}(N_\alpha, N_\beta) = -\frac{N-1}{36}, \quad \alpha \neq \beta, \quad (2.13)$$

where $\alpha, \beta \in \{x+, x-, y+, y-, z+, z-\}$. Since $\langle N_{x+} - N_{x-} \rangle = 0$, we have

$$\begin{aligned} \langle (N_{x+} - N_{x-})^2 \rangle &= \text{Var}(N_{x+} - N_{x-}) \\ &= \text{Var}(N_{x+}) + \text{Var}(N_{x-}) - 2 \text{Cov}(N_{x+}, N_{x-}) \\ &= \frac{N-1}{3}. \end{aligned} \quad (2.14)$$

The same result holds for the y - and z -directions. Therefore,

$$\langle R_{\text{ee}}^2 \rangle = (N-1)b^2. \quad (2.15)$$

Although derived here for the simple cubic lattice, this result is general for ideal freely jointed chains. The reason is that there are no correlations between distinct bond vectors $\vec{u}_i$:

$$\langle \vec{u}_i \cdot \vec{u}_j \rangle = \begin{cases} 0, & i \neq j, \\ b^2, & i = j. \end{cases} \quad (2.16)$$

Writing the end-to-end vector as a sum over bond vectors gives

$$\begin{aligned}
\langle R_{\text{ee}}^2 \rangle &= \left\langle \left(\sum_{i=1}^{N-1} \vec{u}_i \right)^2 \right\rangle \\
&= \sum_{i=1}^{N-1} \langle \vec{u}_i^2 \rangle + \sum_{\substack{i,j=1 \\ i \neq j}}^{N-1} \langle \vec{u}_i \cdot \vec{u}_j \rangle \\
&= (N-1)b^2.
\end{aligned} \tag{2.17}$$

The independence of this result from the microscopic lattice details is an example of universality. Identifying $N-1$ with the contour length in units of the Kuhn length, the scaling relation is

$$\langle R_{\text{ee}}^2 \rangle \sim (N-1)^{2\nu}, \tag{2.18}$$

$$\nu = \frac{1}{2}. \tag{2.19}$$

The value $\nu = 1/2$ is specific to ideal linear-chain statistics. Interactions and changes of connectivity can generate different exponents; the examples needed later in the thesis are introduced in Secs. 2.4 and 3.4.

2.1.2 Gyration radius

Although the end-to-end distance is useful for linear chains, it has no direct analogue for many more general polymeric systems. For this reason, it is often preferable to characterize polymer size by the squared radius of gyration,

$$R_g^2 \equiv \frac{1}{N} \sum_{i=1}^N (\vec{r}_i - \vec{r}_{\text{com}})^2, \tag{2.20}$$

where

$$\vec{r}_{\text{com}} = \frac{1}{N} \sum_{i=1}^N \vec{r}_i \tag{2.21}$$

is the centre of mass. Equivalently,

$$R_g^2 = \frac{1}{2N^2} \sum_{i,j=1}^N (\vec{r}_i - \vec{r}_j)^2. \tag{2.22}$$

Thus, the radius of gyration measures the average squared spatial distance between monomers.

For an ideal linear chain, the expectation value of R_g^2 follows from the same absence of bond-vector correlations used above. For two monomers separated by $|i-j|$ bonds along

the contour,

$$\langle |\vec{r}_i - \vec{r}_j|^2 \rangle = |i - j| b^2. \quad (2.23)$$

Therefore,

$$\langle R_g^2 \rangle = \frac{b^2}{2N^2} \sum_{i,j=1}^N |i - j| \quad (2.24)$$

$$\begin{aligned} &= \frac{b^2}{N^2} \sum_{i=2}^N \sum_{j=1}^{i-1} (i - j) \\ &= \frac{b^2}{N^2} \sum_{s=1}^{N-1} s(N - s) \\ &= \frac{b^2}{N^2} \frac{N(N^2 - 1)}{6} \\ &= \frac{b^2}{6} \left(N - \frac{1}{N} \right). \end{aligned} \quad (2.25)$$

For large N ,

$$\langle R_g^2 \rangle \simeq \frac{Nb^2}{6}. \quad (2.26)$$

The average radius of gyration of an ideal linear chain therefore obeys the same scaling exponent as the average end-to-end distance.

2.1.3 Kramers' theorem

So far, we have interpreted the radius of gyration either as the average squared distance of monomers from the centre of mass, Eq. (2.20), or as an average over all pairwise squared monomer distances, Eq. (2.22). For ideal tree-like polymers, there is a third useful interpretation. Because distinct bond vectors are uncorrelated, the average squared spatial distance between two monomers is equal to b^2 times the number of bonds on the unique path connecting them (Eq. (2.23)).

This observation leads to Kramers' theorem [38]. Consider a connected polymer graph without cycles. Cutting any bond divides the graph into two components containing n and $N - n$ monomers. That bond lies on the path between exactly $n(N - n)$ unordered pairs of monomers. Since Eq. (2.22) counts ordered pairs and contains a prefactor $1/(2N^2)$, the factors of two cancel, and the squared radius of gyration of an ideal tree can be written as

$$R_g^2 = \frac{b^2}{N^2} \sum_{\text{cuts}} n(N - n), \quad (2.27)$$

where the sum is over all $N - 1$ bonds of the polymer. Averaging over an ensemble of conformations or topologies gives

$$\langle R_g^2 \rangle = \frac{b^2}{N^2} \left\langle \sum_{\text{cuts}} n(N - n) \right\rangle. \quad (2.28)$$

If all bonds are statistically equivalent, this can also be written as

$$\langle R_g^2 \rangle = \frac{b^2(N-1)}{N^2} \sum_{n=1}^{N-1} n(N-n) p_N(n), \quad (2.29)$$

where $p_N(n)$ is the normalized probability that a randomly chosen cut separates off a sub-polymer of n monomers. For an ideal linear chain, $p_N(n) = 1/(N-1)$ uniformly, and Eq. (2.29) immediately reproduces Eq. (2.25). Although we introduced Kramers' theorem using ideal linear chains, the theorem applies more generally to ideal polymer graphs without cycles. This will be useful in the next chapter.

2.2 Ring polymers

The polymeric systems discussed so far were open linear chains. We now add a first level of complexity by imposing that monomer 1 and monomer N of a linear chain are separated in space by one Kuhn length, so that they can be connected by an additional bond. After introducing this final bond, the polymer becomes a ring with N monomers and N bonds.

2.2.1 Conformational multiplicity of a ring polymer

Again, consider the three-dimensional simple cubic lattice. For a rooted and oriented ring, equivalently a closed random walk of N steps starting at a fixed lattice site, the number of conformations is

$$\begin{aligned} \Omega_{\text{ring}}(N) &= \sum_{\substack{N_{x+}, N_{x-}, \\ N_{y+}, N_{y-}, \\ N_{z+}, N_{z-}}} \frac{N! \delta_{N_{x+}, N_{x-}} \delta_{N_{y+}, N_{y-}} \delta_{N_{z+}, N_{z-}}}{N_{x+}! N_{x-}! N_{y+}! N_{y-}! N_{z+}! N_{z-}!} \\ &= \sum_{\substack{N_x, N_y, N_z \geq 0 \\ N_x + N_y + N_z = N \\ N_x, N_y, N_z \text{ even}}} \frac{N!}{[(N_x/2)!]^2 [(N_y/2)!]^2 [(N_z/2)!]^2}. \end{aligned} \quad (2.30)$$

The Kronecker deltas impose closure: the number of positive and negative steps must be equal along each axis. On the simple cubic lattice this also implies that $\Omega_{\text{ring}}(N) = 0$ for odd N .

For large even N , Eq. (2.30) can be approximated by first taking Stirling's approximation for the factorials, then converting the sum into an integral and subsequently computing the resulting integral using Laplace's method around its maximum, where $N_x = N_y = N_z$, such that:

$$\Omega_{\text{ring}}(N) \simeq A_{\text{sc}} 6^N N^{-3/2}, \quad A_{\text{sc}} = 2 \left(\frac{3}{2\pi} \right)^{3/2}, \quad (2.31)$$

for even N .

2.2.2 Ring closure as contact probability

Equation (2.31) has a simple interpretation: A ring is obtained by imposing the additional constraint that the end point of a random walk returns to the neighbourhood of its starting point. For an ideal chain of N bonds in d dimensions, the typical size scales as $R_{\text{ee}} \sim N^\nu$ with $\nu = 1/2$ (see Sec. 2.1.1). The volume explored by the end point therefore scales as $R^d \sim N^{d\nu}$, so the probability of finding the end point within a fixed microscopic volume near the starting point scales as $N^{-d\nu}$. Thus, more generally,

$$\Omega_{\text{ring}}(N) \propto c^N N^{-d\nu}, \quad (2.32)$$

or, equivalently, the contact probability of the two ends of a chain scales as

$$P_{\text{contact}}(N) \propto N^{-d\nu}. \quad (2.33)$$

For an ideal chain, $\nu = 1/2$, which gives $P_{\text{contact}}(N) \propto N^{-d/2}$.

2.2.3 The size of ideal rings

For a freely jointed ideal ring, Eq. (2.17) gives:

$$\langle R_{\text{ee},\text{ring}}^2 \rangle = \sum_{i=1}^N \langle \vec{u}_i^2 \rangle + \sum_{\substack{i,j=1 \\ i \neq j}}^N \langle \vec{u}_i \cdot \vec{u}_j \rangle = 0,$$

and because $\langle \vec{u}_i^2 \rangle = b^2$, we find on average:

$$\langle \vec{u}_i \cdot \vec{u}_j \rangle \simeq \begin{cases} -\frac{b^2}{N}, & \text{for } i \neq j, \\ b^2, & \text{for } i = j, \end{cases} \quad (2.34)$$

in the large- N limit. For an ideal ring, the mean squared spatial distance between two monomers separated by a contour distance s is then

$$\left\langle |\vec{r}_{i+s} - \vec{r}_i|^2 \right\rangle_{\text{ring}} \simeq b^2 s \left(1 - \frac{s}{N} \right), \quad 0 \leq s \leq N. \quad (2.35)$$

This expression can be understood as follows: For $s \ll N$, the ring does not yet feel the closure constraint and the distance grows as in a linear chain. The expression is symmetric under $s \rightarrow N - s$, as required for a ring, and it vanishes at $s = N$, where the contour has returned to its starting monomer. Substituting Eq. (2.35) into the pair-distance definition

of the radius of gyration yields

$$\begin{aligned}
\langle R_g^2 \rangle_{\text{ring}} &= \frac{1}{2N^2} \sum_{i,j=1}^N \langle (\vec{r}_i - \vec{r}_j)^2 \rangle_{\text{ring}} \\
&\simeq \frac{b^2}{2N} \sum_{s=0}^{N-1} s \left(1 - \frac{s}{N}\right) \\
&\simeq \frac{b^2}{12} \left(N - \frac{1}{N}\right).
\end{aligned} \tag{2.36}$$

For large N ,

$$\langle R_g^2 \rangle_{\text{ring}} \simeq \frac{Nb^2}{12}. \tag{2.37}$$

Thus, an ideal ring is smaller than an ideal linear chain of the same contour length by a factor of two in $\langle R_g^2 \rangle$.

2.3 Interacting linear chains

The ideal-chain results above follow from the absence of correlations between distinct bond vectors. Real polymers, however, interact with themselves and with their environment. In this thesis, we consider excluded volume effects, which prevent monomers from occupying the same lattice sites. These interactions change the statistics of polymer conformations and, in many cases, change the scaling exponent ν .

2.3.1 Self-avoiding walks

The effect of excluded volume on a single linear chain can be estimated by a Flory argument [39]. A chain of size R_{ee} pays an entropic stretching cost of order $R_{\text{ee}}^2/(Nb^2)$. This factor can be found by approximating the variance of the distribution in Eq. (2.9) in the large- N limit. At the same time, the repulsive interaction energy is proportional to the square of the monomer density in the occupied volume of the polymer, giving a contribution of order vN^2/R_{ee}^d , where v is an effective excluded-volume parameter. In units of $k_{\text{B}}T$,

$$\beta F(R_{\text{ee}}) \sim \frac{R_{\text{ee}}^2}{Nb^2} + v \frac{N^2}{R_{\text{ee}}^d}. \tag{2.38}$$

Minimizing this free energy with respect to R_{ee} gives

$$R_{\text{ee}} \sim b N^{\nu_{\text{F}}}, \quad \nu_{\text{F}} = \frac{3}{d+2}. \tag{2.39}$$

In three dimensions, this gives $\nu_{\text{F}} = 3/5$, close to the more accurate self-avoiding-walk value $\nu \simeq 0.588$ [40]. Thus, when a linear polymer is mainly interacting with itself, such as in a good solvent, it is swollen relative to an ideal chain.

2.3.2 Melts of linear chains

When many polymers occupy the same volume at high monomer density, the system forms a polymer melt. Excluded-volume interactions then occur both between monomers belonging to the same chain and between monomers belonging to different chains. In general these interactions cannot be treated on the same footing.

For linear polymers in three dimensions, however, these interactions are screened on sufficiently large length scales, and the chain recovers ideal statistics. A useful quantity for illustrating this behaviour is the self-density of a chain, i.e. the density of its own monomers within the volume that it occupies:

$$\rho_{\text{self}} \sim \frac{N}{R_{\text{ee}}^d} \sim b^{-d} N^{1-d\nu}. \quad (2.40)$$

For $\nu > 1/d$, the self-density decreases with increasing N . A sufficiently long chain therefore occupies a volume containing predominantly monomers belonging to other chains rather than to itself. Contrary to the self-avoiding case, the self-interactions that swell the polymers are thus heavily suppressed and the isotropic contributions from other polymers cancel each other's effect on average [35, 36, 37]. In a dense melt, the resulting screening of excluded-volume interactions leads asymptotically to Gaussian statistics,

$$\nu = \frac{1}{2}. \quad (2.41)$$

2.3.3 Coil-to-globule transition

The situation changes when interactions between monomers are sufficiently attractive, as for a polymer in a poor solvent. The chain then collapses from an extended coil into a compact globule.

A compact globule has a finite monomer density in the large- N limit. Using Eq. (2.40), this requires

$$R_{\text{ee}}^d \sim N, \quad \nu = \frac{1}{d}. \quad (2.42)$$

In three dimensions, this gives $\nu = 1/3$. This constant-density scaling corresponds to a compact, territorial conformation and will be useful below when discussing rings in melt conditions.

2.3.4 Contact probabilities for interacting linear chains

The different conformational regimes discussed above are reflected not only in polymer size but also in contact probabilities. For an ideal chain, two monomers separated by a contour distance s span a typical spatial distance of order $s^{1/2}$, and their contact probability scales as $s^{-d/2}$. More generally, replacing this ideal size by s^ν gives the mean-field estimate

$$P_{\text{contact}}(s) \sim s^{-d\nu}. \quad (2.43)$$

Equation (2.33) is recovered for $\nu = 1/2$. For interacting polymers, this relation is generally only an estimate. Different scaling regimes can occur within the same polymer [37], and local correlations can affect contact statistics independently of the global polymer size. It is therefore often useful to introduce a local or path exponent ν_{path} , defined through the size of a polymer sub-segment,

$$R(s) \sim s^{\nu_{\text{path}}}. \quad (2.44)$$

Within the same mean-field argument, the corresponding contact probability scales as

$$P_{\text{contact}}(s) \sim s^{-d\nu_{\text{path}}}. \quad (2.45)$$

This distinction between global and local scaling is also useful for the branched and ring polymers considered below [41, 42, 43].

2.4 Tree-like descriptions of ring melts

A melt of closed rings is subject to topological constraints that have no analogue for linear chains. If rings are prepared unknotted and nonconcatenated and strand crossings are forbidden, they cannot pass through one another or freely interpenetrate. Simulations of sufficiently long rings consequently show compact and territorial conformations [44, 45, 46, 47, 48, 10, 11, 49], with large-scale behaviour consistent with

$$\langle R_g^2(N) \rangle \sim N^{2\nu_{\text{ring}}}, \quad \nu_{\text{ring}} \simeq \frac{1}{3}. \quad (2.46)$$

Topological constraints are not important on the smallest length scales. There, linear chain statistics are expected and the crossover towards the compact regime can be slow. Effective exponents between the linear-melt value $1/2$ and the compact value $1/3$ are therefore commonly observed and predicted [11, 50, 51].

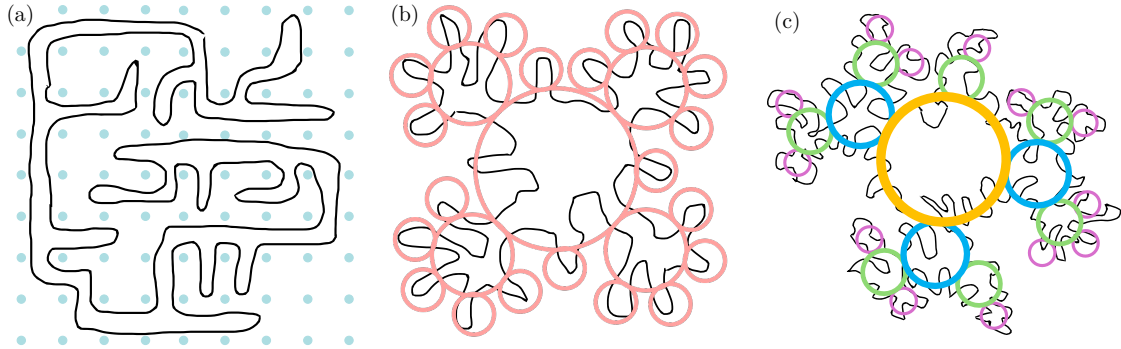


Figure 2.1: Three coarse-grained pictures of ring-polymer melts. (a) A strictly double-folded ring constrained by a fixed array of obstacles, giving rise to a branched primitive path [52, 53]. (b) The decorated-ring model [54], in which a ring fills its territorial volume through a collection of loops. (c) The fractal loopy-globule model [55], in which loops are organized hierarchically over multiple length scales.

2.4.1 Double folding and branched primitive paths

A particularly simple setting is a ring moving through a fixed array of obstacles, as illustrated in Fig. 2.1(a) [52]. To remain unentangled with the obstacles while exploring the available space, the ring repeatedly folds back on itself. The path traced by this double-folded structure defines a *primitive path* [56]. Khokhlov and Nechaev showed that this problem can be mapped onto a random walk on a Cayley tree [52], providing a direct example of a ring whose coarse-grained organization is described by a branched backbone.

For an ideal branched primitive path this construction gives the tree exponent $\nu_{\text{tree}} = 1/4$, rather than the compact value $\nu_{\text{ring}} \simeq 1/3$ observed for long rings in melts. The obstacle model therefore cannot by itself describe the spatial statistics of a real three-dimensional ring melt.

The tree-like picture nevertheless remains useful beyond the fixed-obstacle problem. Branched primitive paths have also been observed for unconcatenated rings in two-dimensional melts [57]. For three-dimensional melts, Ref. [11] described the large-scale organization of rings as a combination of three entropy maximizing strategies: double-folding reduces the surface available for threading, branching increases the number of primitive-path configurations, and spatial interactions prevent the resulting branched structure from becoming overcompact. Within a tightly double-folded description, the last effect is represented by interactions acting on the underlying branched polymer. The appropriate reference is therefore an interacting annealed tree rather than an ideal one.

2.4.2 Scope of the double-folded approximation

A strictly double-folded ring is more constrained than a microscopic ring melt. More general models allow the ring to open locally while retaining an underlying branched organization. Examples are the decorated-ring model [54] and the fractal loopy-globule model [55], shown in Fig. 2.1(b,c).

An important consequence of such openings is the possibility of *threadings*, where part of one ring passes through a loop of another without producing permanent concatenation [58]. Threadings may have modest effects on some static observables but can strongly influence long-time dynamics, and have been discussed in connection with slow or glass-like relaxation [59]. These effects are not included explicitly in the tightly double-folded model studied in this thesis.

At sufficiently coarse resolution, however, branched primitive paths remain a useful description of ring structure on and above the entanglement scale [49]. The tightly double-folded ring should therefore be used as a controlled standard model, rather than as a complete microscopic description of a ring melt.

2.5 Chapter summary

Ideal chains provide the reference relations $\langle R_{ee}^2 \rangle \sim N$, $\langle R_g^2 \rangle \sim N$, and $\nu = 1/2$, while Kramers' theorem rewrites the size of any ideal acyclic polymer in terms of the masses separated by cutting its bonds. Ring closure introduces a return constraint and connects conformational counting to contact probabilities. Spatial interactions and topological constraints can then change the scaling exponent and the associated contact statistics.

For long nonconcatenated rings in melts, topological constraints motivate a coarse-grained description in terms of a branched primitive path. In the tightly double-folded limit, this path can be treated as an annealed tree, but the compact melt statistics require spatial interactions and therefore differ from those of an ideal tree. The following chapter first isolates the configurational statistics of this ideal annealed-tree reference ensemble. Interactions and the explicit wrapped ring contour are reintroduced later.

Chapter 3

Exact configurational statistics of randomly branched polymers

Chapter 2 introduced the polymer observables used throughout this thesis and showed how topological constraints in dense ring systems motivate an effectively double-folded, tree-like backbone on sufficiently coarse length scales [52, 53, 10, 11]. We now isolate that connectivity degree of freedom and study randomly branched polymers as statistical ensembles of annealed trees, beginning with the ideal case in which connectivity can be treated independently of spatial interactions.

The main purpose of this chapter is to establish an exact reference theory for the *ideal* tree ensemble. We first define the tree constraints and the statistical weights assigned to different node functionalities. We then use Prüfer codes to count labelled tree connectivities exactly and obtain the corresponding configurational partition function. The distinction between labelled and unlabelled trees is discussed before turning to structural consequences of the exact ensemble, in particular the dependence of the radius of gyration on branching and the relation between the branch point chemical potential and the activity parameter λ .

3.1 Tree ensemble, constraints, and statistical weights

A randomly branched polymer is represented by a connected acyclic graph $\mathcal{G}$ containing N_{tree} nodes. Such a tree contains

$$N_{\text{bond}}^{\text{tree}} = N_{\text{tree}} - 1 \tag{3.1}$$

tree bonds.

As in Chapter 1, we distinguish a *configuration*, which specifies the connectivity $\mathcal{G}$, from a *conformation*, which additionally specifies the spatial embedding of that connectivity. In the ideal lattice ensemble the $N_{\text{tree}} - 1$ tree bonds may be oriented independently. With

one reference node fixed, a given tree configuration therefore has

$$\Omega_{\text{emb}}(N_{\text{tree}}) = c^{N_{\text{tree}}-1} \quad (3.2)$$

possible embeddings on a regular lattice of coordination number c . This factor will be kept separate from the connectivity multiplicity below. In the simulations used later in the thesis, the embedding lattice is the three-dimensional face-centred cubic (FCC) lattice, for which $c = 12$.

3.1.1 Functionality constraints

We denote by f_i the number of tree bonds emanating from node i . This quantity is referred to as the node *functionality*, or vertex degree. We further denote by N_f the total number of nodes with functionality f . The numbers N_f must satisfy

$$\sum_{f=1}^{f_{\text{max}}} N_f = N_{\text{tree}} , \quad (3.3)$$

where the maximum functionality f_{max} is a model-dependent parameter. When no additional functionality constraint is imposed, $f_{\text{max}} = N_{\text{tree}} - 1$.

Because each tree bond is shared by two nodes, the sum of the node functionalities satisfies

$$\sum_{i=1}^{N_{\text{tree}}} f_i = \sum_{f=1}^{f_{\text{max}}} f N_f = 2(N_{\text{tree}} - 1) . \quad (3.4)$$

Combining Eqs. (3.3) and (3.4) shows that the total number of endpoints, or leaves, N_1 , obeys

$$N_1 = 2 + \sum_{f=3}^{f_{\text{max}}} (f - 2) N_f . \quad (3.5)$$

For most applications, we restrict the maximum functionality to $f_{\text{max}} = 3$. Equations (3.4) and (3.5) then reduce to the linear constraints

$$N_1 = N_3 + 2 , \quad (3.6)$$

$$N_2 = N_{\text{tree}} - 2N_3 - 2 . \quad (3.7)$$

At fixed N_{tree} , the condition $N_2 \geq 0$ implies

$$N_{3,\max} = \frac{N_{\text{tree}} - 2}{2} \quad \text{for } N_{\text{tree}} \text{ even}, \quad (3.8)$$

$$N_{3,\max} = \frac{N_{\text{tree}} - 3}{2} \quad \text{for } N_{\text{tree}} \text{ odd}. \quad (3.9)$$

3.1.2 Controlled activity

Following Refs. [41, 60, 42], we control the numbers N_f of the different node types by assigning a chemical potential μ_f to nodes of functionality f and introducing the Hamiltonian

$$\mathcal{H}_{\mathcal{G}} = - \sum_f \mu_f N_f. \quad (3.10)$$

For maximum functionality $f_{\max} = 3$, all N_f are linearly related through Eqs. (3.6) and (3.7). At fixed N_{tree} , terms in Eq. (3.10) that are independent of N_3 contribute only an overall constant to the statistical weight. After absorbing the remaining linear combination of chemical potentials into the definition of μ_3 , the Hamiltonian can therefore be written as

$$\mathcal{H}_{\mathcal{G}} = -\mu_3 N_3. \quad (3.11)$$

Here and below, μ_3 denotes this effective branch point chemical potential for the tree ensemble. The corresponding chemical potential for the ring ensemble is introduced separately in Chapter 4, where the difference in configurational entropy between the two descriptions becomes relevant.

3.1.3 Ideal and interacting tree ensembles

The exact counting developed in the remainder of this chapter concerns the ideal ensemble, in which the statistical weight of a configuration depends only on its connectivity and on the functionality chemical potentials. Spatial interactions can nevertheless be added at the conformational level without changing the definition of a tree configuration.

For later use, let $\vec{x}_i$ denote the position of tree node i . Hard excluded volume between distinct tree nodes is represented by

$$\beta u_{\text{EV}}(\vec{x}_i, \vec{x}_j) = \begin{cases} +\infty, & \vec{x}_i = \vec{x}_j, \\ 0, & \text{otherwise,} \end{cases} \quad (3.12)$$

with corresponding Boltzmann weight

$$w_{\text{EV}}(\vec{x}_i, \vec{x}_j) = \exp[-\beta u_{\text{EV}}(\vec{x}_i, \vec{x}_j)] = \begin{cases} 0, & \vec{x}_i = \vec{x}_j, \\ 1, & \text{otherwise.} \end{cases} \quad (3.13)$$

The total interaction contribution is

$$\beta\mathcal{H}_{\text{int}} = \sum_{1 \leq i < j \leq N_{\text{tree}}} \beta u_{\text{EV}}(\vec{x}_i, \vec{x}_j). \quad (3.14)$$

Unless explicitly stated otherwise, $\mathcal{H}_{\text{int}} = 0$ in this chapter. This ideal limit is what allows the embedding factor in Eq. (3.2) to factor from the configurational partition function.

3.2 Exact configurational entropy of randomly branched polymers

In the ideal ensemble, the partition function of a randomly branched polymer can be separated into conformational and configurational contributions. At fixed N_{tree} , it can be written as

$$Z_{\text{tree}}(N_{\text{tree}}, \{\mu_f\}) = c^{N_{\text{tree}}-1} \mathcal{Z}_{\text{tree}}(N_{\text{tree}}, \{\mu_f\}), \quad (3.15)$$

where

$$\mathcal{Z}_{\text{tree}}(N_{\text{tree}}, \{\mu_f\}) = \sum'_{\{N_f\}} \Omega_{\text{tree}}(N_{\text{tree}}, \{N_f\}) \exp \left[\beta \sum_f \mu_f N_f \right] \quad (3.16)$$

is the configurational partition function. Here, $\{\mu_f\}$ denotes the set of chemical potentials conjugate to the numbers $\{N_f\}$ of nodes belonging to the different functionality classes. The prime indicates that the sum is restricted to compositions satisfying the tree constraints introduced in Sec. 3.1.1.

The multiplicity

$$\Omega_{\text{tree}}(N_{\text{tree}}, \{N_f\}) \quad (3.17)$$

counts the number of distinct tree configurations with the same composition $\{N_f\}$. The corresponding configurational entropy is given by Boltzmann's relation,

$$S_{\text{tree}}^{\text{conf}}(N_{\text{tree}}, \{N_f\}) = k_{\text{B}} \ln \Omega_{\text{tree}}(N_{\text{tree}}, \{N_f\}). \quad (3.18)$$

For completeness, the conformational, or embedding, entropy follows from the number of ideal lattice embeddings given in Eq. (3.2). With the position of one reference node fixed, it is

$$S_{\text{tree}}^{\text{emb}}(N_{\text{tree}}) = k_{\text{B}} \ln (c^{N_{\text{tree}}-1}) = k_{\text{B}} (N_{\text{tree}} - 1) \ln c. \quad (3.19)$$

The total entropy at fixed tree composition is therefore the sum of the configurational and embedding contributions.

To determine $\Omega_{\text{tree}}(N_{\text{tree}}, \{N_f\})$, we first represent a configuration $\mathcal{G}$ by the connectivity

matrix of its individually labelled nodes. Node labels are regarded as part of the configuration, so trees that differ only by the assignment of labels are counted as distinct. We thus count distinct connectivity matrices rather than different tree topologies. Unlabelled trees are discussed separately in Sec. 3.3.

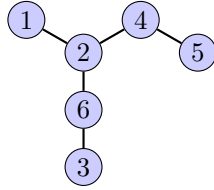
Determining $\Omega_{\text{tree}}(N_{\text{tree}}, \{N_f\})$ is therefore equivalent to counting all connectivity matrices whose corresponding graphs are connected and acyclic and whose node functionalities satisfy the prescribed composition $\{N_f\}$. In graph theory, this is a problem of counting labelled trees.

3.2.1 The total number of labelled trees

Cayley's formula states that the total number of distinct trees on N_{tree} labelled nodes is

$$\sum_{\{N_f\}}' \Omega_{\text{tree}}(N_{\text{tree}}, \{N_f\}) = N_{\text{tree}}^{N_{\text{tree}}-2}. \quad (3.20)$$

Cayley introduced this result in 1889 [61]. In 1918, Prüfer [62] proved it by establishing a one-to-one correspondence between labelled trees and ordered sequences of node labels, now referred to as Prüfer codes. The same correspondence has also been used directly in the statistical analysis of ideal randomly branched polymers [63].



Prüfer code: $[2, 6, 4, 2]$

Figure 3.1: Example of a labelled tree and its corresponding Prüfer code.

To construct the Prüfer code of a labelled tree, one repeatedly selects the current leaf j with the smallest label, records the label i of its unique neighbouring node, and removes j together with its incident bond. The selected leaf j is therefore determined implicitly by the state of the reduced tree, and only its neighbour i is recorded in the code. Detailed examples of the encoding and decoding procedures are given in Tables A.1 and A.2.

For a node i with functionality f_i , its label occurs exactly $f_i - 1$ times in the completed Prüfer code. The code has length

$$N_{\text{tree}} - 2, \quad (3.21)$$

because after $N_{\text{tree}} - 2$ leaves have been removed, only two nodes remain and the bond connecting them is uniquely determined.

The one-to-one correspondence between Prüfer codes and labelled trees makes it possible

to count trees by counting codes. In particular, Eq. (3.20) follows because each of the $N_{\text{tree}} - 2$ entries of an unrestricted Prüfer code can be chosen independently from the labels of the N_{tree} tree nodes [62].

3.2.2 Counting trees with specified functionalities

The one-to-one correspondence between labelled tree connectivities and Prüfer codes also allows trees with prescribed functionalities to be counted [64].

First, consider all labelled nodes of a single tree to have fixed functionalities $\{f_i\}_{i=1}^{N_{\text{tree}}}$. The label i appears $f_i - 1$ times in any possible Prüfer code corresponding to the fixed functionalities. The number of distinct Prüfer codes is thus

$$\Omega_{\text{tree}}(N_{\text{tree}}, \{f_i\}) = \frac{(N_{\text{tree}} - 2)!}{\prod_{i=1}^{N_{\text{tree}}} (f_i - 1)!}. \quad (3.22)$$

When only the numbers $\{N_f\}$ of nodes in each functionality class are prescribed, the labels can be assigned to these classes in

$$\frac{N_{\text{tree}}!}{\prod_f N_f!} \quad (3.23)$$

different ways. Multiplying this factor by Eq. (3.22) gives

$$\Omega_{\text{tree}}(N_{\text{tree}}, \{N_f\}) = N_{\text{tree}}! (N_{\text{tree}} - 2)! \prod_f \frac{1}{N_f! [(f - 1)!]^{N_f}}. \quad (3.24)$$

3.2.3 The partition function for ideal trees

For the models considered in this thesis, we restrict the maximum node functionality to $f_{\text{max}} = 3$. Equation (3.24) then reduces to

$$\Omega_{\text{tree}}(N_{\text{tree}}, N_3) = \frac{N_{\text{tree}}!}{N_1! N_2! N_3!} \frac{(N_{\text{tree}} - 2)!}{2^{N_3}}, \quad (3.25)$$

where Eqs. (3.6) and (3.7) have been used to express $\{N_1, N_2, N_3\}$ in terms of the single independent variable N_3 .

As a useful limiting case, consider a linear chain, for which

$$N_1 = 2, \quad N_2 = N_{\text{tree}} - 2, \quad N_3 = 0. \quad (3.26)$$

Equation (3.25) then gives

$$\Omega_{\text{tree}}(N_{\text{tree}}, N_3 = 0) = \frac{N_{\text{tree}}!}{2}. \quad (3.27)$$

Thus, there are $N_{\text{tree}}!/2$ distinct labelled linear-chain connectivities. Every ordering of

the node labels defines a chain, but an ordering and its reverse generate the same set of bonds.

Using Eq. (3.18), the configurational entropy at fixed N_{tree} and N_3 is

$$S_{\text{tree}}^{\text{conf}}(N_{\text{tree}}, N_3) = k_B \ln \Omega_{\text{tree}}(N_{\text{tree}}, N_3) . \quad (3.28)$$

Introducing the branch point chemical potential μ_3 from Eq. (3.11), the configurational partition function becomes

$$\mathcal{Z}_{\text{tree}}(N_{\text{tree}}, \mu_3) = \sum_{N_3=0}^{N_{3,\text{max}}} \Omega_{\text{tree}}(N_{\text{tree}}, N_3) \exp[\beta \mu_3 N_3] , \quad (3.29)$$

where $N_{3,\text{max}}$ is given by Eqs. (3.8) and (3.9). The full ideal partition function is therefore

$$Z_{\text{tree}}(N_{\text{tree}}, \mu_3) = c^{N_{\text{tree}}-1} \mathcal{Z}_{\text{tree}}(N_{\text{tree}}, \mu_3) . \quad (3.30)$$

The probability of observing N_3 branch nodes at fixed N_{tree} is

$$p(N_3|N_{\text{tree}}, \mu_3) = \frac{\Omega_{\text{tree}}(N_{\text{tree}}, N_3) \exp[\beta \mu_3 N_3]}{\mathcal{Z}_{\text{tree}}(N_{\text{tree}}, \mu_3)} . \quad (3.31)$$

The embedding factor $c^{N_{\text{tree}}-1}$ does not appear in Eq. (3.31) because it is independent of N_3 and cancels upon normalization.

3.3 Labelled and unlabelled tree ensembles

The Prüfer construction counts *labelled* trees: permuting node labels generally produces a different configuration. It is therefore useful to ask how this ensemble differs from the ensemble of *unlabelled* trees, in which a tree state is only defined by its graph topology. The distinction is conceptually important because the two ensembles assign different statistical weights to trees with the same topology.

As an example, consider an ensemble of labelled trees without functionality restrictions. In this ensemble, there are N_{tree} perfect star polymers, with one $f_{\text{max}} = N_{\text{tree}} - 1$ -functional node and $N_1 = N_{\text{tree}} - 1$ leaves by Eq. (3.24). There are, on the other hand, also $N_{\text{tree}}!/2$ (Eq. (3.27)) linear chains in the same ensemble. As a consequence, in an ensemble of labelled trees, linear chains are in general entropically highly favoured over perfect star polymers. In an ensemble of unlabelled trees both topologies have by convention the same weight, as they each represent one state.

For unlabelled trees, the exact number $\Omega_{\text{tree}}^{(\text{unlabelled})}(N_{\text{tree}})$ is numerically known only to finite N_{tree} ; tabulated exact values are available to large finite sizes (the dataset used in the comparisons of Chapter 6 is the one cited in Ref. [65]). Its large- N_{tree} behaviour is well

described by Otter’s asymptotic result [66],

$$\Omega_{\text{tree}}^{(\text{unlabelled})}(N_{\text{tree}}) \simeq 0.5349485 \frac{(2.9557649)^{N_{\text{tree}}}}{N_{\text{tree}}^{5/2}}, \quad (3.32)$$

which is an excellent large- N_{tree} numerical approximation.

At first sight, the difference between Eq. (3.32) and Eq. (3.20) is enormous. For instance, for $N_{\text{tree}} = 20$ the ratio $\Omega_{\text{tree}}^{(\text{labelled})}(N_{\text{tree}})/\Omega_{\text{tree}}^{(\text{unlabelled})}(N_{\text{tree}}) \simeq 3 \cdot 10^{17}$. Much of this difference is due to an overall labelling entropy rather than a difference in the relative statistical weight of tree topologies. A more informative comparison is therefore obtained by normalizing Cayley’s multiplicity by the number $N_{\text{tree}}!/2$ of labelled linear chains. Using Eq. (3.27) and Stirling’s approximation gives

$$\frac{N_{\text{tree}}^{N_{\text{tree}}-2}}{N_{\text{tree}}!/2} \simeq \sqrt{\frac{2}{\pi}} \frac{e^{N_{\text{tree}}}}{N_{\text{tree}}^{5/2}} \simeq 0.7978846 \frac{(2.7182818)^{N_{\text{tree}}}}{N_{\text{tree}}^{5/2}} \quad (3.33)$$

which has the same algebraic prefactor $N_{\text{tree}}^{-5/2}$ as Eq. (3.32), but a slightly different exponential growth constant.

The inverses of Eqs. (3.32) and (3.33) make distinct predictions for the fraction of linear chains in the respective ensembles. We test these predictions explicitly against our simulation results in Sec. 6.1.4.

3.4 Structural consequences of the exact tree ensemble

The exact tree multiplicities derived above can also be used to obtain structural observables. We first consider the radius of gyration at fixed branching and then introduce the activity parameter λ , which provides a compact description of the crossover between weakly and strongly branched ideal trees.

3.4.1 Radius of gyration at fixed branching

The exact multiplicity at fixed (N_{tree}, N_3) can be combined with Kramers’ theorem (Sec. 2.1.3) to obtain the mean-square radius of gyration of an ideal tree. At fixed N_{tree} and N_3 , every allowed labelled tree configuration carries the same weight; the value of μ_3 therefore drops out, which simplifies the calculation.

Following Kramers’ theorem, the gyration radius of randomly branched polymers follows from counting the number of ways a tree can be cut. Cutting one tree bond produces two rooted subtrees, A and B, of sizes n_A and $n_B = N_{\text{tree}} - n_A$. If subtree A contains n_{3A} branch points, then $n_{3B} = N_3 - n_{3A}$. Because the cut bond acts as one external bond at

the root of each subtree, the functionality counts in subtree A obey

$$n_{1A} = n_{3A} + 1, \quad (3.34)$$

$$n_{2A} = n_A - 2n_{3A} - 1, \quad (3.35)$$

and similarly for rooted subtree B . Non-negativity of n_{2A} gives the upper bound

$$n_{3A,\max} = \min\left(\frac{n_A - 2}{2}, N_3\right), \quad n_A \text{ even}, \quad (3.36)$$

$$n_{3A,\max} = \min\left(\frac{n_A - 1}{2}, N_3\right), \quad n_A \text{ odd}. \quad (3.37)$$

The corresponding bound $n_{3B,\max}$ follows by exchanging A and B .

The possible rooted subtrees can be counted with the same Prüfer-code logic used above, with one additional external bond at the root. For a given split (n_A, n_{3A}) and (n_B, n_{3B}) , including the ways to distribute the global node labels between the two subtrees gives

$$\Omega_{\text{cut}}(N_{\text{tree}}, N_3, n_A, n_{3A}) = \frac{N_{\text{tree}}!}{n_A!n_B!} \frac{n_A!}{n_{1A}!n_{2A}!n_{3A}!} \frac{n_B!}{n_{1B}!n_{2B}!n_{3B}!} \frac{(n_A - 1)!}{2^{n_{3A}}} \frac{(n_B - 1)!}{2^{n_{3B}}} \quad (3.38)$$

$$= \frac{N_{\text{tree}}!(n_A - 1)!(n_B - 1)!}{2^{N_3} n_{1A}!n_{2A}!n_{3A}!n_{1B}!n_{2B}!n_{3B}!}. \quad (3.39)$$

At fixed (N_{tree}, N_3) the common factors $N_{\text{tree}}!/2^{N_3}$ cancel upon normalization. Summing over all allowed allocations of branch points gives

$$\Omega_{\text{cut}}(N_{\text{tree}}, N_3, n_A) = \sum_{n_{3A,\min}}^{n_{3A,\max}} \Omega_{\text{cut}}(N_{\text{tree}}, N_3, n_A, n_{3A}), \quad (3.40)$$

where the lower bound follows from the requirement $n_{3B} = N_3 - n_{3A} \leq n_{3B,\max}$,

$$n_{3A,\min} = \max(0, N_3 - n_{3B,\max}). \quad (3.41)$$

The normalized probability for a randomly chosen cut to produce a component of size n_A is then

$$p(n_A | N_{\text{tree}}, N_3) = \frac{\Omega_{\text{cut}}(N_{\text{tree}}, N_3, n_A)}{\mathcal{N}(N_{\text{tree}}, N_3)}, \quad \mathcal{N}(N_{\text{tree}}, N_3) = \sum_{n_A=1}^{N_{\text{tree}}-1} \Omega_{\text{cut}}(N_{\text{tree}}, N_3, n_A). \quad (3.42)$$

Kramers' theorem then gives

$$\langle R_g^2(N_{\text{tree}}, N_3) \rangle = b^2 \frac{N_{\text{tree}} - 1}{N_{\text{tree}}^2} \sum_{n_A=1}^{N_{\text{tree}}-1} p(n_A | N_{\text{tree}}, N_3) n_A (N_{\text{tree}} - n_A). \quad (3.43)$$

Note that in this discussion we have omitted any counting of $c^{N_{\text{tree}}-1}$ bond orientations

(Eq. (3.2)), as the embedding degrees of freedom drop out in all of the above.

This fixed-branching result is equivalent to the expression derived for branched polymers by Zimm and Stockmayer [6]. Their derivation starts from a different statistical construction, but at fixed (N_{tree}, N_3) the same uniform weighting of allowed tree connectivities is recovered. In a long derivation (not shown here), they found

$$\langle R_g^2(N_{\text{tree}}, N_3) \rangle \simeq \left(\frac{\sqrt{\pi}}{4} \frac{1}{\sqrt{N_3}} - \frac{5}{12N_3} \right) N_{\text{tree}} b^2 \quad \text{for } N_{\text{tree}} \gg 1. \quad (3.44)$$

The corresponding crossover is tested against independent simulation data in Sec. 6.2.1 (Fig. 6.4). To obtain an analytic description that also includes the nearly linear regime, we next allow N_3 to fluctuate and introduce a branch activity.

3.4.2 Low-branching activity and the Daoud–Joanny form

The fixed- N_3 expression can be averaged with Eq. (3.31), but the low-branching crossover is more transparent if branch points are allowed to fluctuate already at the rooted-subtree level. We therefore define the rooted-subtree partition function

$$\zeta(n_A, \mu_3) = \sum_{n_{3A}=0}^{n_{3A,\text{max}}} \frac{n_A!}{n_{1A}! n_{2A}! n_{3A}!} \frac{(n_A - 1)!}{2^{n_{3A}}} \exp(\beta \mu_3 n_{3A}), \quad (3.45)$$

with $n_{3A,\text{max}}$ given by Eqs. (3.36) and (3.37). The probability $p(n_A | N_{\text{tree}}, \mu_3)$ can now be written as

$$p(n_A | N_{\text{tree}}, \mu_3) = \frac{\frac{N_{\text{tree}}!}{n_A!(N_{\text{tree}}-n_A)!} \zeta(n_A, \mu_3) \zeta(N_{\text{tree}} - n_A, \mu_3)}{\sum_{n_A=1}^{N_{\text{tree}}-1} \frac{N_{\text{tree}}!}{n_A!(N_{\text{tree}}-n_A)!} \zeta(n_A, \mu_3) \zeta(N_{\text{tree}} - n_A, \mu_3)} \quad (3.46)$$

$$= \frac{\zeta^*(n_A, \mu_3) \zeta^*(N_{\text{tree}} - n_A, \mu_3)}{\sum_{n_A=1}^{N_{\text{tree}}-1} \zeta^*(n_A, \mu_3) \zeta^*(N_{\text{tree}} - n_A, \mu_3)}, \quad (3.47)$$

where factors common to numerator and denominator have been cancelled by defining

$$\zeta^*(n_A, \mu_3) = \sum_{n_{3A}=0}^{n_{3A,\text{max}}} \frac{(n_A - 1)!}{n_{1A}! n_{2A}! n_{3A}!} \left(\frac{\exp(\beta \mu_3)}{2} \right)^{n_{3A}}. \quad (3.48)$$

We now consider the combined low-branching and large-subtree limit, $\beta \mu_3 \ll 0$ with n_A large. In this limit, $n_{2A} \approx n_A$, such that $\frac{(n_A-1)!}{n_{2A}!} \approx n_A^{2n_{3A}}$, giving:

$$\zeta^*(n_A, \mu_3) \approx \sum_{n_{3A}=0}^{n_{3A,\text{max}}} \frac{1}{(n_{3A} + 1)! n_{3A}!} \left(n_A^2 \frac{\exp(\beta \mu_3)}{2} \right)^{n_{3A}} \quad (3.49)$$

$$\approx \sum_{n_{3A}=0}^{\infty} \frac{1}{(n_{3A} + 1)! n_{3A}!} \left(n_A^2 \frac{\exp(\beta \mu_3)}{2} \right)^{n_{3A}} \quad (3.50)$$

$$\approx \frac{I_1(2\lambda n_A)}{\lambda n_A}, \quad (3.51)$$

where $I_1(x)$ is the modified Bessel function of the first kind, and we introduced

$$\lambda = \frac{\exp(\beta\mu_3/2)}{\sqrt{2}} = \exp((\beta\mu_3 - \ln 2)/2), \quad (3.52)$$

as the low-branching activity of the branch nodes. A closely related result to Eq. (3.51) was first derived by Zimm and Stockmayer [6] and later derived in a different manner by De Gennes [67] and extended by Daoud and Joanny [68]. This closed form allows asymptotic observables to be obtained without evaluating the full finite-size combinatorics. In particular, for Bessel functions:

$$\frac{I_1(2\lambda n_A)}{\lambda n_A} \simeq \begin{cases} 1 + \frac{(\lambda n_A)^2}{2}, & \lambda n_A \ll 1 \\ \frac{\exp(2\lambda n_A)}{2\pi^{1/2}(\lambda n_A)^{3/2}}, & \lambda n_A \gg 1 \end{cases}, \quad (3.53)$$

such that the gyration radius follows as

$$\langle R_g^2(N_{\text{tree}}, \lambda) \rangle \simeq \frac{\sum_{n_A=1}^{N_{\text{tree}}-1} \frac{I_1(2\lambda n_A) I_1(2\lambda(N_{\text{tree}}-n_A))}{(n_A(N_{\text{tree}}-n_A))^{1/2}}}{\sum_{n_A=1}^{N_{\text{tree}}-1} \frac{I_1(2\lambda n_A) I_1(2\lambda(N_{\text{tree}}-n_A))}{(n_A(N_{\text{tree}}-n_A))^{3/2}}} \quad (3.54)$$

$$\approx \begin{cases} \frac{N_{\text{tree}}}{6} \left(1 - \frac{\lambda^2 N_{\text{tree}}^2}{30}\right) b^2, & \text{for } \lambda N_{\text{tree}} \ll 1 \\ \frac{\sqrt{\pi}}{4} \sqrt{\frac{N_{\text{tree}}}{\lambda}} b^2, & \text{for } \lambda N_{\text{tree}} \gg 1 \end{cases}. \quad (3.55)$$

Equation (3.54) recovers the linear-chain result, Eq. (2.25), for $\lambda N_{\text{tree}} \ll 1$. In the strongly branched regime one may identify, to leading order,

$$\lambda \simeq \frac{N_3}{N_{\text{tree}}}, \quad (3.56)$$

which reduces Eq. (3.54) to the Zimm–Stockmayer form in Eq. (3.44). At fixed nonzero λ , $\langle R_g^2 \rangle \propto N_{\text{tree}}^{1/2}$, so the spatial size scales as $R_g \propto N_{\text{tree}}^{1/4}$; equivalently, $\nu_{\text{tree}} = 1/4$, consistent with Chapter 2.

3.5 Branching activity over the full branching range

The low-branching expression in Eq. (3.52) grows without bound as μ_3 increases and therefore cannot describe the strongly branched regime, where the functionality constraint imposes $N_3/N_{\text{tree}} \leq 1/2$. A definition that remains meaningful across the crossover is

$$\lambda \equiv \frac{\langle N_3 \rangle}{N_{\text{tree}}} + \mathcal{O}(N_{\text{tree}}^{-1}). \quad (3.57)$$

Starting from the exact distribution in Eq. (3.31), and omitting factors independent of

N_3 , one has for $f_{\max} = 3$

$$p(N_3 \mid N_{\text{tree}}, \mu_3) \propto \frac{1}{N_1! N_2! N_3!} \left(\frac{e^{\beta\mu_3}}{2} \right)^{N_3}. \quad (3.58)$$

For $N_3 \gg 1$ and $N_{\text{tree}} \gg 1$, Stirling's approximation together with $N_1 \simeq N_3$ and $N_2 \simeq N_{\text{tree}} - 2N_3$ gives, to leading exponential order,

$$\ln p \simeq -N_1 \ln N_1 - N_2 \ln N_2 - N_3 \ln N_3 + N_3(\beta\mu_3 - \ln 2) + \text{const.} \quad (3.59)$$

Maximizing this expression with respect to N_3 yields

$$\langle N_3 \rangle \simeq \frac{N_{\text{tree}}}{2 + \exp[(-\beta\mu_3 + \ln 2)/2]} \quad (3.60)$$

$$\equiv \lambda N_{\text{tree}}, \quad (3.61)$$

with

$$\lambda(\mu_3) = \frac{1}{2 + \exp[(-\beta\mu_3 + \ln 2)/2]}. \quad (3.62)$$

Equation (3.62) reduces to Eq. (3.52) for $\beta\mu_3 \ll 0$ and saturates at $\lambda = 1/2$ for $\beta\mu_3 \rightarrow \infty$, as required by the functionality constraint.

Retaining the leading finite-size corrections implied by Eqs. (3.6) and (3.7) gives

$$\langle N_1 \rangle \simeq \lambda N_{\text{tree}} + 1, \quad (3.63)$$

$$\langle N_2 \rangle \simeq (1 - 2\lambda) N_{\text{tree}}, \quad (3.64)$$

$$\langle N_3 \rangle \simeq \lambda N_{\text{tree}} - 1. \quad (3.65)$$

In the opposite low-activity regime, where the mean number of branch points is small, expanding the exact partition function in $e^{\beta\mu_3}$ gives

$$\langle N_3 \rangle \simeq \frac{N_{\text{tree}}^2 e^{\beta\mu_3}}{6} + \mathcal{O}(N_{\text{tree}}^4 e^{2\beta\mu_3}) \quad (3.66)$$

$$\simeq \frac{(\lambda N_{\text{tree}})^2}{3}, \quad \beta\mu_3 \ll 0, \quad (3.67)$$

consistent with the low-activity results of De Gennes and Daoud–Joanny [67, 68].

The resulting crossover for $\langle N_3 \rangle$ and $\langle R_g^2 \rangle$ is compared with exact numerical sums and independently Prüfer-sampled trees in Sec. 6.2.2 (Fig. 6.5).

3.6 Chapter summary

For ideal randomly branched polymers, connectivity and spatial embedding factorize. Prüfer's one-to-one correspondence between labelled trees and label sequences therefore gives an exact configurational multiplicity at fixed functionality composition and, for $f_{\max} = 3$, an exact finite-size distribution of the number of branch points. The comparison

with unlabelled trees gives a falsifiable test that distinguishes simulated samples of labelled and unlabelled trees.

The same combinatorics also determines structural observables. Combining the cut multiplicity with Kramers' theorem gives $\langle R_g^2 \rangle$ at fixed branching, while the activity λ provides a compact description of the crossover from almost linear chains to strongly branched ideal trees. These exact tree weights are the reference against which the ring ensemble is compared in Chapter 4 and the sampling algorithms are tested in Chapter 6.

Chapter 4

Tightly double-folded rings and the tree–ring correspondence

Chapter 3 established the exact configurational statistics of labelled trees. We now return to the continuous ring contour and ask how many distinct tightly double-folded ring configurations correspond to an underlying tree structure. The tree connectivity is no longer the complete configurational specification: the order in which the sequentially labelled ring contour traverses that tree provides an additional degree of freedom.

The chapter proceeds in three steps. First, a *wrapping code* is introduced and used to count tightly double-folded ring configurations exactly. Second, stored length is incorporated through zero-length bonds (reptons), so that N_{tree} may fluctuate at fixed ring length N_{ring} . Third, the tree and ring weights are compared directly, yielding the statistical weight and chemical-potential shift required to switch coherently between the two descriptions. The final sections collect asymptotic consequences for elastic rings and the distribution of stored length.

4.1 Ring representation and notation

The tree notation and functionality constraints are inherited from Chapter 3. In particular, a tree with N_{tree} nodes has $N_{\text{tree}} - 1$ bonds [Eq. (3.1)]. We reserve the term *node* for this tree backbone and *monomer* for the sequential units of the ring contour.

4.1.1 Ring size and stored length

A closed ring of N_{ring} monomers has N_{ring} ring bonds. In the tightly double-folded limit, each tree bond is traversed once in each direction, so the minimal ring associated with a tree of size N_{tree} contains

$$N_{\text{ring}}^{(0)} = 2(N_{\text{tree}} - 1) \quad (4.1)$$

finite-length ring bonds.

To connect with the elastic lattice model [20, 21, 22], ring bonds are allowed to have

length either b or zero. A zero-length bond stores one unit of contour length and is referred to as a repton. At fixed total ring length,

$$N_{\text{ring}} = 2(N_{\text{tree}} - 1) + N_{\text{rep}}, \quad (4.2)$$

where N_{rep} is the number of reptons. Consequently, the underlying tree size fluctuates when stored length is exchanged between zero-length and finite-length ring bonds.

4.1.2 Ring configurations and conformations

Figure 4.1 illustrates why the distinction between configuration and conformation is slightly richer for a ring than for a tree. A tree configuration is fixed by its connectivity $\mathcal{G}$. A tightly double-folded ring configuration additionally specifies how the sequentially labelled ring monomers are placed along and around that tree. Thus a cyclic shift of the ring labels around an unchanged tree is a configurational change of the ring but not of the tree. A conformation additionally specifies the common spatial embedding of the underlying tree bonds.

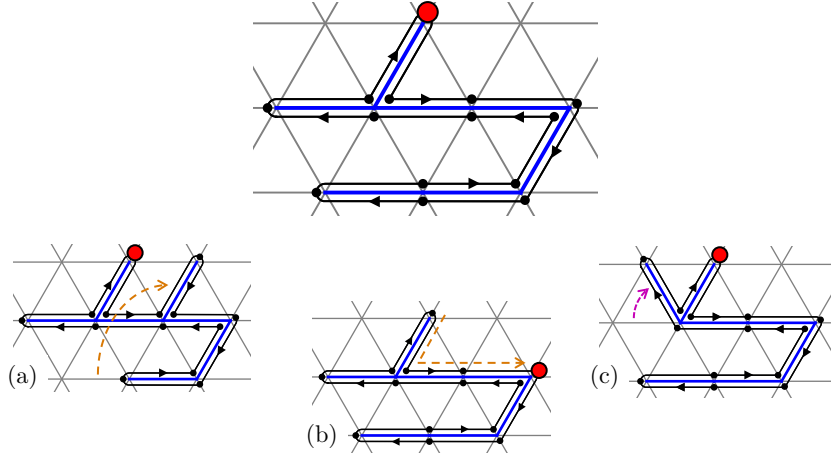


Figure 4.1: Examples of configurational and conformational changes relative to the reference state in the upper panel. The blue backbone is the underlying tree and the black contour is the tightly double-folded ring; the red marker follows one labelled ring monomer. (a) Changing the tree connectivity changes both tree and ring configurations. (b) A cyclic permutation of the ring labels changes the ring configuration while leaving the tree configuration fixed. (c) Reorienting a tree bond without changing connectivity changes the conformation but not the configuration.

4.1.3 Branching activity and spatial interactions

For $f_{\text{max}} = 3$, branching in the ring ensemble is controlled by

$$\mathcal{H}_{\mathcal{G}}^{\text{ring}} = -\tilde{\mu}_3 N_3, \quad (4.3)$$

where the tilde distinguishes the ring chemical potential from the tree chemical potential μ_3 introduced in Eq. (3.11). Their quantitative relation is derived in Sec. 4.3.3.

When excluded volume is included, interactions are assigned between distinct tree node positions as in Sec. 3.1.3. Ring monomers that belong to the same coarse-grained tree node may therefore coincide, as required by tight double folding and by zero-length bonds. This convention ensures that corresponding tree and ring conformations carry the same spatial interaction energy; their statistical weights then differ only through the configurational multiplicities derived below.

4.2 Exact configurational entropy of tightly double-folded rings

A configuration of a tightly double-folded ring is specified by the topology of its underlying tree and by the manner in which the labelled ring is wrapped around it. By convention, all monomers belonging to the same tree node occupy the same position in space, while neighbouring nodes on the tree occupy adjacent lattice sites and their ring monomers are connected by a pair of oppositely oriented finite-length bonds (see Fig. 1.1(b2)). We take the sequentially labelled ring monomers as the relevant distinguishable degrees of freedom. The nodes of the underlying tree are therefore not independently labelled and, in contrast to the tree ensemble considered in Sec. 3.2, permutations of tree node labels do not generate additional ring configurations.

Throughout the following counting argument, we consider $N_{\text{tree}} \geq 2$. The trivial state $N_{\text{tree}} = 1$ may be included separately with unit multiplicity if required.

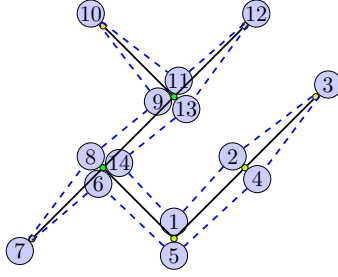
4.2.1 Wrapped trees and a wrapping code

To count the number of double-folded ring states, we introduce first the wrapping procedure originally outlined in [43] and described in detail in Table A.4. Starting from a given tree, the ring monomers are placed sequentially on tree node sites. The procedure starts by placing ring monomer $i = 1$ on a tree node and choosing an initial direction of traversal. The tree connectivity is then followed until every tree bond has been traversed twice and the ring returns to its starting point. During the traversal, every tree node t of functionality f_t is visited exactly f_t times, such that in total $2(N_{\text{tree}} - 1)$ ring monomers are placed (see also Eqs. (4.1) and (3.4)).

To characterize the resulting ring configuration, we introduce a wrapping code. Tree nodes are indexed by their order of first encounter during the wrapping procedure, $t = 1, \dots, N_{\text{tree}}$, and for every newly encountered node its functionality f_t is appended to the code. The wrapping code is therefore the ordered sequence

$$[f_1, f_2, \dots, f_{N_{\text{tree}}}] . \quad (4.4)$$

It contains exactly N_{tree} entries. Figure 4.2 shows an example.



Wrapping code: $[2, 2, 1, 3, 1, 3, 1, 1]$

Figure 4.2: Example of a labelled tightly double-folded ring (blue barred line) and its corresponding wrapping code. Ring labels are placed in the order of the wrapping.

As demonstrated by Table A.5, a wrapping sequence contains the information to reconstruct the unique configuration of a double-folded ring polymer. In particular, any viable permutation of the wrapping sequence corresponds either to a different set of monomer placements on the tree backbone and/or a different tree topology. There is therefore a one-to-one correspondence between double-folded ring configurations and viable wrapping codes.

4.2.2 The total number of viable wrapping codes

Not every length- N_{tree} sequence containing the required numbers of 1's, 2's and 3's corresponds to a legitimate double-folded ring. For instance, the procedure illustrated in Table A.4 implies that a proper wrapping code must always end with $f_{N_{\text{tree}}} = 1$, because the last newly encountered tree node must be a leaf. For given values (N_{tree}, N_3) , this constraint leaves

$$\frac{(N_{\text{tree}} - 1)!}{(N_1 - 1)! N_2! N_3!} \quad (4.5)$$

potential orderings of the first $N_{\text{tree}} - 1$ entries of a wrapping code.

The condition that the ring must only be closed with the last placed bond rules out a code like $[1, 1, 3, 1]$, where the ring would already be closed after two steps (see Table A.6). To see when the reading of a wrapping code leads to an error, it is useful to keep a count n_f of the numbers of f 's read. The ring cannot close prematurely as long as $n_1 < n_3 + 2$, because then the encountered tree nodes do not fulfil the condition for a wrappable sub-tree. Using the global constraint $N_1 = N_3 + 2$, this is equivalent to the condition $N_1 - n_1 > N_3 - n_3$ for the rest of the wrapping code. Reading the code backwards makes the constraint particularly transparent. The first entry is necessarily a '1', and among all subsequent prefixes the number of 3's may never exceed the number of 1's. The probability for this to be the case in a random sequence of $N_1 - 1 = N_3 + 1$ 1's and N_3 3's is given by a generalized form of Bertrand's "ballot theorem" [69] that permits *ties* (see Sec. B.1), namely the fraction of valid sequences is

$$\frac{2}{N_3 + 2}. \quad (4.6)$$

Multiplying the total number of admissible permutations in Eq. (4.5) by the ballot-theorem fraction in Eq. (4.6), we obtain the total number of viable wrapping codes and thus of tightly double-folded ring configurations for given (N_{tree}, N_3) ,

$$\Omega_{\text{ring}}(N_{\text{tree}}, N_3) = \frac{2(N_{\text{tree}} - 1)!}{N_1! N_2! N_3!}, \quad (4.7)$$

with N_1 and N_2 given by Eqs. (3.6) and (3.7).

4.2.3 The partition function for ideal tightly double-folded rings

At fixed (N_{tree}, N_3) , Eq. (4.7) is sufficient to compute the configurational entropy of ideal tightly double-folded rings using Boltzmann's relation, as in Eq. (3.18):

$$S_{\text{ring}}^{\text{conf}}(N_{\text{tree}}, N_3) = k_B \ln \Omega_{\text{ring}}(N_{\text{tree}}, N_3). \quad (4.8)$$

Using the Hamiltonian in Eq. (4.3), the branching activity of tightly double-folded rings can be controlled via a chemical potential $\tilde{\mu}_3$. The configurational partition function for tightly double-folded rings in the ideal ensemble is therefore

$$\mathcal{Z}_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3) = \sum_{N_3=0}^{N_{3,\text{max}}} \Omega_{\text{ring}}(N_{\text{tree}}, N_3) \exp(\beta \tilde{\mu}_3 N_3), \quad (4.9)$$

where $N_{3,\text{max}}$ is defined in Eqs. (3.8) and (3.9). Including $c^{N_{\text{tree}}-1}$ different tree bond orientations, we find the conformational partition function for ideal tightly double-folded rings

$$Z_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3) = c^{N_{\text{tree}}-1} \mathcal{Z}_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3). \quad (4.10)$$

4.2.4 Accounting for the reptons in the elastic lattice model

In the elastic lattice model of tightly double-folded rings studied here [20, 22], stored length is represented by ring bonds whose length is zero rather than equal to the lattice spacing (see red arcs in Fig. 1.1(b2) and (c)). However, up to this point, the theory was formulated at fixed tree size N_{tree} . To compare with the elastic lattice model, we now switch to the ensemble at fixed ring length N_{ring} in which the ring-bond length, and so the size N_{tree} of the underlying tree, fluctuates. The joint probability of observing (N_{tree}, N_3) at fixed $(N_{\text{ring}}, \tilde{\mu}_3)$ is therefore

$$p(N_{\text{tree}}, N_3 \mid N_{\text{ring}}, \tilde{\mu}_3) = \frac{\Omega_{\text{rep}}(N_{\text{ring}}, N_{\text{tree}}) \Omega_{\text{ring}}(N_{\text{tree}}, N_3) c^{N_{\text{tree}}-1} e^{\beta \tilde{\mu}_3 N_3}}{Z_{\text{elastic}}(N_{\text{ring}}, \tilde{\mu}_3)}. \quad (4.11)$$

where $\Omega_{\text{rep}}(N_{\text{ring}}, N_{\text{tree}})$ counts the possible placements of the zero-length bonds and

$$Z_{\text{elastic}}(N_{\text{ring}}, \tilde{\mu}_3) = \sum_{N_{\text{tree}}} \Omega_{\text{rep}}(N_{\text{ring}}, N_{\text{tree}}) Z_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3) \quad (4.12)$$

is the partition function of the elastic system. Assuming that all choices of the N_{rep} zero-length bonds among the N_{ring} (see Eq. (4.2)) labelled ring bonds carry equal statistical weight, the number of possible repton arrangements is the binomial multiplicity

$$\Omega_{\text{rep}}(N_{\text{ring}}, N_{\text{tree}}) = \binom{N_{\text{ring}}}{N_{\text{rep}}} = \frac{N_{\text{ring}}!}{[2(N_{\text{tree}} - 1)]! [N_{\text{ring}} - 2(N_{\text{tree}} - 1)]!}. \quad (4.13)$$

The corresponding mean tree size and mean number of branch points are obtained directly from this distribution:

$$\langle N_{\text{tree}} \rangle = \sum_{N_{\text{tree}}} \sum_{N_3} N_{\text{tree}} p(N_{\text{tree}}, N_3 \mid N_{\text{ring}}, \tilde{\mu}_3), \quad (4.14)$$

$$\langle N_3 \rangle = \sum_{N_{\text{tree}}} \sum_{N_3} N_3 p(N_{\text{tree}}, N_3 \mid N_{\text{ring}}, \tilde{\mu}_3). \quad (4.15)$$

The mean values of N_1 and N_2 then follow from Eqs. (3.6) and (3.7).

4.3 Exact correspondence between tree and ring ensembles

Chapter 3 and the preceding sections count the tree and ring configurations using two different one-dimensional codes: Prüfer codes for labelled trees and wrapping codes for labelled rings. We now compare the two statistical weights directly. The purpose is not merely to show that both descriptions use the same underlying topology, but to determine the multiplicity that must be included when one representation is used to sample the other. We first treat $f_{\text{max}} = 3$ and then generalize the result to arbitrary functionality.

4.3.1 The multiplicity of labelled trees revisited

The multiplicity $\Omega_{\text{tree}}(N_{\text{tree}}, N_3)$ derived in Eq. (3.25) relied on the fact that we are studying labelled trees. However, here and in the following, we will divide out the $N_{\text{tree}}!$ factor that accounts for permutation of the tree node labels. The resulting expression, now also explicitly including the trivial $N_{\text{tree}} = 1$ -state,

$$\Omega_{\text{tree}}^*(N_{\text{tree}}, N_3) = \begin{cases} 1, & \text{if } N_{\text{tree}} = 1 \\ \frac{(N_{\text{tree}}-2)!}{N_1! N_2! N_3!} \frac{1}{2^{N_3}}, & \text{if } N_{\text{tree}} \geq 2 \end{cases}, \quad (4.16)$$

is no longer guaranteed to be an integer number. Hence, $\Omega_{\text{tree}}^*(N_{\text{tree}}, N_3)$ is no longer a multiplicity that directly counts states. We can nevertheless use $\Omega_{\text{tree}}^*(N_{\text{tree}}, N_3)$ for the following two reasons:

1. At fixed N_{tree} , the factor $N_{\text{tree}}!$ is common to all tree configurations and therefore drops out of normalized observables. This can be seen explicitly, for example, in the sampling probability in Eq. (3.31).
2. When instead considering an ensemble with a fixed total number N_{nodes} of labelled nodes, of which a fluctuating number N_{tree} belongs to a particular randomly branched polymer, there are

$$\Omega_{\text{labelling}} = \frac{N_{\text{nodes}}!}{N_{\text{tree}}! (N_{\text{nodes}} - N_{\text{tree}})!} \quad (4.17)$$

ways to choose the labels belonging to that tree. This factor multiplies $\Omega_{\text{tree}}(N_{\text{tree}}, N_3)$ in the corresponding partition function and cancels the factor $N_{\text{tree}}!$ contained in the labelled-tree multiplicity. The remaining factor $(N_{\text{nodes}} - N_{\text{tree}})!$ can subsequently be accounted for in the statistical weight associated with the remaining labelled nodes.

The corresponding reduced tree partition function is therefore

$$Z_{\text{tree}}^*(N_{\text{tree}}, \mu_3) = c^{N_{\text{tree}}-1} \sum_{N_3=0}^{N_{3,\text{max}}} \Omega_{\text{tree}}^*(N_{\text{tree}}, N_3) \exp[\beta \mu_3 N_3]. \quad (4.18)$$

4.3.2 Wrapping labelled trees

In Sec. 4.2.1 we introduced the wrapping procedure for an arbitrary tree topology. Now consider a particular tree whose nodes are distinguished by labels. A cyclic shift of the sequentially labelled ring around such a tree then leads to a different placement of ring monomers on the tree nodes. Including the directions of wrapping at each of the N_3 branch points, there are then

$$\Omega_{\text{wrap}}(N_{\text{tree}}, N_3) = \begin{cases} 1, & \text{if } N_{\text{tree}} = 1 \\ 2(N_{\text{tree}} - 1) \times 2^{N_3}, & \text{if } N_{\text{tree}} \geq 2 \end{cases} \quad (4.19)$$

distinct ways to wrap a labelled tree. Inspection of Eqs. (4.7) and (4.16) shows that

$$\Omega_{\text{ring}}(N_{\text{tree}}, N_3) = \Omega_{\text{tree}}^*(N_{\text{tree}}, N_3) \Omega_{\text{wrap}}(N_{\text{tree}}, N_3). \quad (4.20)$$

Reassuringly, counting the wrappings of trees represented by Prüfer codes and counting wrapping codes directly lead to equivalent results.

4.3.3 Quantitative link between tree and ring ensembles

We are now in a position to establish an exact correspondence between the tree and ring descriptions in the ideal ensemble.

One coherent choice is to establish a correspondence between *one* tree conformation and *an ensemble* of $\Omega_{\text{rep}}(N_{\text{ring}}, N_{\text{tree}}) \Omega_{\text{wrap}}(N_{\text{tree}}, N_3)$ ring conformations. Then, according to Eqs. (4.12), (4.18), and (4.20), one must choose the two branching chemical potentials

μ_3 and $\tilde{\mu}_3$ equal to each other.

For the sampling methods developed later, we instead adopt a one-tree–one-ring convention: one tree conformation represents one ring conformation, while the multiplicity of the ring states represented by that tree is included explicitly through the additional statistical weight

$$\begin{aligned} w_{\text{ring}}(N_{\text{tree}}, N_{\text{ring}}, N_3) \\ = \Omega_{\text{rep}}(N_{\text{ring}}, N_{\text{tree}}) \Omega_{\text{wrap}}(N_{\text{tree}}, N_3) \end{aligned} \quad (4.21)$$

in the tree-level statistical weight: tree conformations corresponding to a larger multiplicity of ring wrappings and repton distributions are generated with a correspondingly increased probability. Requiring the tree-size weight $w_{\text{tree}}(N_{\text{tree}}, N_{\text{ring}})$ and the branch point weight to reproduce the corresponding ring multiplicity gives

$$\begin{aligned} w_{\text{tree}}(N_{\text{tree}}, N_{\text{ring}}) \exp[\beta\mu_3 N_3] &= w_{\text{ring}}(N_{\text{tree}}, N_{\text{ring}}, N_3) \exp[\beta\tilde{\mu}_3 N_3] \\ &= \Omega_{\text{rep}}(N_{\text{ring}}, N_{\text{tree}}) \Omega_{\text{wrap}}(N_{\text{tree}}, N_3) \exp[\beta\tilde{\mu}_3 N_3] \\ &= \frac{N_{\text{ring}}! (2(N_{\text{tree}} - 1))}{(2(N_{\text{tree}} - 1))! (N_{\text{ring}} - 2N_{\text{tree}} + 2)!} 2^{N_3} \exp[\beta\tilde{\mu}_3 N_3] \\ &= \frac{N_{\text{ring}}!}{(2N_{\text{tree}} - 3)! (N_{\text{ring}} - 2N_{\text{tree}} + 2)!} \exp[(\beta\tilde{\mu}_3 + \ln(2))N_3], \end{aligned} \quad (4.22)$$

which implies

$$\begin{aligned} w_{\text{tree}}(N_{\text{tree}}, N_{\text{ring}}) \\ = \begin{cases} 1, & \text{if } N_{\text{tree}} = 1 \\ \frac{N_{\text{ring}}!}{(2N_{\text{tree}} - 3)! (N_{\text{ring}} - 2N_{\text{tree}} + 2)!}, & \text{if } N_{\text{tree}} \geq 2 \end{cases} \end{aligned} \quad (4.23)$$

and

$$\beta\mu_3 = \beta\tilde{\mu}_3 + \ln(2). \quad (4.24)$$

Equation (4.24) prescribes a simple shift between the chemical potentials to be used with the two representations as a convenient way to account for the “entropy of wrapping” (Sec. 4.3.2) of double-folded rings with respect to trees.

The statistical weight (4.23) controls the average tree size $\langle N_{\text{tree}} \rangle$ through the competition between the tree and repton entropies. The repton multiplicity is maximal when the numbers of zero-length and finite-length ring bonds are approximately equal,

$$N_{\text{rep}} \simeq 2(N_{\text{tree}} - 1) \simeq \frac{N_{\text{ring}}}{2}. \quad (4.25)$$

At fixed N_{ring} , it decreases on either side of this maximum: for larger N_{tree} because the number of reptons decreases, and for smaller N_{tree} because the number of finite-length

bonds decreases.

With Eqs. (4.23) and (4.24), tree-level sampling can therefore reproduce the configurational weights of the elastic ring ensemble. To transform a tree into a double-folded ring, the starting point of wrapping, repton locations and the directions of wrapping at branch points are chosen randomly.

4.3.4 Including spatial interactions

The multiplicity relation above is combinatorial, but the representation switch is also compatible with the spatial interactions used in this thesis. Corresponding tree and tightly double-folded ring conformations share the same set of tree node coordinates. Because excluded-volume interactions are defined between distinct tree nodes [Eq. (3.14)], the interaction Boltzmann factor is identical in the two representations. The factors Ω_{rep} and Ω_{wrap} therefore remain the only additional configurational weights required when switching between the tree and ring descriptions. This is the convention tested for ideal, dilute self-avoiding, and melt conditions in Chapter 6.

4.3.5 Arbitrary node functionalities

We finally consider a ring wrapping of a labelled tree with arbitrary maximum functionality f_{max} . In general, to fulfil the procedure in Table A.4, a ring wrapping around a tree passes every tree node t exactly f_t times. At a node of functionality f_t , the $f_t - 1$ initially unvisited branches can be traversed in $(f_t - 1)!$ different orders. Including also the $2(N_{\text{tree}} - 1)$ possible choices of the origin and direction of the sequential ring labels, the wrapping multiplicity (Eq. (4.19)) becomes

$$\Omega_{\text{wrap}}(N_{\text{tree}}, \{N_f\}) = 2(N_{\text{tree}} - 1) \prod_f [(f - 1)!]^{N_f}. \quad (4.26)$$

According to Eq. (4.20), multiplication by the reduced tree weight (Eq. (3.24) divided by $N_{\text{tree}}!$) then gives

$$\Omega_{\text{ring}}(N_{\text{tree}}, \{N_f\}) = 2 \frac{(N_{\text{tree}} - 1)!}{\prod_{f=1}^{f_{\text{max}}} N_f!}. \quad (4.27)$$

Similarly, a generalized shift in chemical potentials between the tree and ring ensembles follows from Eq. (4.24):

$$\beta\mu_f = \beta\tilde{\mu}_f + \ln[(f - 1)!]. \quad (4.28)$$

The arbitrary-functionality extension is tested against sampled elastic-ring ensembles in Sec. 6.3.2 (Fig. 6.8).

4.4 Measures for elastic rings

In Appendix C.1, the partition function in Eq. (4.10) for ideal tightly double-folded rings without reptons is approximated in its asymptotic limit. The functional form of the partition function is

$$Z_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3) \sim \frac{(g(\beta\tilde{\mu}_3, c))^{N_{\text{tree}}}}{N_{\text{tree}}^{3/2}}, \quad (4.29)$$

where $g(\beta\tilde{\mu}_3, c)$ is an exponential growth constant that depends on the chemical potential and the lattice embedding. Comparing this result with the functional form of Eq. (3.33) now has a straightforward interpretation: Apart from the additional multiplicity associated with wrapping branch points, trees and double-folded ring ensembles differ by a factor $\sim N_{\text{tree}}$ associated with the circular permutations of the ring monomers.

In the following, we use the approximate result of Sec. C.1 and the random placement of reptons along the ring suggested in Sec. 4.2.4 to derive results applicable for the analysis of the configurations of elastic tightly double-folded rings.

4.4.1 λ for rings

The branching activity λ in Sec. 3.5 for trees can be defined analogously for tightly double-folded rings. Because the elastic ring ensemble is subject to a fluctuating tree size, however, we do so using a mean-field approximation: We first derive the average size $\langle N_{\text{tree}} \rangle$ for fixed N_{ring} , after which λ follows as the ratio of the expected number of branch points to the expected tree size.

As derived in Sec. C.1, the exponential growth factor in Eq. (4.29) is then given by the term

$$g(\beta\tilde{\mu}_3, c) \sim \left(\frac{c}{1 - 2\lambda} \right), \quad (4.30)$$

with

$$\lambda(\tilde{\mu}_3) = (2 + \exp(-\beta\tilde{\mu}_3/2))^{-1}. \quad (4.31)$$

As shown in Sec. C.2, this leading-order contribution to the partition function can be combined with Stirling's approximation for Ω_{rep} in Eq. (4.13) to find the expected tree size as

$$\frac{\langle N_{\text{tree}} \rangle - 1}{N_{\text{ring}}/2} \approx \left(\sqrt{\frac{1 - 2\lambda}{c}} + 1 \right)^{-1} \quad (4.32)$$

and analogously to Eqs. (3.63), (3.64) and (3.65) we thus find asymptotically

$$\langle N_1 \rangle \simeq \lambda \langle N_{\text{tree}} \rangle + 1 \quad (4.33)$$

$$\langle N_2 \rangle \simeq (1 - 2\lambda) \langle N_{\text{tree}} \rangle \quad (4.34)$$

$$\langle N_3 \rangle \simeq \lambda \langle N_{\text{tree}} \rangle - 1, \quad (4.35)$$

Equations (4.32)–(4.35) therefore give explicit large-system predictions for the dominant tree size and functionality fractions. These predictions are tested in Sec. 6.3.2.

4.4.2 Repton distribution according to tree nodes' functionality

We next derive the distribution of stored length over tree nodes. The construction starts from the assumption, used in the elastic lattice ensemble above, that the N_{rep} zero-length bonds are placed uniformly among the labelled ring bonds.

We start from a tree of size N_{tree} around which a minimal ring of size $2(N_{\text{tree}} - 1)$ monomers is wrapped. By construction of the wrapping procedure, each tree node i with functionality f_i has exactly f_i ring monomers on it before placing any reptons. Reptons are then placed uniformly at random on the ring. This implies that the probability to place a repton on a particular tree node i is proportional to the number of ring monomers already located on that node. As a consequence, the probability of choosing a given tree node changes when sequentially placing reptons. This ‘pick and add one’ process is described by the Pólya urn model [70]. For the two-colour urn formed by node i and the remainder of the ring, the initial monomer distributions are $\alpha = f_i$ and $\beta = 2(N_{\text{tree}} - 1) - f_i$. After all reptons have been placed, the number k of reptons associated with a single node therefore follows the beta-binomial distribution

$$P(k | f_i, N_{\text{tree}}, N_{\text{rep}}) = \binom{N_{\text{rep}}}{k} \frac{B[k + f_i, N_{\text{rep}} - k + 2(N_{\text{tree}} - 1) - f_i]}{B[f_i, 2(N_{\text{tree}} - 1) - f_i]}. \quad (4.36)$$

where $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha + \beta)$ is the beta function and $\Gamma(x)$ is the standard gamma function. Averaging this node-level distribution over functionality classes gives

$$P(k | N_{\text{tree}}, N_{\text{rep}}, N_3) = \sum_{f=1}^3 \frac{N_f}{N_{\text{tree}}} P(k | f, N_{\text{tree}}, N_{\text{rep}}). \quad (4.37)$$

For comparison, the distribution of reptons proposed in the supplementary material of [22] was not Eq. (4.36), but the following Poisson distribution:

$$P_{\text{Pois}}(k | f_i, N_{\text{tree}}, N_{\text{rep}}) = \frac{\bar{k}_{f_i}^k e^{-\bar{k}_{f_i}}}{k!}, \quad (4.38)$$

with mean $\bar{k}_{f_i} = N_{\text{rep}} f_i / [2(N_{\text{tree}} - 1)]$. The latter assumes a constant probability for picking tree nodes in the repton distribution process. The constant probability is proportional to the functionality of the tree nodes. Both predictions are tested against local dynamics simulations in Sec. 6.3.3.

4.5 Chapter summary

A tightly double-folded ring contains more configurational information than its underlying tree because the sequentially labelled contour can wrap a given backbone in multiple ways.

The wrapping code accounts for these additional degrees of freedom and yields an exact ring multiplicity. When stored length is included, the ring ensemble acquires a second combinatorial factor, Ω_{rep} , associated with the placement of zero-length bonds.

Comparing these factors with the labelled-tree multiplicity gives an exact relation between the two ensembles. In the representation where one tree corresponds to one ring, the wrapping entropy appears as the shift $\beta\mu_3 = \beta\tilde{\mu}_3 + \ln(2)$ together with a size-dependent entropic factor corresponding to the cyclic permutations of ring monomers and the number of accessible placements of stored length. The same construction yields a partition function for tightly double-folded rings with arbitrary node functionalities. Finally, the elastic-ring partition function yields asymptotic predictions for the expected tree size and functionality fractions. The beta-binomial distribution of stored length over tree nodes follows from the random placement of stored length along the ring. We test these findings against independent simulations in Chapter 6.

Chapter 5

Sampling algorithms for branched polymers and double-folded rings

The previous two chapters derived the equilibrium statistical weights of randomly branched polymers and tightly double-folded rings. This chapter translates those weights into practical sampling procedures. We distinguish two different situations. In the ideal tree ensemble, the exact combinatorial solution permits direct generation of statistically independent configurations. Once spatial interactions are included, or when a local dynamical representation is required, the ensemble must instead be explored by Markov-chain Monte Carlo algorithms. We therefore first construct direct Prüfer samplers and then introduce several Monte Carlo schemes that equilibrate polymers on the tree and ring levels.

The purpose of the present chapter is methodological: for each algorithm we specify what it aims to sample, its trial moves, its proposal probabilities and the relevant statistical weights. Whether the resulting implementations reproduce the intended equilibrium ensembles is tested in Chapter 6. Their equilibration times, dynamical bottlenecks and computational performance are treated separately in Chapter 7.

5.1 Direct Prüfer sampling of ideal trees

The exact labelled-tree multiplicity derived in Sec. 3.2 can be turned into a direct generator of ideal randomly branched polymers. A labelled tree with N_{tree} nodes is in one-to-one correspondence with a Prüfer code of length $N_{\text{tree}} - 2$. Once the required distribution of functionalities has been sampled, the code, connectivity graph and ideal lattice embedding can all be generated in $\mathcal{O}(N_{\text{tree}})$ operations per independent tree. A closely related use of Prüfer sequences to generate ideal randomly branched polymer ensembles and calculate size measures was given by Singaram, Gopal and Ben-Shaul [63].

For a prescribed functionality sequence $\{f_i\}$, the procedure consists of three steps:

1. sample a Prüfer code of length $N_{\text{tree}} - 2$ with label i appearing exactly $f_i - 1$ times;
2. decode the code into the unique labelled connectivity graph, as illustrated in Table A.2;

3. choose a position for one reference node and orient the $N_{\text{tree}} - 1$ tree bonds independently on the lattice.

The third step samples the ideal embedding factor $c^{N_{\text{tree}}-1}$ from Chapter 3. The only ensemble-dependent part is therefore the generation of the functionality sequence.

5.1.1 Unconstrained labelled trees

When no functionality constraints or functionality-dependent weights are imposed, every sequence of $N_{\text{tree}} - 2$ labels drawn from $\{1, \dots, N_{\text{tree}}\}$ is a valid Prüfer code. Sampling every entry independently and uniformly therefore generates every labelled tree with equal probability, reproducing Cayley’s multiplicity in Eq. (3.20).

The code can subsequently be decoded in linear time (see Table A.2). If the code itself is generated sequentially, the complete ideal-tree sample—connectivity plus embedding—therefore costs $\mathcal{O}(N_{\text{tree}})$ operations.

5.1.2 Functionality constraint $f \leq 3$

For $f_{\text{max}} = 3$, the exact distribution of branch points is known from Eq. (3.31). This allows the functionality sampling to be separated cleanly from the sampling of a tree at fixed functionality composition.

First sample N_3 from $p(N_3 \mid N_{\text{tree}}, \mu_3)$. The remaining node counts follow from Eqs. (3.6) and (3.7). Next assign the N_{tree} labels uniformly to the N_1 , N_2 and N_3 functionality classes. For a label i , define its remaining Prüfer-code multiplicity as

$$m_i = f_i - 1. \quad (5.1)$$

At each of the $N_{\text{tree}} - 2$ code positions, choose label i with probability

$$p_i = \frac{m_i}{\sum_j m_j}, \quad (5.2)$$

append it to the code, and decrease m_i by one. Equivalently, this procedure samples random permutations of allowed Prüfer codes for the given functionality set. An initially unused $f = 3$ label therefore has twice the probability of an $f = 2$ label because it has two remaining occurrences rather than one. The complete procedure, including simultaneous decoding of the connectivity, is illustrated in Table A.3.

For $f_{\text{max}} = 3$, evaluating the discrete distribution $p(N_3 \mid N_{\text{tree}}, \mu_3)$ once requires $\mathcal{O}(N_{\text{tree}})$ work. After this preprocessing step, each independent tree again requires $\mathcal{O}(N_{\text{tree}})$ operations.

5.1.3 Arbitrary functionality constraints

When more than three functionality classes are allowed, the two tree constraints Eqs. (3.3) and (3.4) no longer reduce to a tree composition $\{N_f\}$ that depends on a single variable.

Summing multiplicities over all admissible compositions therefore becomes increasingly expensive when the number of allowed functionalities grows.

An exact alternative is to sample the labelled functionality sequence $\{f_i\}$ recursively for an arbitrary set of chemical potentials $\{\mu_f\}$. In Sec. A.7 we formulate this construction. We start with the last label added to the Prüfer code and base the probability of its functionality f_i on the chemical potentials and the number of ways the remaining $N_{\text{tree}} - f_i - 1$ entries of the Prüfer code can be arranged. This process continues until all tree nodes have been assigned a functionality and the steps above can be repeated for sampling a Prüfer code and corresponding labelled tree configuration.

The required probabilities can be tabulated before the first sample. With the implementation described in the appendix, this preprocessing requires at most $\mathcal{O}(N_{\text{tree}}^3)$ operations, whereas drawing a functionality sequence and the corresponding Prüfer code remains linear in N_{tree} .

As an alternative, one may also approximate the average tree node composition of an ensemble using the methods of Sec. C.3. Expanding for different sets $\{N_f\}$ around this maximum until a cut-off yields an approximate $P(\{N_f\}|N_{\text{tree}}, \{\mu_f\})$ that can be sampled.

5.1.4 Sampling other codes

Prüfer codes are not the only one-dimensional sequences that encode labelled trees. Alternative $N_{\text{tree}} - 2$ -integer sequences, including Blob and Dandelion codes, have been developed more recently [71]. For the task of decoding a *bare* code into a tree, these representations can be more effective. Ref. [72], for example, reports decoding routines that are roughly twice as fast as decoding conventional Prüfer codes to a labelled tree. One reason is that a Prüfer code does not store the graph connectivity *locally* in its code. If the code is the only available object, the multiplicities of all labels must first be read in order to determine the functionalities before the connectivity can be reconstructed. The latter extra $\mathcal{O}(N_{\text{tree}})$ operations are suppressed in the Blob and Dandelion code.

That advantage is, however, not present for the sampler developed here. Before any tree code is generated, the functionality sequence itself must already be drawn from the ensemble. This information is required for constructing Prüfer codes as well as Blob and Dandelion codes. It therefore remains an open question whether Blob and Dandelion codes can speed up sampling significantly in this scenario as well.

5.2 Amoeba sampling at fixed N_{tree}

Direct Prüfer sampling relies on the exact results for ideal trees found in Chapter 3. Once excluded volume or other spatial interactions are included, the connectivity and embedding no longer factorize and sampling cannot be directly done from the exact partition function. We use variants of the Amoeba algorithm [73], which equilibrate trees by cutting and pasting leaves sequentially.

The algorithm follows the branched-polymer analogue of the reptation, or slithering-snake, Monte Carlo algorithm for linear chains [74]. In reptation a segment is removed from one chain end and regrown at an end, so that a linear chain loses memory of its initial contour on a characteristic scale of order N_{lin}^2 Monte Carlo steps. The Amoeba update retains this end-pruning logic but allows the detached leaf to be attached to any admissible tree node. Because different attachment functionalities change the number of endpoints and branch points, their proposal probabilities and Boltzmann weights must be included explicitly.

Throughout this section the tree size is fixed, functionality is restricted to $f \leq 3$, and the branching contribution to the Hamiltonian is $-\mu_3 N_3$ [Eq. (3.11)]. For a proposed transition from state $|i\rangle$ to state $|f\rangle$, the Metropolis–Hastings construction [75] gives

$$\text{acc}_{|i\rangle \rightarrow |f\rangle} = \min \left\{ 1, \frac{w(|f\rangle) \alpha(|f\rangle \rightarrow |i\rangle)}{w(|i\rangle) \alpha(|i\rangle \rightarrow |f\rangle)} \right\}, \quad (5.3)$$

where w is the equilibrium statistical weight and α is the proposal probability. In the ideal case, $w \propto \exp(\beta\mu_3 N_3)$. With interactions, the same expression is used with the appropriate additional Boltzmann factor from $\mathcal{H}_{\text{int}}$.

The three variants below differ predominantly in how trial moves are proposed.

5.2.1 Original Amoeba algorithm

A trial move of the original Amoeba algorithm proceeds as follows:

1. choose uniformly a leaf A with $f_A = 1$ and identify its neighbour B;
2. detach A from B, producing the intermediate tree in which $f_B \rightarrow f_B - 1$;
3. choose a target node C uniformly among all nodes of the intermediate tree with $f_C < 3$;
4. choose one of the c lattice directions from C and place A at the corresponding neighbouring lattice site;
5. accept or reject the proposed state using Eq. (5.3); on acceptance, set $f_C \rightarrow f_C + 1$, replace bond A–B by A–C, and update the coordinate of A.

In an interacting system, any target placement with zero Boltzmann weight is rejected.

With the proposal convention used in the original implementation, the acceptance probability is [73, 41]

$$\text{acc}_{|i\rangle \rightarrow |f\rangle} = \min \left\{ 1, \frac{N_1(|i\rangle)}{N_1(|f\rangle)} \exp[\beta\mu_3(N_3(|f\rangle) - N_3(|i\rangle))] \right\}. \quad (5.4)$$

A practical implementation maintains the local information needed to update a move without reconstructing the complete tree. In the implementation used here the relevant datasets are

- the connectivity graph, stored as the list of directly connected neighbours of every node;

- the spatial coordinate of every node;
- the functionality f_i of every node;
- dynamically updated lists of the $f = 1$, $f = 2$, and $f = 3$ nodes used for randomly selecting leaves to cut and nodes to attach to.
- the compositional counts N_1 , N_2 , and N_3 .

Because one move changes only a small neighbourhood, all of these datasets can be updated efficiently.

Each possible labelled tree connectivity is in principle accessible through the repeated procedure of cutting and attaching leaves. Therefore, the algorithm is ergodic in principle. However, spatial interactions and, as we will discuss, statistical weights, can restrict the algorithm significantly, motivating the need for the more advanced algorithms described below.

5.2.2 Semi-kinetic Amoeba algorithm

The original Amoeba algorithm does not account for the possible very different statistical weights associated with moves that create or remove a branch point. The semi-kinetic variant introduced here biases the attaching step using the branching weights while retaining a uniformly chosen leaf for the pruning step. Conceptually this is related to kinetic or rejection-reduced Monte Carlo constructions developed under several names in the 1960s and 1970s [76, 77, 78]: moves that would otherwise have very small acceptance are proposed very little, but with acceptance ratios close to unity, whereas moves with already high acceptance are proposed a lot without a significant loss in their acceptance ratios.

We implement this here by first defining the $(N_{\text{tree}} - 1)$ -node intermediate tree as the transition state $|t\rangle$. Equation (5.3) can then be written as

$$\text{acc}_{|i\rangle \rightarrow |f\rangle} = \min \left\{ 1, \frac{w(|f\rangle) \alpha(|f\rangle \rightarrow |t\rangle) \alpha(|t\rangle \rightarrow |i\rangle)}{w(|i\rangle) \alpha(|i\rangle \rightarrow |t\rangle) \alpha(|t\rangle \rightarrow |f\rangle)} \right\}. \quad (5.5)$$

We still select the leaf to be cut uniformly, such that:

$$\alpha(|i\rangle \rightarrow |t\rangle) = \frac{1}{N_1(|i\rangle)}, \quad \alpha(|f\rangle \rightarrow |t\rangle) = \frac{1}{N_1(|f\rangle)}. \quad (5.6)$$

However the node to attach to is instead drawn according to the statistical weight of the resulting state,

$$\alpha(|t\rangle \rightarrow |i\rangle) = \frac{w(|i\rangle)}{z_{|t\rangle}}, \quad \alpha(|t\rangle \rightarrow |f\rangle) = \frac{w(|f\rangle)}{z_{|t\rangle}}, \quad (5.7)$$

where $z_{|t\rangle}$ is the sum of the weights of all allowed states after attaching a single leaf. The explicit equilibrium weights now cancel from Eq. (5.5), leaving

$$\text{acc}_{|i\rangle \rightarrow |f\rangle} = \min \left\{ 1, \frac{N_1(|i\rangle)}{N_1(|f\rangle)} \right\}. \quad (5.8)$$

For $f \leq 3$, attachment to a leaf creates an $f = 2$ node and carries no branching weight, whereas attachment to an $f = 2$ node creates a branch point and carries the factor $\exp(\beta\mu_3)$. Thus

$$z_{|t\rangle} = N_1(|t\rangle) + N_2(|t\rangle)e^{\beta\mu_3}. \quad (5.9)$$

A node to attach to can therefore be chosen by drawing a random number $q \in [0, z_{|t\rangle})$. If $q < N_1(|t\rangle)$ choose uniformly among the leaves, and otherwise choose uniformly among the $f = 2$ nodes. In the final step of a semi-kinetic move, choose the lattice orientation of the new bond with respect to the new node, and the proposal is accepted with Eq. (5.8) together with any interaction Boltzmann factor that could be required.

The adjective “semi-kinetic” refers to the difference in choosing pruned leaves and attachment sites: The leaf to be pruned is still selected uniformly, whereas the attachment step is biased by equilibrium weights. No additional global data structure is required relative to the original Amoeba algorithm. The extra operations required are the evaluation of the two functionality-class weights and the weighted choice of the target class. Their cost is therefore $\mathcal{O}(1)$ operations per attempted move and does not change the computational time associated with a single Monte Carlo step significantly. Fully kinetic constructions and alternative choices of transition state are discussed in Sec. 7.7.1.

5.2.3 Double-leaf Amoeba algorithm

At large positive branching activity, many branch points occur close to the tree periphery. The double-leaf variant augments the single-leaf move set with a move that transports two leaves attached to the same $f = 3$ parent.

A *double leaf* is the pair of $f = 1$ nodes attached to an $f = 3$ node whose third bond connects the pair to the rest of the tree. The algorithm samples one object from the combined pool of N_1 single leaves and N_{DL} double leaves:

1. choose uniformly one single-leaf or double-leaf object;
2. for a single leaf, perform the semi-kinetic update described above;
3. for a double leaf, detach both leaves from their common parent B, so that $f_B : 3 \rightarrow 1$;
4. choose uniformly a leaf C of the resulting transition tree and attach both detached leaves to C, so that $f_C : 1 \rightarrow 3$; choose lattice orientations for the two new bonds;
5. accept or reject the proposal and update connectivity, coordinates and functionality lists.

For the branching Hamiltonian in Eq. (3.11), a pure double-leaf move conserves N_3 and therefore connects states with equal branching weight. Single-leaf moves are retained so that the move set can change the number and locations of branch points rather than only transport peripheral ones.

With our uniform object-selection convention, the acceptance probability is

$$\text{acc}_{|i\rangle \rightarrow |f\rangle} = \min \left\{ 1, \frac{N_1(|i\rangle) + N_{\text{DL}}(|i\rangle)}{N_1(|f\rangle) + N_{\text{DL}}(|f\rangle)} \right\}. \quad (5.10)$$

The implementation therefore requires, in addition to the data used by the semi-kinetic algorithm, a dynamically updated list of double leaves and local bookkeeping of the number of leaf neighbours of each node. The active neighbourhood of an accepted move is correspondingly larger than for a single-leaf update because changing one branch point can create or destroy double-leaf status on adjacent nodes. Nevertheless, all affected entries are local and the additional work remains $\mathcal{O}(1)$ per Monte Carlo step; there is no N_{tree} -dependent overhead associated merely with maintaining the double-leaf list. The effect of the enlarged move set on equilibration is analysed in Chapter 7.

5.2.4 Generalization to other Hamiltonians

The semi-kinetic construction is not tied to $f \leq 3$. For a general set of functionality chemical potentials $\{\mu_f\}$ in Eq. (3.10), each allowed node for attachment can be weighted by the Boltzmann factor associated with the resulting change in node functionalities. The transition-state sum z then generalizes to the sum over all allowed target classes. The double-leaf idea similarly extends to multiple-leaf moves when a higher-functionality junction carries several peripheral leaves. This increases the number of locally maintained classes but does not by itself require a system-size-dependent amount of work per move.

Likewise, spatial interactions can be included through their Boltzmann factors. In very dense systems choosing targets without regard to occupancy can lead to a large number of trivially impossible proposals. This can be mitigated by defining the transition-state sum only over spatially available targets, or, more generally, by biasing the proposal simultaneously by functionality and spatial Boltzmann weight. For hard excluded volume, a target site with vanishing weight is simply excluded from the admissible set or rejected. A construction along these lines has recently been implemented in Ref. [79]. Such modifications alter efficiency but not the equilibrium target distribution when their proposal probabilities are included consistently in Eq. (5.3).

5.3 Local Monte Carlo dynamics of tightly double-folded rings

The tree-level algorithms above change connectivity directly. A complementary description keeps the sequential ring contour as the dynamical object and evolves the elastic tightly double-folded ring through local monomer displacements. We use the lattice dynamics introduced for this model in Refs. [20, 22].

5.3.1 Local trial moves on the ring level

The state contains the coordinates of the ring monomers, the coordinates and connectivity of the underlying tree nodes, and the assignment of ring monomers to tree nodes. Two elementary classes of local moves are used:

1. **Repton moves** transport stored length along the ring. A ring monomer moves longitudinally across an extended bond to the neighbouring tree node. The underlying

tree topology and N_{tree} remain unchanged. These moves descend from the repton description introduced for polymer melt dynamics in Ref. [80].

2. **Hairpin moves** create, extend, shorten or remove a terminal branch. A monomer flanked by two zero-length bonds can move transversely to a neighbouring lattice site, replacing the two reptons by a pair of oppositely oriented extended bonds and creating a new $f = 1$ tree node. The inverse move removes such a bond pair. Consequently N_{tree} changes by one.

Both move classes preserve ring closure and the strictly double-folded tree representation. In this explicit implementation one therefore maintains, simultaneously, (i) the ring-monomer coordinates, (ii) the tree-node coordinates and dynamically changing tree connectivity, and (iii) the assignment of each ring monomer to a tree node; the sequential ring connectivity itself is fixed by construction.

With uniformly selected local proposals, Eq. (5.3) is used with the ring Hamiltonian and any spatial interactions. The proposal asymmetry associated with the choice of a new lattice direction gives

$$\frac{\alpha(|f\rangle \rightarrow |i\rangle)}{\alpha(|i\rangle \rightarrow |f\rangle)} = 1 \quad \text{for repton moves,} \quad (5.11)$$

$$= \frac{1}{c} \quad \text{for hairpin retraction,} \quad (5.12)$$

$$= c \quad \text{for hairpin extension,} \quad (5.13)$$

where c is the lattice coordination number. This is the same c that appears in the ideal embedding factor $c^{N_{\text{tree}}-1}$.

5.3.2 Extensions of the local move set

The repton and hairpin moves form an ergodic set of local moves when the ring monomers are treated as the primary degrees of freedom, but additional local dynamics can be introduced without changing the target equilibrium ensemble. Brownian tree-node moves, in which all ring monomers assigned to a single tree node are displaced together by one lattice step without stretching any ring bond beyond the lattice spacing, were implemented in Ref. [22]. In dense conditions, combining those moves with the monomer-level dynamics was found to accelerate relaxation relative to using only one of the two move families.

A second example is provided by branch-sliding moves [18, 19]. These moves alter the tree connectivity without changing its conformation. A tree bond between two nodes is then replaced by a bond between one of the original nodes and an adjacent tree node. These moves allow branches emanating from a particular branch point to “slide” along a linear segment, thereby modifying connectivity more directly than a sequence of elementary hairpin moves. These moves were introduced as a possible dynamic mechanism that accelerates finding contacts in bacterial genomes [18].

5.4 Semi-grand-canonical tree sampling of fixed-length rings

A fixed-length elastic ring has fixed N_{ring} but a fluctuating number of extended bonds, so the size N_{tree} of its underlying tree will fluctuate. The fixed- N_{tree} Amoeba moves of Sec. 5.2 cannot sample this ensemble because every removed leaf is reattached within the same Monte Carlo step.

We therefore decouple cutting and pasting. The resulting semi-grand-canonical tree algorithm targets the one-tree–one-ring representation derived in Sec. 4.3: a sampled tree carries the tree-size weight $w_{\text{tree}}(N_{\text{tree}}, N_{\text{ring}})$ from Eq. (4.23) and the shifted branch point chemical potential

$$\beta\mu_3 = \beta\tilde{\mu}_3 + \ln(2). \quad (5.14)$$

After equilibration, a particular ring conformation is obtained by wrapping the tree as in Table A.4 and distributing the required reptons as described in Chapter 4.

5.4.1 Monte Carlo move classes

For $f \leq 3$, four different move classes alter the allowed tree sizes and compositions:

1. cut a leaf from an $f = 2$ parent: $N_2 \rightarrow N_2 - 1$ and $N_{\text{tree}} \rightarrow N_{\text{tree}} - 1$;
2. cut a leaf from an $f = 3$ parent: $N_3 \rightarrow N_3 - 1$, $N_2 \rightarrow N_2 + 1$, $N_1 \rightarrow N_1 - 1$ and $N_{\text{tree}} \rightarrow N_{\text{tree}} - 1$;
3. paste a leaf onto an $f = 1$ node: $N_2 \rightarrow N_2 + 1$ and $N_{\text{tree}} \rightarrow N_{\text{tree}} + 1$;
4. paste a leaf onto an $f = 2$ node: $N_3 \rightarrow N_3 + 1$, $N_2 \rightarrow N_2 - 1$, $N_1 \rightarrow N_1 + 1$ and $N_{\text{tree}} \rightarrow N_{\text{tree}} + 1$.

Classes 1 and 3 are inverse moves, as are classes 2 and 4. Together with the special treatment of the $N_{\text{tree}} = 1$ boundary described below, they sample tree sizes in the entire interval $1 \leq N_{\text{tree}} \leq N_{\text{ring}}/2 + 1$.

5.4.2 Proposal and acceptance probabilities

As in the semi-kinetic fixed-size algorithm, we bias proposal probabilities by the chemical potential μ_3 . For a state $|i\rangle$ with $N_{\text{tree}} \geq 2$, we thus assign the proposal weights

$$\alpha(\text{cut any leaf}) \propto N_1(|i\rangle), \quad (5.15)$$

$$\alpha(\text{paste to any leaf}) \propto N_1(|i\rangle), \quad (5.16)$$

$$\alpha(\text{paste to any linear node}) \propto N_2(|i\rangle)e^{\beta\mu_3}. \quad (5.17)$$

Thus, for a particular node, we have:

$$\alpha(\text{cut specific leaf}) = \frac{1}{z_{|i\rangle}}, \quad (5.18)$$

$$\alpha(\text{paste to specific leaf}) = \frac{1}{z_{|i\rangle}}, \quad (5.19)$$

$$\alpha(\text{paste to specific linear node}) = \frac{e^{\beta\mu_3}}{z_{|i\rangle}}, \quad (5.20)$$

with

$$z_{|i\rangle} = 2N_1(|i\rangle) + N_2(|i\rangle)e^{\beta\mu_3}. \quad (5.21)$$

For paste moves, one of the c lattice directions is additionally chosen. Equivalently, this orientation proposal may be regarded as the origin of the factors $c^{N_{\text{tree}}-1}$ in the conformational statistical weight below.

Because the factors $e^{\beta\mu_3}$ associated with creating a branch point have been included in the proposal distribution, the explicit branching weight cancels from the ideal acceptance ratio. Using the tree-size weight from Eq. (4.23),

$$\text{acc}_{|i\rangle \rightarrow |f\rangle} = \min \left\{ 1, \frac{w_{\text{tree}}(N_{\text{tree}}(|f\rangle), N_{\text{ring}}) z_{|i\rangle} c^{N_{\text{tree}}(|f\rangle)-1}}{w_{\text{tree}}(N_{\text{tree}}(|i\rangle), N_{\text{ring}}) z_{|f\rangle} c^{N_{\text{tree}}(|i\rangle)-1}} \right\}. \quad (5.22)$$

For $N_{\text{tree}}(|i\rangle) \geq 3$ when cutting and $N_{\text{tree}}(|i\rangle) \geq 2$ when pasting, the factorials reduce this expression to

$$\text{acc}_{\text{cut}} = \min \left\{ 1, \frac{z_{|i\rangle}}{z_{|f\rangle}} \frac{(2N_{\text{tree}}(|i\rangle) - 3)(2N_{\text{tree}}(|i\rangle) - 4)}{(N_{\text{ring}} - 2N_{\text{tree}}(|i\rangle) + 3)(N_{\text{ring}} - 2N_{\text{tree}}(|i\rangle) + 4)} \frac{1}{c} \right\}, \quad (5.23)$$

$$\text{acc}_{\text{paste}} = \min \left\{ 1, \frac{z_{|i\rangle}}{z_{|f\rangle}} \frac{(N_{\text{ring}} - 2N_{\text{tree}}(|i\rangle) + 2)(N_{\text{ring}} - 2N_{\text{tree}}(|i\rangle) + 1)}{(2N_{\text{tree}}(|i\rangle) - 1)(2N_{\text{tree}}(|i\rangle) - 2)} c \right\}. \quad (5.24)$$

When interactions are present, the right-hand side of Eq. (5.22) is multiplied by the corresponding Boltzmann-weight ratio, for example $\exp[-\beta(\mathcal{H}_{\text{int}}(|f\rangle) - \mathcal{H}_{\text{int}}(|i\rangle))]$. Hard-overlap proposals therefore have zero acceptance.

5.4.3 The single-node boundary

Equation (5.23) does not cover the transition between $N_{\text{tree}} = 2$ and the single-node state. To include $N_{\text{tree}} = 1$, the single node is treated as an endpoint for proposal bookkeeping, so that

$$z_{|N_{\text{tree}}=1\rangle} = 2. \quad (5.25)$$

With the boundary normalization used in the present implementation, Eq. (5.22) gives the special acceptance probabilities

$$\text{acc}_{|i\rangle \rightarrow |f\rangle} = \begin{cases} \min \left\{ 1, \frac{2}{N_{\text{ring}}(N_{\text{ring}} - 1)} \frac{1}{c} \right\}, & N_{\text{tree}} : 2 \rightarrow 1, \\ \min \left\{ 1, \frac{N_{\text{ring}}(N_{\text{ring}} - 1)}{2} c \right\}, & N_{\text{tree}} : 1 \rightarrow 2, \\ 0, & \text{attempted cutting at } N_{\text{tree}} = 1. \end{cases} \quad (5.26)$$

These expressions complete the proposal rules at the lower tree-size boundary.

5.4.4 Complete semi-grand-canonical algorithm

For a ring of fixed length N_{ring} and ring branching potential $\tilde{\mu}_3$, first set

$$N_{\text{tree,max}} = \frac{N_{\text{ring}}}{2} + 1, \quad \beta\mu_3 = \beta\tilde{\mu}_3 + \ln(2). \quad (5.27)$$

Maintain the active tree nodes, their coordinates and connectivity, lists for the $f = 1$, $f = 2$ and $f = 3$ classes, and a list of unused node labels. One Monte Carlo proposal is then generated as follows:

1. compute $z_{|i\rangle}$ from Eq. (5.21), or use Eq. (5.25) for $N_{\text{tree}} = 1$;
2. draw p uniformly from $[0, z_{|i\rangle})$;
3. choose the move class:
 - (a) if $0 \leq p < N_1$, choose a leaf uniformly and propose cutting it;
 - (b) if $N_1 \leq p < 2N_1$, choose an $f = 1$ target uniformly and propose adding an unused node at one of its c neighbouring lattice positions;
 - (c) if $2N_1 \leq p < z_{|i\rangle}$, choose an $f = 2$ target uniformly and propose adding an unused node at one of its c neighbouring lattice positions.

Paste proposals are disabled when $N_{\text{tree}} = N_{\text{tree,max}}$ or when no unused label is available; cut proposals are disabled at $N_{\text{tree}} = 1$.

4. evaluate Eq. (5.23) or the boundary expression (5.26), together with any interaction factor. On acceptance, update connectivity, coordinates, functionality lists and the unused-label list.
5. repeat the Markov chain for the required equilibration and sampling interval; the corresponding diagnostics are defined in Chapter 7.

After a tree sample has been generated, compute $N_{\text{rep}} = N_{\text{ring}} - 2(N_{\text{tree}} - 1)$. A ring conformation representing that tree is obtained by the wrapping procedure in Table A.4 and distributing the N_{rep} zero-length bonds according to the construction of Chapter 4.

The implementation here maintains the following data structures for efficiency: coordinates and connectivity of all labelled nodes that form a tree; lists of active nodes with $f = 1$, $f = 2$, and $f = 3$; the functionality of each active node; an ordered list of unused labels from which newly pasted nodes are drawn; and a lattice-occupation structure used to reject overlaps efficiently. Storing the inverse indices of nodes within the functionality

lists allows deletions and class changes to be performed in constant time.

Given any admissible starting state—for example one-node trees—the only inputs required for sampling the elastic-ring ensemble after a sufficient number of Monte Carlo steps are N_{ring} and $\tilde{\mu}_3$.

5.4.5 Generalizations

The semi-grand-canonical Amoeba algorithm is not intrinsically tied to the tightly double-folded elastic ring ensemble. For a different fluctuating tree-size ensemble, the factor $w_{\text{tree}}(N_{\text{tree}}, N_{\text{ring}})$ in Eq. (5.22) can be replaced by the desired size-dependent statistical weight with the corresponding change carried through the acceptance ratios. In the ring ensembles evaluated in Chapters 6 and 7, the average tree size $\langle N_{\text{tree}} \rangle$ is such that acceptance ratios are of order unity for the above implementation. Therefore, acceptance is not causing any additional slowdown of the equilibration process. If acceptance becomes a practical bottleneck in other ensembles, these factors can nevertheless be absorbed into the proposal distribution in the same spirit as the branching chemical potential.

The fluctuating-size systems considered later do not enter the high- μ_3 regime for which we introduced double-leaf moves in Sec. 5.2.3. For that reason, the semi-grand-canonical implementation introduced here contains only single-node cut and paste moves. A multiple-leaf generalization of the semi-grand-canonical Amoeba algorithm can nevertheless be designed straightforwardly for strongly branching ensembles.

5.5 Alternative tree-sampling algorithms and their ensembles

The algorithms above were chosen for a particular state space: labelled annealed tree connectivity plus a lattice embedding, and, for the ring representation, the additional wrapping and stored-length multiplicities. Other well-established algorithms have been designed to sample tree-like states, but do not necessarily sample the same distributions. Here, we review a few of those algorithms and discuss how they relate to the ensembles of interest discussed here.

5.5.1 Non-local branch-moving algorithms

The Monte Carlo algorithms discussed here transport only a small amount of mass in one update: one node for an ordinary Amoeba move and two nodes for a double-leaf move. This is advantageous when excluded volume effects are implemented at high lattice occupation, as a collective displacement of a large sub-tree would often be rejected because of its conformational incompatibility with the environment. For an isolated or dilute tree, however, moving only one node is not necessarily optimal.

Janse van Rensburg and Madras [81] demonstrated that self-avoiding lattice trees can be equilibrated efficiently using non-local branch-moving updates. Instead of selecting an endpoint, one cuts a bond so that an entire rooted subtree of arbitrary mass is detached

and subsequently attached at another position of the tree. Both the connectivity and the spatial conformation of a labelled tree can then change substantially in one Monte Carlo step.

Those algorithms were originally not constructed for controlled branching activity. The ideas developed above suggest a natural extension: define a transition state by pruning a randomly selected bond, then choose the node to glue the branch onto with a semi-kinetic weight that includes the functionality change. At large positive μ_3 , one can similarly imagine updates in which two branches are removed simultaneously from a single branch node and attached to an endpoint. This generalizes the double-leaf Monte Carlo moves to “double-branch” moves.

5.5.2 Growth and sequential-importance algorithms

A different strategy constructs a branched object sequentially from a single seed. This was in fact the process that Flory [2, 3, 4] had in mind for modelling chemical branching. It involves adding nodes of given functionality sequentially to open bonds based on their chemical potential. Once enough endpoints are added, such that the constraint in Eq. (3.4) is satisfied, the process stops. This procedure generates an ensemble of polydisperse trees in linear time. This algorithm, however, is extremely sensitive to the probabilities of producing endpoints and branch points: if endpoints appear too frequently growth terminates prematurely, whereas if they are too rare the process may reinforce itself and trees continue to grow to infinite sizes. The same difficulty makes direct growth of wrapping codes less straightforward than the fixed-size Prüfer construction.

One can also grow a randomly branched polymer by directly percolating the embedding lattice. Starting from one occupied site, candidate neighbouring sites are continuously tested and added until a target size is reached. This procedure creates tree topologies, as long as already placed neighbouring sites are not connected to form cycles. Its naive implementation does not sample an equilibrium randomly branched polymer ensemble, because the probability of the final object depends on the entire growth history. Nodes introduced early can participate in many more subsequent choices than late nodes, and are thus much more likely to branch, creating dendrimeric structures.

This history dependence can be corrected by assigning a compensating statistical weight at each new growth step. Hsu, Nadler, and Grassberger [82] developed an efficient realization for lattice animals and trees using a percolation-like proposal process combined with the pruned-enriched Rosenbluth method (PERM) [83]. Candidate growth sites are activated with a finite probability p ; the inverse generation probability contributes to the statistical weight of a polymer, so p controls efficiency rather than defining the final ensemble. As growth continues, the weights become broadly distributed. PERM controls this variance by pruning low-weight partial configurations and cloning high-weight ones, allowing an ensemble of lattice trees with thousands of sites to be sampled efficiently.

The microscopic state in those lattice-tree studies is nevertheless different from labelled

trees. In a labelled tree, each polymer node carries a label and the abstract connectivity graph $\mathcal{G}$ is an independently identifiable part of the state. The complementary spatial embedding specifies the conformation. Two states can therefore share an embedding but differ in labelled connectivity, or share a connectivity and differ in embedding. In the ensembles such as those discussed in Ref. [82], the occupied lattice sites themselves carry the only important label, and nearest-neighbour adjacency defines the available bonds. The *embedded subgraph* of the lattice, rather than an independently labelled abstract tree followed by an embedding, is the microscopic state. The two classes of algorithms can therefore not straightforwardly be assumed to sample the same distributions.

5.5.3 Uniform spanning trees and Wilson’s algorithm

A related but distinct problem is to generate a completely space-filling tree on a finite lattice. Such an object can be viewed as a maze: every lattice site belongs to the tree and the absence of loops guarantees a unique path between every pair of sites. In graph-theoretical language this maze is the embedded subgraph of the underlying lattice, called its *spanning tree*.

Wilson’s algorithm [84] samples these mazes efficiently. One lattice site is chosen as the root and forms the initial tree. In each consecutive step a random walk is launched on the lattice to find the existing tree. Whenever a walk intersects itself in this process, the formed loop is erased, before continuing. When the resulting loop-erased random walk first reaches the already existing tree, the entire path is attached. The construction is repeated from sites not yet included until the tree spans the entire lattice.

Applied to a finite lattice, Wilson’s method produces a uniform ensemble of maze-like, space-filling lattice trees. This uniformity should again not be confused with uniform sampling of all abstract labelled trees at fixed N_{tree} . In the spanning-tree problem the sites and admissible bonds are fixed in advance by the embedding lattice; in the present polymer model the abstract connectivity graph itself is annealed and is only subsequently embedded in space. In future implementations it might be interesting to see, however, how the spanning trees’ configurations and conformations differ from melts of labelled trees.

5.6 Chapter summary

This chapter introduced four complementary routes through the state spaces developed in Chapters 3 and 4. Exact Prüfer sampling produces independent ideal labelled trees in linear time once the required functionality distribution has been prepared. Fixed-size Amoeba algorithms instead explore interacting or otherwise non-factorizable tree ensembles through local connectivity updates. Local ring dynamics samples the explicitly double-folded contour, while the semi-grand-canonical Amoeba construction samples the statistically equivalent fluctuating tree representation of a fixed-length elastic ring.

The next two chapters separate two questions relevant to algorithm evaluation. Chapter 6

asks whether these independent constructions reproduce the equilibrium distributions predicted by the exact theory and whether tree- and ring-level representations are statistically equivalent. Chapter 7 then asks how quickly they decorrelate, which physical modes control relaxation, and how computational cost scales with polymer size and branching activity.

Chapter 6

Validation and ensemble equivalence

Chapters 3–5 introduced two exact equilibrium theories and several independent ways of sampling them. The purpose of the present chapter is to establish that these routes lead to the same statistical ensembles. This question is distinct from equilibration speed: here a method is judged by the equilibrium distribution it produces, whereas Chapter 7 concerns the time required to reach and decorrelate that distribution.

This validation can be done in several ways. Here, we first benchmark the fixed-size Amoeba algorithms against previously published data and against one another. We then use a histogram-reweighting approach to approximate the full configurational N_3 -dependent entropy in the sampled ensembles. Next, independent Amoeba-generated and Prüfer-sampled trees test the structural predictions derived in Chapter 3. Finally, we test the elastic-ring theory and the exact tree-ring mapping by comparing local ring dynamics with tree-level sampling, including interacting systems.

6.1 Validation of fixed-size Amoeba tree sampling

We begin with ideal randomly branched polymers of fixed size N_{tree} and maximum functionality $f_{\text{max}} = 3$. We test the original, semi-kinetic, and double-leaf Amoeba algorithms defined in Sec. 5.2. Unless stated otherwise, the simulations use the three-dimensional FCC lattice. Two observables are used repeatedly: the branch point number N_3 and the squared radius of gyration R_g^2 , defined in Sec. 2.1.2. These data were obtained after long runs of 1.2×10^8 Monte Carlo steps per polymer starting from linear chains. The evidence that these run lengths are sufficient, and the comparison of decorrelation times between algorithms, belong to Chapter 7.

6.1.1 Historical benchmark

The original Amoeba algorithm has previously been used to simulate ideal annealed trees at $\beta\mu_3 = 2$ [41]. The supplementary data of that work therefore provide an external

benchmark for the implementations used here. We repeated the calculation for

$$N_{\text{tree}} \in \{6, 11, 21, 46, 76, 151\},$$

using ensembles of 25000 independently sampled configurations for each size. Table 6.1 compares the mean gyration radius and mean number of branch points with the published two- and three-dimensional simple-cubic results.

$\langle R_g^2 \rangle$					
N_{tree}	2D SC [41]	3D SC [41]	Original	Semi-kinetic	Double-leaf
6	0.845 ± 0.003	0.851 ± 0.010	0.844 ± 0.002	0.849 ± 0.002	0.846 ± 0.002
11	1.389 ± 0.005	$1.378 \pm 0.008^*$	1.386 ± 0.003	1.381 ± 0.003	1.388 ± 0.003
21	2.195 ± 0.008	2.199 ± 0.005	2.210 ± 0.005	2.187 ± 0.005	2.204 ± 0.005
46	3.684 ± 0.009	3.688 ± 0.008	3.682 ± 0.008	3.670 ± 0.008	3.676 ± 0.008
76	5.031 ± 0.011	5.006 ± 0.012	5.017 ± 0.011	4.993 ± 0.011	4.998 ± 0.011
151	7.500 ± 0.022	7.468 ± 0.016	7.483 ± 0.017	7.488 ± 0.017	7.488 ± 0.017
$\langle N_3 \rangle$					
N_{tree}	2D SC [41]	3D SC [41]	Original	Semi-kinetic	Double-leaf
6	1.429 ± 0.006	1.450 ± 0.015	1.438 ± 0.004	1.429 ± 0.004	1.433 ± 0.004
11	3.334 ± 0.006	3.320 ± 0.012	3.323 ± 0.004	3.328 ± 0.004	3.329 ± 0.004
21	7.292 ± 0.009	7.305 ± 0.006	7.293 ± 0.006	7.290 ± 0.006	7.292 ± 0.006
46	17.206 ± 0.008	17.202 ± 0.008	17.194 ± 0.009	17.211 ± 0.009	17.211 ± 0.009
76	29.102 ± 0.009	29.120 ± 0.012	29.099 ± 0.011	29.106 ± 0.011	29.103 ± 0.011
151	58.862 ± 0.013	58.874 ± 0.017	58.872 ± 0.016	58.864 ± 0.016	58.882 ± 0.016

Table 6.1: Historical benchmark for the three Amoeba implementations at $\beta\mu_3 = 2$. The second and third columns reproduce the two- and three-dimensional simple-cubic results reported in Ref. [41]; the final three columns are the present three-dimensional FCC results. Errors are standard errors of the mean over 25000 configurations. *The $N_{\text{tree}} = 11$ three-dimensional value from Ref. [41] is shown with the round-off correction identified during the present analysis.

The agreement across lattices is expected for these ideal fixed-size ensembles. The connectivity distribution is independent of the lattice coordination number because the embedding factor $c^{N_{\text{tree}}-1}$ in Eq. (3.2) is common to all connectivities and cancels from normalized probabilities such as Eq. (3.31). Moreover, for an ideal embedding, Kramers' theorem expresses $\langle R_g^2 \rangle$ entirely through connectivity path lengths and the bond length b (Sec. 2.1.3).

6.1.2 Consistency between Amoeba variants across branching activity

Additionally, we compared the different Amoeba algorithms by varying the branch point chemical potential over the full crossover between linear chains and strongly branched trees. For

$$N_{\text{tree}} \in \{10, 20, 40, 80, 81, 161, 321\}, \quad -15 \leq \beta\mu_3 \leq 15,$$

we generated 500 independently sampled tree conformations for each parameter set with each of the three Amoeba algorithms. Figure 6.1 plots the semi-kinetic and double-leaf results directly against the original-Amoeba values.

Over the complete parameter range, both observables collapse onto the diagonal within statistical uncertainty. This comparison is useful because the three algorithms construct their trial moves differently while targeting the same equilibrium measure. Agreement therefore shows that the modified proposal rules do not change the sampled fixed- N_{tree} ensemble.

6.1.3 Recovering the labelled-tree configurational entropy

Agreement of a few mean observables is necessary but not sufficient to identify the sampled connectivity ensemble. A stronger test uses the N_3 histograms themselves. For a fixed (N_{tree}, μ_3) ensemble, Eq. (3.31) implies

$$h(N_3 | N_{\text{tree}}, \mu_3) \propto \Omega_{\text{tree}}(N_{\text{tree}}, N_3) e^{\beta \mu_3 N_3}, \quad (6.1)$$

where h denotes the normalized Monte Carlo histogram. For adjacent non-zero histogram bins,

$$\frac{\Omega_{\text{tree}}(N_{\text{tree}}, N_3)}{\Omega_{\text{tree}}(N_{\text{tree}}, N_3 + 1)} \simeq \frac{h(N_3 | N_{\text{tree}}, \mu_3)}{h(N_3 + 1 | N_{\text{tree}}, \mu_3)} e^{\beta \mu_3}. \quad (6.2)$$

Consequently, the entropy increment is

$$\frac{S_{\text{tree}}^{\text{conf}}(N_{\text{tree}}, N_3 + 1) - S_{\text{tree}}^{\text{conf}}(N_{\text{tree}}, N_3)}{k_B} \simeq \ln \left(\frac{h(N_3 + 1 | N_{\text{tree}}, \mu_3)}{h(N_3 | N_{\text{tree}}, \mu_3)} \right) - \beta \mu_3. \quad (6.3)$$

This form is simply the logarithm of Eq. (6.2); it is the quantity that can be accumulated across successive values of N_3 .

For each adjacent pair of populated bins we combine the estimates obtained at different μ_3 , using the available statistical uncertainties as weights, and reconstruct the entropy by discrete thermodynamic integration [85]. The integration constant is fixed by the exactly known linear-chain multiplicity,

$$S_{\text{tree}}^{\text{conf}}(N_{\text{tree}}, 0) = k_B \ln \left(\frac{N_{\text{tree}}!}{2} \right). \quad (6.4)$$

The result is shown in Fig. 6.2.

The reconstructed entropy agrees with the exact labelled-tree multiplicity over the accessible range of N_3 .

6.1.4 Excluding unlabelled trees

Following the reasoning in Sec. 3.3, a direct numerical check that Amoeba algorithms do not sample unlabelled trees is shown in Fig. 6.3. To make this comparison, we simulated FCC lattice trees of *unrestricted* functionality while varying a chemical potential μ_1 to

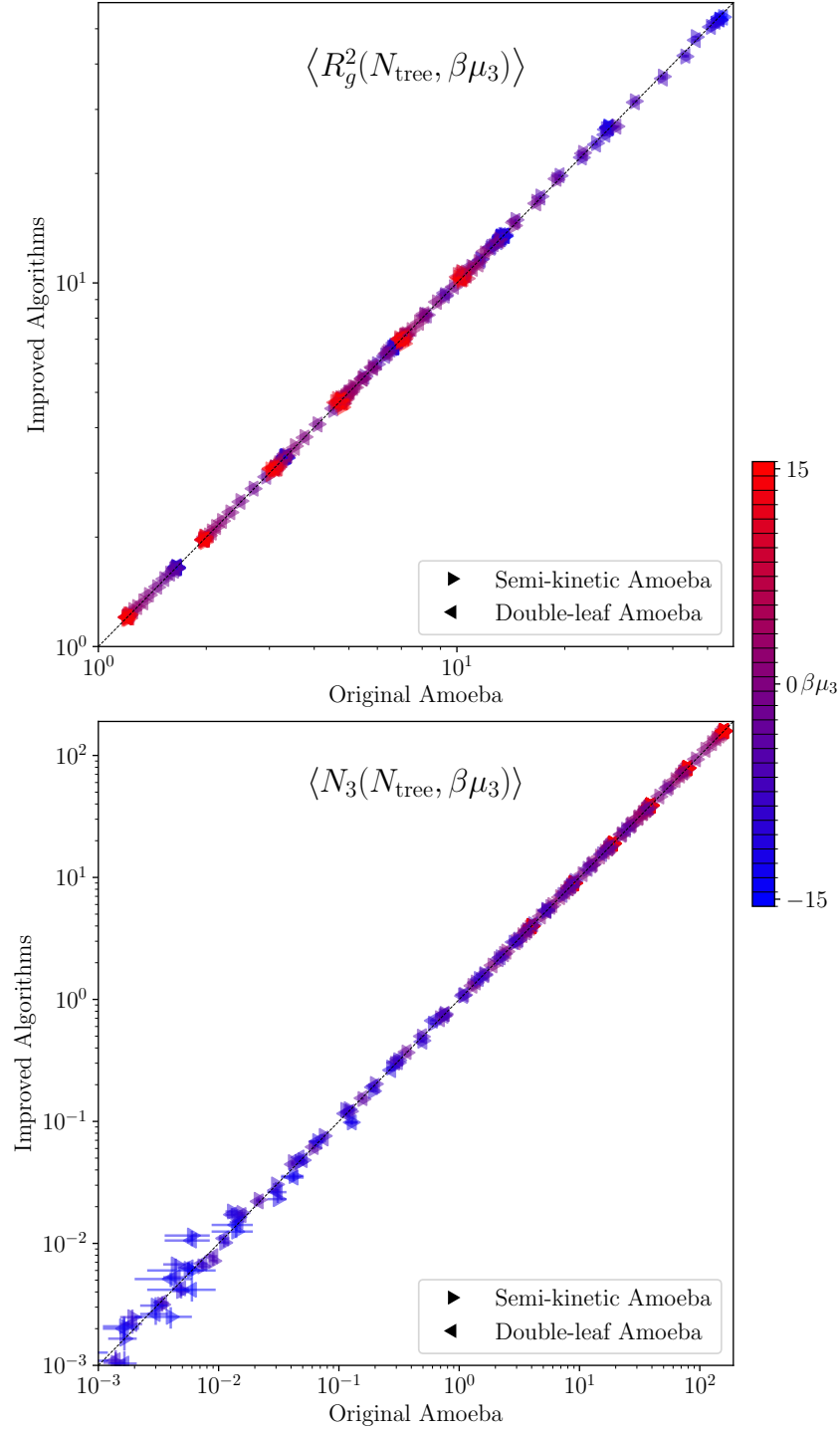


Figure 6.1: Direct comparison of equilibrium observables obtained with the three fixed-size Amoeba algorithms. The horizontal axis shows the original-Amoeba result; the vertical axis shows the semi-kinetic and double-leaf results. The upper panel compares $\langle R_g^2 \rangle$ and the lower panel $\langle N_3 \rangle$. Colours encode $\beta\mu_3$. The diagonal is $y = x$. Error bars are generally smaller than the symbol size except when the mean number of branch points becomes very small.

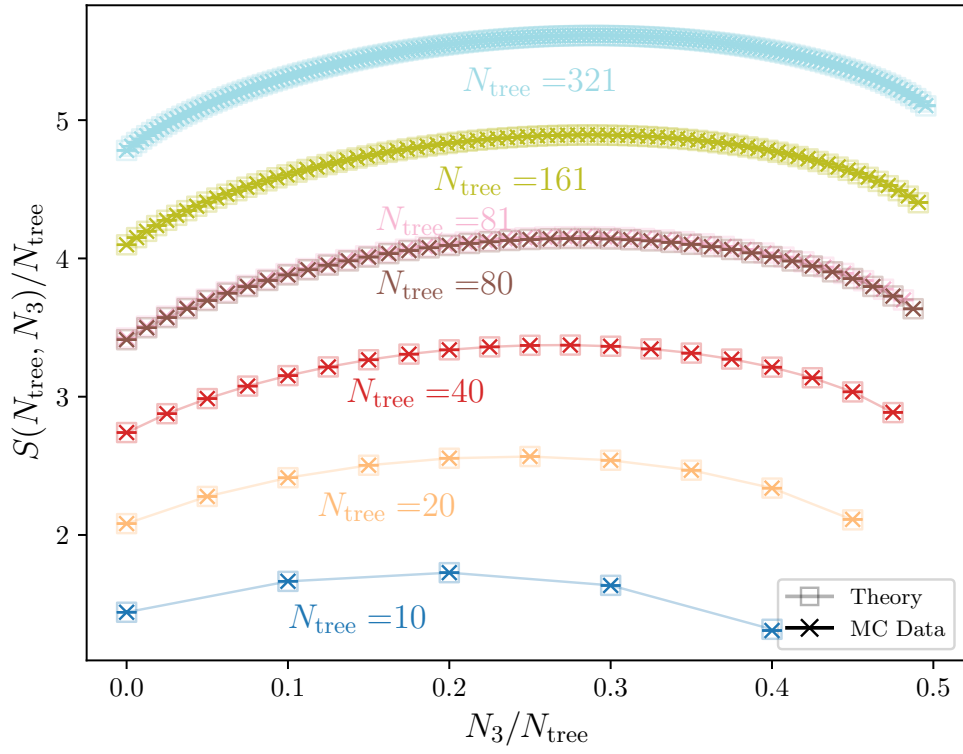


Figure 6.2: Connectivity entropy per node as a function of branch point fraction. Crosses are reconstructed from Monte Carlo histograms through Eqs. (6.2) and (6.3); square symbols are the exact labelled-tree result of Eq. (3.28). Curves for different N_{tree} are offset only by their finite-size dependence and cover the crossover from weak to strong branching.

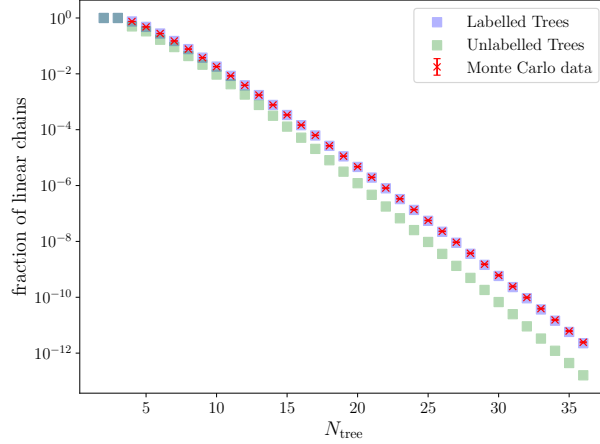


Figure 6.3: Fraction of linear chains with respect to all trees in the ensemble. Results for: labelled trees (exact, blue symbols), unlabelled trees (exact, green symbols) and randomly branched polymers on the 3D FCC lattice (Monte Carlo simulations, red symbols).

control the number of endpoints. By adapting the thermodynamic-integration strategy used above to find an N_1 -dependent entropy, we reconstructed the total number of trees up to $N_{\text{tree}} = 36$ and from it the fraction of linear chains in the ensemble. The unlabelled prediction uses the exact tabulated counts of Ref. [65], whereas the labelled prediction follows from Cayley’s formula. The Monte Carlo data agree with the labelled-tree prediction and not with the unlabelled one, confirming that the Amoeba algorithm indeed samples labelled trees.

6.2 Structural observables of the exact ideal-tree ensemble

In the previous section, we found that Amoeba algorithms sample labelled trees. We now test the structural consequences derived for this ensemble in Chapter 3.

6.2.1 Radius of gyration at fixed branching

Section 3.4.1 used Kramers’ theorem to obtain the classic result for $\langle R_g^2(N_{\text{tree}}, N_3) \rangle$ of Zimm and Stockmayer [6]. In Section 3.4.2 we extended these results for a fluctuating N_3 following the formulation of De Gennes [67] and later Daoud and Joanny [68]. In the latter ensemble a parameter λ controls the branching activity. By expressing λ as the ratio of branch points to total tree size as in Sec. 3.5, both results should give the same dependence of $\langle R_g^2(N_{\text{tree}}, \langle N_3 \rangle) \rangle$ on the expected number of branch points $\langle N_3(N_{\text{tree}}, \beta\mu_3) \rangle$. Figure 6.4 compares this prediction with Amoeba-generated ideal-tree simulation data over a broad range of branch point chemical potentials.

We observe a collapse of all data on a universal curve that is well approximated by the Zimm and Stockmayer result in the high-branching limit, and by the Daoud and Joanny

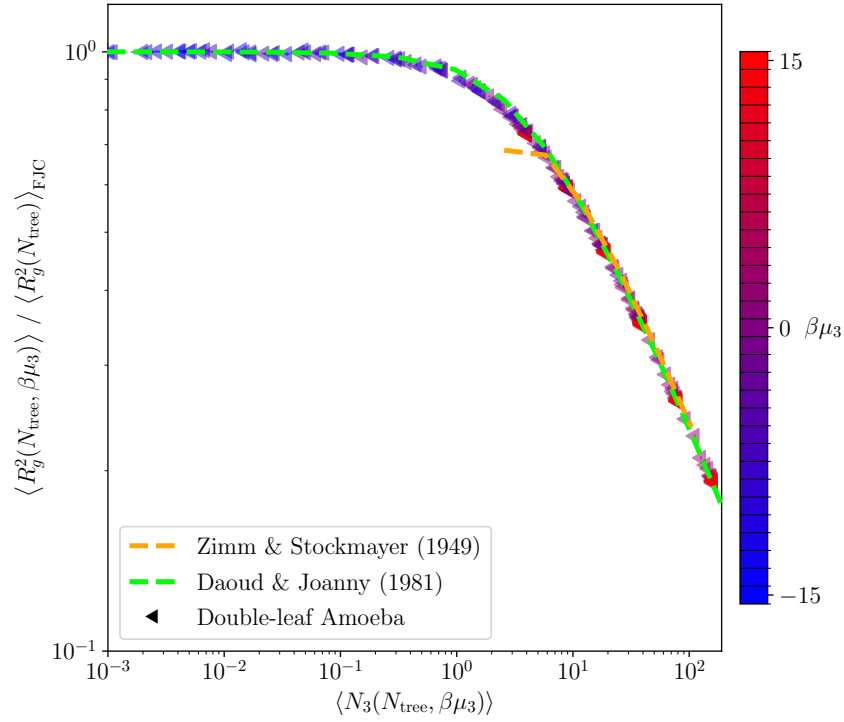


Figure 6.4: Mean-square radius of gyration of ideal trees, normalized by the freely-jointed-chain value, as a function of the mean number of branch points. Symbols are simulation results and the green dashed curve is the ideal-tree prediction of Daoud and Joanny [68]. The orange curve corresponds to the classical result of Zimm and Stockmayer [Eq. (3.44)] [6]. Different symbol colours correspond to different values of $\beta\mu_3$.

result in both the high- and low branching limit. The expectation value $\langle N_3 \rangle$ thus removes the chemical potential as an explicit parameter. The agreement shown here directly supports the relation between connectivity statistics and spatial size derived in Secs. 3.4.1 and 3.4.2.

6.2.2 Full crossover controlled by the branching activity

The branching activity λ derived in Secs. 3.4.2 and 3.5 provides a compact description from the almost-linear limit to the maximally branched regime. Figure 6.5 compares three independent routes: exact finite-size sums, direct Prüfer sampling, and the asymptotic analytical formulas.

The Prüfer data follow the exact finite-size evaluation, while the asymptotic expressions reproduce the expected crossover with increasing system size. This comparison validates both the direct sampler of Sec. 5.1 and the physical interpretation of λ as the extensive branch point density away from the extremely weak-branching limit.

6.3 Validation of elastic double-folded-ring statistics

We next test the ring partition function and stored-length statistics derived in Chapter 4. These tests use the local ring dynamics of Sec. 5.3, which evolves the explicitly represented double-folded ring rather than sampling its reduced tree representation.

6.3.1 Joint distribution of tree size and branching

For ring lengths

$$N_{\text{ring}} \in \{64, 216, 512, 1000\}$$

and branch point chemical potentials

$$\beta\tilde{\mu}_3 \in \{0, -2, -4, -6, -8\},$$

we generated samples of 200 independent double-folded-ring conformations for each parameter set. Equation (4.11) predicts the full joint probability $p(N_{\text{tree}}, N_3 \mid N_{\text{ring}}, \tilde{\mu}_3)$ after accounting for wrapping and stored-length multiplicities. Figure 6.6 overlays the simulated tuples on this predicted distribution.

The sampled points lie in the high-probability region of the theoretical distribution for all displayed parameter sets, with no systematic displacement from its maximum; in the original binning, the displayed samples fall in regions whose predicted probabilities are typically of order 10^{-2} or larger. Because this comparison overlays samples on the predicted density rather than fitting the complete empirical two-dimensional histogram, we use it as a visual consistency test of the location and concentration of the joint distribution. More quantitative tests are provided below by comparing ensemble averages and one-dimensional stored-length distributions.

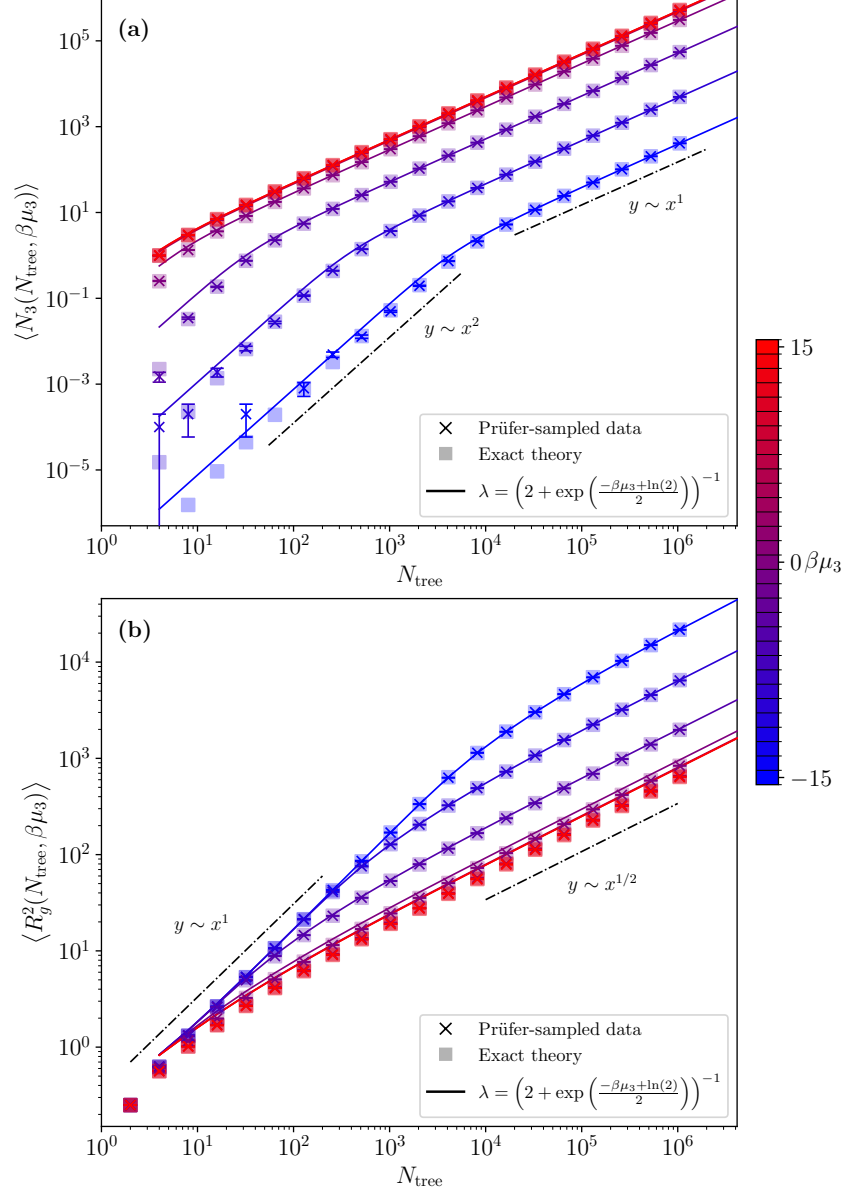


Figure 6.5: Mean number of branch nodes $\langle N_3 \rangle$ (a) and mean-square radius of gyration $\langle R_g^2 \rangle$ (b) as functions of N_{tree} . Symbols show exact numerical evaluation and independently Prüfer-sampled trees; lines show the asymptotic predictions based on Eq. (3.62) and the Daoud–Joanny form. Different colours correspond to different values of $\beta\mu_3$.

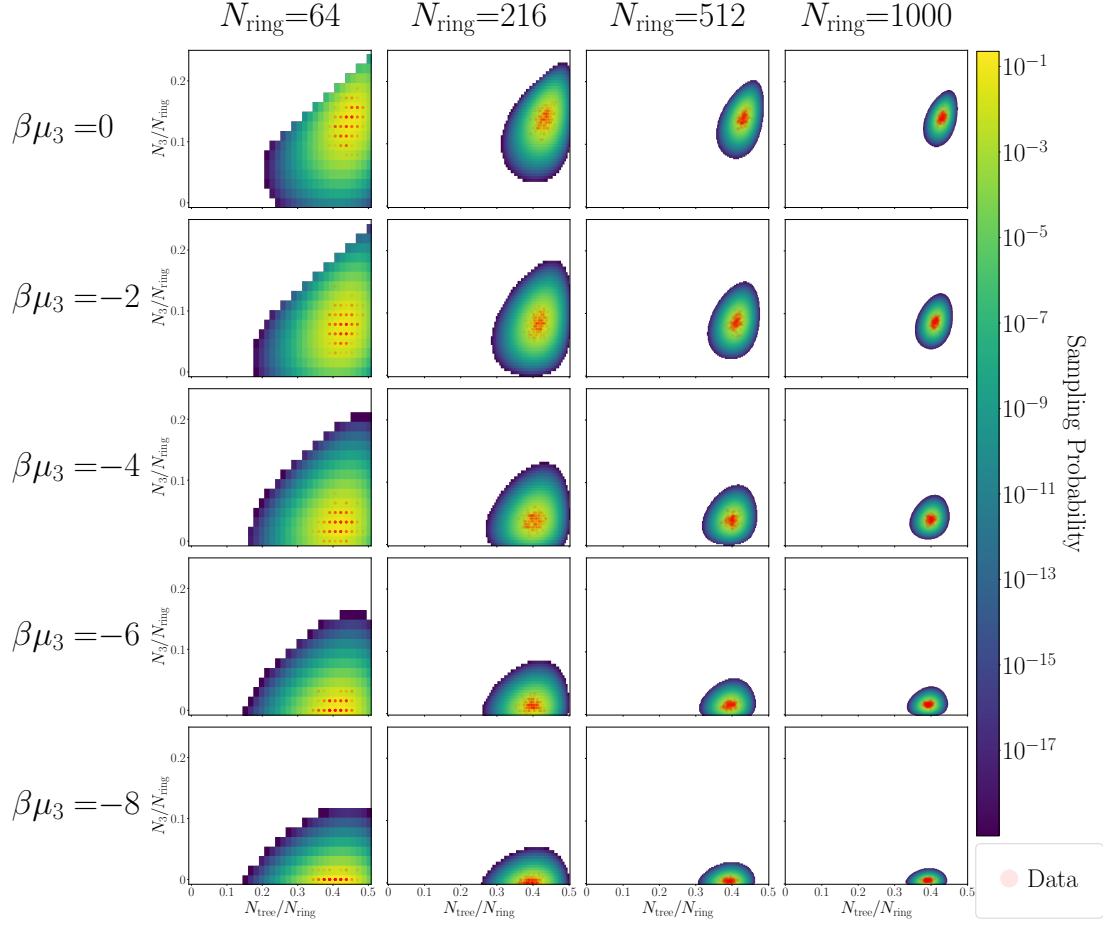


Figure 6.6: Predicted joint probability distributions $p(N_{\text{tree}}, N_3 \mid N_{\text{ring}}, \tilde{\mu}_3)$ from Eq. (4.11), shown as density maps for several ring lengths and branch point chemical potentials. Red points are tuples sampled by the local elastic-ring dynamics. White regions have probability below 10^{-19} . The sampled states track the high-probability regions without a systematic displacement of the peak.

6.3.2 Tree size and functionality averages

The elastic-ring theory also predicts mean tree size and functionality fractions. For $f_{\max} = 3$, Eqs. (4.32)–(4.34) give the large-system behaviour in terms of the ring branching activity λ . Figure 6.7 compares these asymptotic expressions with exact numerical sums of the finite-size partition function and with samples generated by the elastic lattice model.

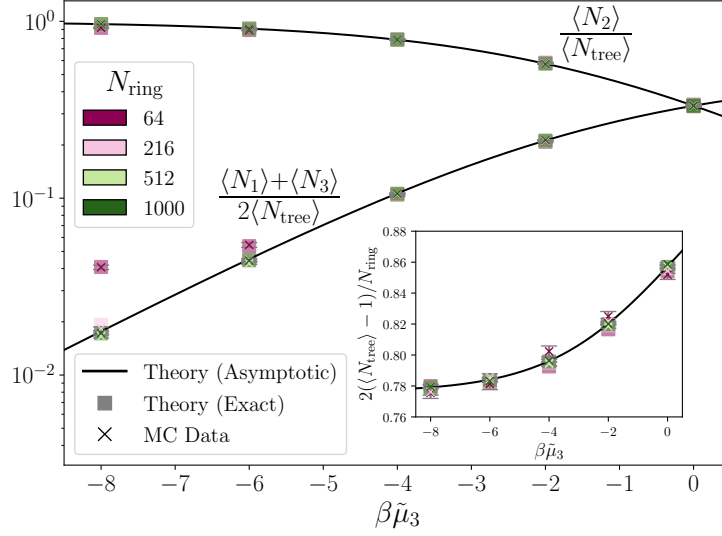


Figure 6.7: Functionality fractions and mean tree size in the elastic-ring ensemble as functions of $\beta\tilde{\mu}_3$. Crosses show sampled averages, square symbols show exact numerical sums based on Eqs. (4.14) and (4.15), and lines show the asymptotic predictions in Eqs. (4.32)–(4.34).

The exact sums and sampled averages agree over the explored range, while the asymptotic expressions capture the same trends once finite-size corrections become small. This comparison tests simultaneously the ring wrapping multiplicity, the repton factor, and the shift between tree and ring branching activities derived in Sec. 4.3.3.

The extension to arbitrary maximum functionality can be tested independently. Figure 6.8 compares the predicted numbers of nodes in each functionality class (see Sec. C.3) with samples generated by the elastic lattice model when the allowed $f_{\max}$ is varied and the functionality chemical potentials vanish.

The agreement supports the cancellation of the functionality-dependent wrapping factors embodied in Eq. (4.28); it also shows that the correspondence between labelled trees and tightly double-folded rings is not restricted to the $f \leq 3$ model used for most of the thesis.

6.3.3 Stored-length distribution

Another test comparing the data and our theory concerns how the N_{rep} units of stored length are distributed over nodes of different functionality. Section 4.4.2 derives the beta-binomial form in Eq. (4.36) and its functionality-averaged version in Eq. (4.37). We

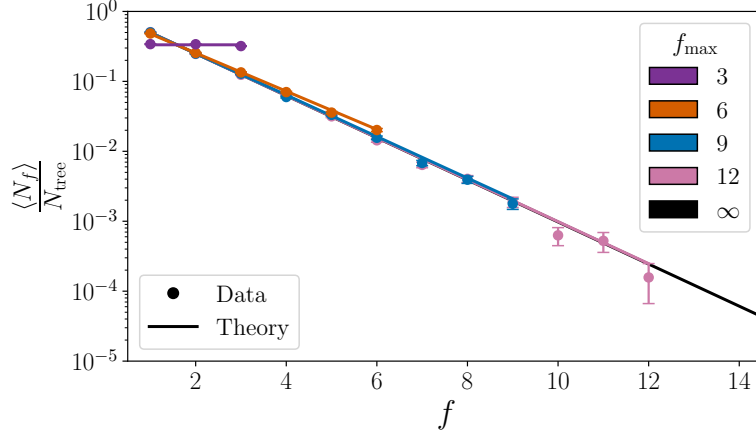


Figure 6.8: Mean functionality fractions $\langle N_f \rangle / N_{\text{tree}}$ for trees with different maximum functionalities f_{max} and vanishing functionality chemical potentials $\{\tilde{\mu}_f = 0\}$. Symbols are sampled averages of the elastic lattice model and lines are the theoretical predictions following from the arbitrary-functionality extension of Sec. 4.3.3.

evaluate those expressions using the asymptotic mean values of N_{tree} and N_f and compare them with data generated by the elastic lattice model for $N_{\text{ring}} \in \{64, 216, 512, 1000\}$ and $\beta\tilde{\mu}_3 \in \{0, -2, -4, -6\}$.

The beta-binomial form reproduces both the broadening of the distributions and their systematic dependence on node functionality.

For comparison, Fig. 6.10 shows the same data against the Poissonian distribution in Eq. (4.38) that was suggested in the supplementary material of [22]. In the latter case, the sampled distribution functions show a significant deviation from the theoretical description in the tails.

Therefore, we conclude that the data support the beta-binomial distribution consistent with distributing reptons randomly over the ring rather than a Poissonian distribution for distributing reptons randomly over the nodes.

6.4 Equivalence of tree-level and ring-level sampling

The final test of our theoretical considerations concerns the central correspondence of Chapter 4. The local ring algorithm evolves an explicit fixed-length ring, while the semi-grand-canonical Amoeba algorithm of Sec. 5.4 evolves a tree with fluctuating N_{tree} and assigns it the exact wrapping and repton weights needed to represent that ring ensemble. If the mapping is implemented consistently, both routes should produce the same equilibrium observables.

Figure 6.11 compares the two samplers for the mean number of branch points, mean tree size, and mean-square radius of gyration. The parameter sets span several $(N_{\text{ring}}, \beta\tilde{\mu}_3)$ combinations, specifically $N_{\text{ring}} \in \{64, 216, 512\}$ and $\beta\tilde{\mu}_3 \in [-6, 0]$. We implemented three interaction conditions: ideal, dilute self-avoiding, and melt systems. For the interacting

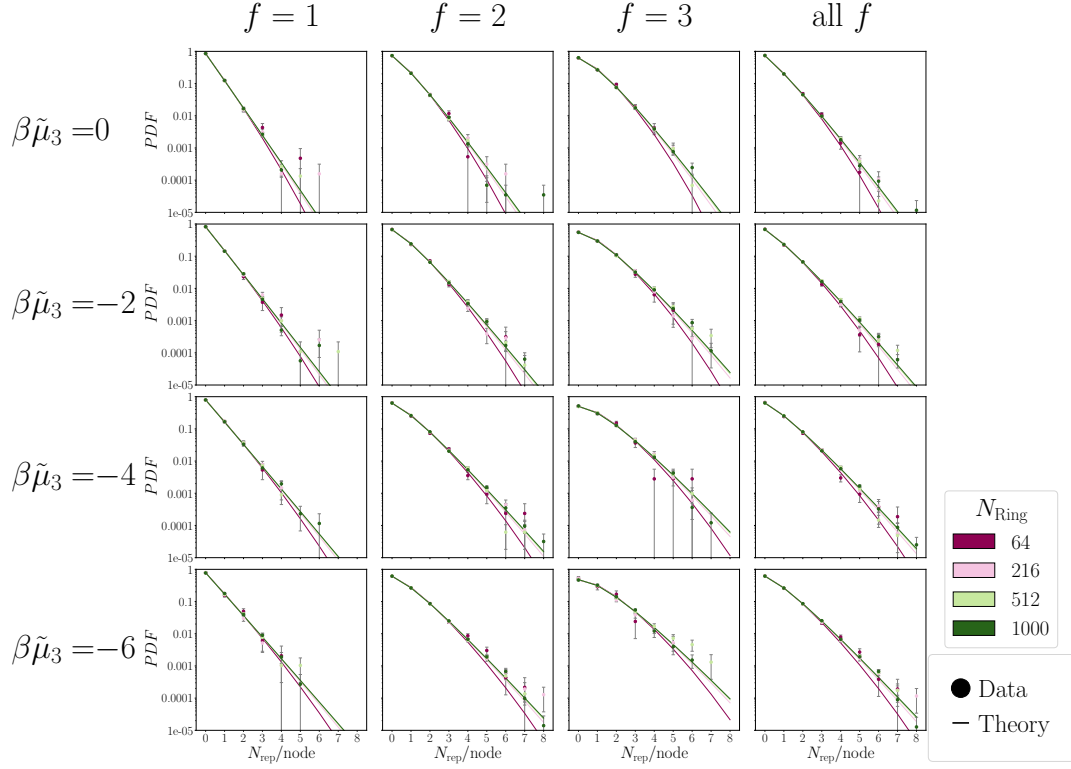


Figure 6.9: Probability distribution functions (PDFs) for the number of reptons on nodes of different functionality. Symbols are simulation data generated by the elastic lattice model, where colours represent different ring sizes. Lines are the beta-binomial predictions in Eqs. (4.36) and (4.37), evaluated with the asymptotic mean values from Eqs. (4.32)–(4.34).

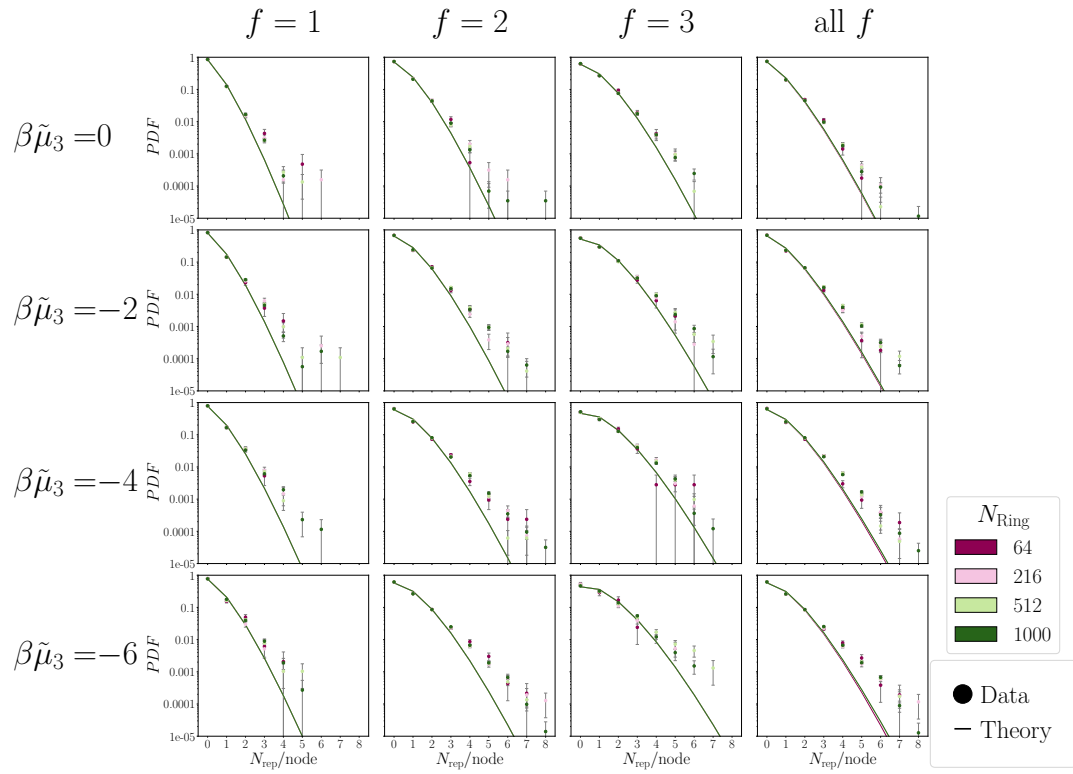


Figure 6.10: The same stored-length data as in Fig. 6.9, compared with the Poissonian distribution of Eq. (4.38). Its systematic deviations in the tails of the distributions are not present for the beta-binomial description.

cases, the interactions are those specified in Sec. 4.3.4 and are applied consistently at the tree level in both representations.

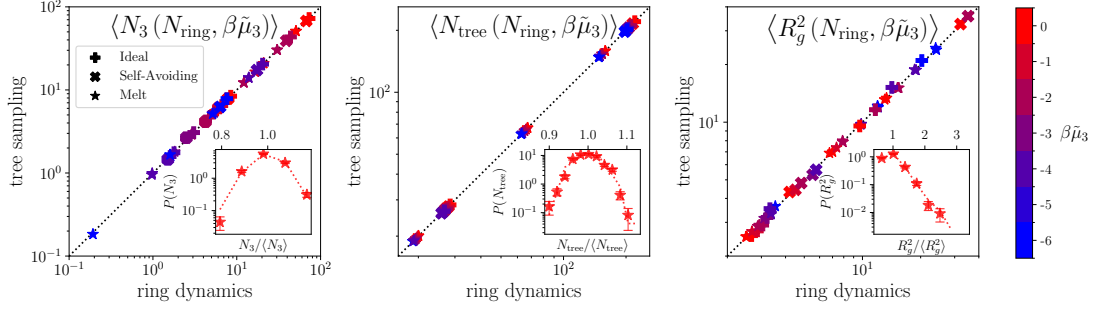


Figure 6.11: Equilibrium comparison between explicit ring dynamics (horizontal axis) and semi-grand-canonical tree sampling (vertical axis). The panels show $\langle N_3 \rangle$, $\langle N_{\text{tree}} \rangle$, and $\langle R_g^2 \rangle$ for ideal, dilute self-avoiding, and melt conditions over several values of N_{ring} and $\beta\tilde{\mu}_3$. The dashed diagonal is $y = x$. Insets compare the corresponding probability distributions for a ring melt with $N_{\text{ring}} = 512$ and $\beta\tilde{\mu}_3 = 0$.

The mean observables collapse onto the diagonal, and the representative inset distributions agree between the two sampling routes. This comparison demonstrates that two different microscopic representations—an explicitly double-folded contour and its weighted annealed-tree reduction—generate the same measured equilibrium statistics. The agreement in the self-avoiding and melt cases further shows that the correspondence survives the inclusion of spatial interactions when those interactions are defined at the common tree-node level.

6.5 Chapter summary

The validation tests presented here can be summarized as follows: First, the three fixed-size Amoeba implementations reproduce historical benchmark data and agree with one another regardless of the chemical potential chosen. Second, histogram reweighting recovers the exact N_3 -dependent configurational entropy of labelled trees from the data. Third, independent Prüfer samples and exact finite-size sums reproduce the structural consequences of the tree theory. Fourth, local ring dynamics agrees with the predicted elastic-ring statistics at the levels of the joint (N_{tree}, N_3) distribution, mean functionality fractions, and stored-length distributions. Finally, explicit ring dynamics and weighted tree sampling give the same equilibrium observables in ideal and interacting conditions.

These comparisons support the equilibrium correctness of the statistical weights and sampling constructions developed in Chapters 3–5. They do not imply that the algorithms have comparable relaxation times. The next chapter therefore turns from equilibrium correctness to the distinct questions of autocorrelation, equilibration, mass transport, and computational cost.

Chapter 7

Performance and equilibration dynamics of the sampling algorithms

Chapter 6 established that the algorithms introduced in Chapter 5 generate the intended equilibrium ensembles. Correct sampling, however, does not imply efficient sampling. A Markov chain may have the right stationary distribution while spending an impractically long time in correlated configurations, and two statistically equivalent representations of the same polymer may have radically different relaxation times because their elementary moves transport polymer mass on different length scales.

The performance results developed in this thesis fall naturally into three complementary parts. First, we study fixed-size Amoeba algorithms for ideal randomly branched polymers with controlled branching activity. This comparison reveals why the original leaf-moving algorithm becomes inefficient in both the weakly and strongly branched limits, how semi-kinetic and double-leaf proposals remove the corresponding local bottlenecks, and why a residual, algorithm-independent slowing down remains even after those bottlenecks are removed. Second, the exact Prüfer construction of Chapter 3 provides a qualitatively different benchmark: for ideal trees it bypasses Markov-chain relaxation altogether and generates independent configurations in linear time. Third, the tree-ring mapping of Chapter 4 allows us to compare local ring dynamics with non-local tree-level sampling for tightly double-folded rings, including melts. In that setting the choice of representation itself becomes an algorithmic acceleration mechanism.

In evaluating the efficiency of the algorithms, we separate three sources of computational cost: the probability that a proposed Monte Carlo move produces an effective connectivity change; the collective relaxation that remains after local move inefficiencies have been divided out; and the CPU cost of representing the same equilibrium ensemble at different coarse-graining levels. This separation leads to practical rules for choosing between direct Prüfer sampling, fixed-size Amoeba sampling, semi-grand-canonical tree sampling, and explicit ring dynamics.

7.1 Equilibration criteria and reference time scales

7.1.1 Centre-of-mass diffusion as an internal equilibration diagnostic

Leaf-moving algorithms equilibrate a tree by repeatedly pruning and regrowing leaves at different positions along the connectivity graph. A convenient way to monitor this process is through the mean-square displacement of the polymer centre of mass. For a tree of N_{tree} nodes,

$$\vec{x}_{\text{cm}}(t) = \frac{1}{N_{\text{tree}}} \sum_{i=1}^{N_{\text{tree}}} \vec{x}_i(t), \quad (7.1)$$

and we use the ensemble average

$$g_3(t) = \left\langle [\vec{x}_{\text{cm}}(t) - \vec{x}_{\text{cm}}(0)]^2 \right\rangle. \quad (7.2)$$

The same construction is used for the explicit ring representation, with the centre of mass calculated from the N_{ring} ring monomers.

A robust empirical equilibration criterion, used throughout the simulations underlying this thesis, is that the polymer has lost memory of its initial large-scale conformation once its centre of mass has diffused over a distance comparable to the polymer size,

$$g_3(\tau_{\text{eq}}) \simeq \langle R_g^2 \rangle. \quad (7.3)$$

After this crossover, the centre of mass is in its terminal diffusive regime and

$$g_3(t \gtrsim \tau_{\text{eq}}) \simeq \langle R_g^2 \rangle \frac{t}{\tau_{\text{eq}}}. \quad (7.4)$$

For a finite trajectory this suggests

$$\tau_{\text{eq}} \simeq \frac{\langle R_g^2 \rangle}{g_3(t)} t, \quad t \geq \tau_{\text{eq}} \quad (7.5)$$

which approaches the characteristic equilibration time once the measured point t lies safely in the terminal diffusive regime.

This diffusion-based test is preferable to judging equilibration from a single structural observable. Quantities such as $\langle N_3(t) \rangle$ or $\langle R_g^2(t) \rangle$ can approach their asymptotic values while the polymer still retains memory of its initial configuration. By contrast, the condition in Eq. (7.3) tests whether the entire object has moved over its own spatial size. Because mass transport in the algorithms discussed here is always small, this criterion typically requires roughly all bonds in a tree configuration to be pruned and reattached before equilibration is reached.

7.1.2 Naive reptation-like scaling for leaf movers

The connection between centre-of-mass diffusion and equilibration also provides a simple reference scaling for Amoeba algorithms. Consider one accepted leaf move in which a node is removed from a position i and reattached at a position j . The centre of mass changes by

$$\delta \vec{x}_{\text{cm}} = \frac{1}{N_{\text{tree}}} \delta \vec{x}_{ij}. \quad (7.6)$$

If the pruned leaf and its new position are sampled uniformly over the polymer, the characteristic squared displacement $|\delta \vec{x}_{ij}|^2$ is of the order of $\langle R_g^2 \rangle$. Hence one expects, per effective move,

$$g3|_{\text{effective move}} \sim \frac{\langle R_g^2 \rangle}{N_{\text{tree}}^2}. \quad (7.7)$$

Combining Eqs. (7.3) and (7.7) gives the reptation-like estimate

$$\tau_{\text{eq}} \sim N_{\text{tree}}^2 \tau_0, \quad (7.8)$$

where τ_0 is a microscopic time associated with producing an effective connectivity change. A feature of this estimate is that the explicit factor N_{tree}^2 is independent of the equilibrium branching structure. Much of the analysis below is devoted to understanding precisely how and why the measured relaxation departs from this naive expectation.

7.1.3 Monte Carlo steps, CPU time, and independent samples

Three notions of computational cost need to be kept distinct. For the fixed-size Amoeba algorithms, time is naturally measured in attempted Monte Carlo steps, as the computational work required for a single step is comparable between the proposed schemes. Differences between those schemes then appear through the number of attempted steps required to generate one effective connectivity change and through the collective relaxation that follows from many such changes. For comparisons between substantially different implementations, such as local ring dynamics and tree-level sampling, CPU seconds are more informative because the computational cost of one attempted move is not identical between algorithms. Finally, direct Prüfer sampling is not a Markov process at all: every generated ideal-tree configuration is independent, and the relevant performance metric is simply the CPU time required for a sample.

7.2 Fixed-size Amoeba algorithms with controlled branching

The equilibrium comparisons in Sec. 6.1 used the same data set for the performance analysis here: ideal FCC-lattice trees with fixed N_{tree} , functionality restricted to $f \leq 3$, and branching chemical potentials spanning the crossover from almost linear to maximally branched structures. More specifically, the performance study considered $N_{\text{tree}} \in \{10, 20, 40, 80, 81, 161, 321\}$ and $\beta\mu_3 \in [-15, 15]$. The adjacent sizes 80 and 81 make an

odd–even effect in the maximally branched regime directly visible. We showed in Sec. 6.1 that the original, semi-kinetic, and double-leaf Amoeba algorithms sample the same equilibrium distribution. Here we examine how their equilibration rates change as $\beta\mu_3$ is varied away from zero.

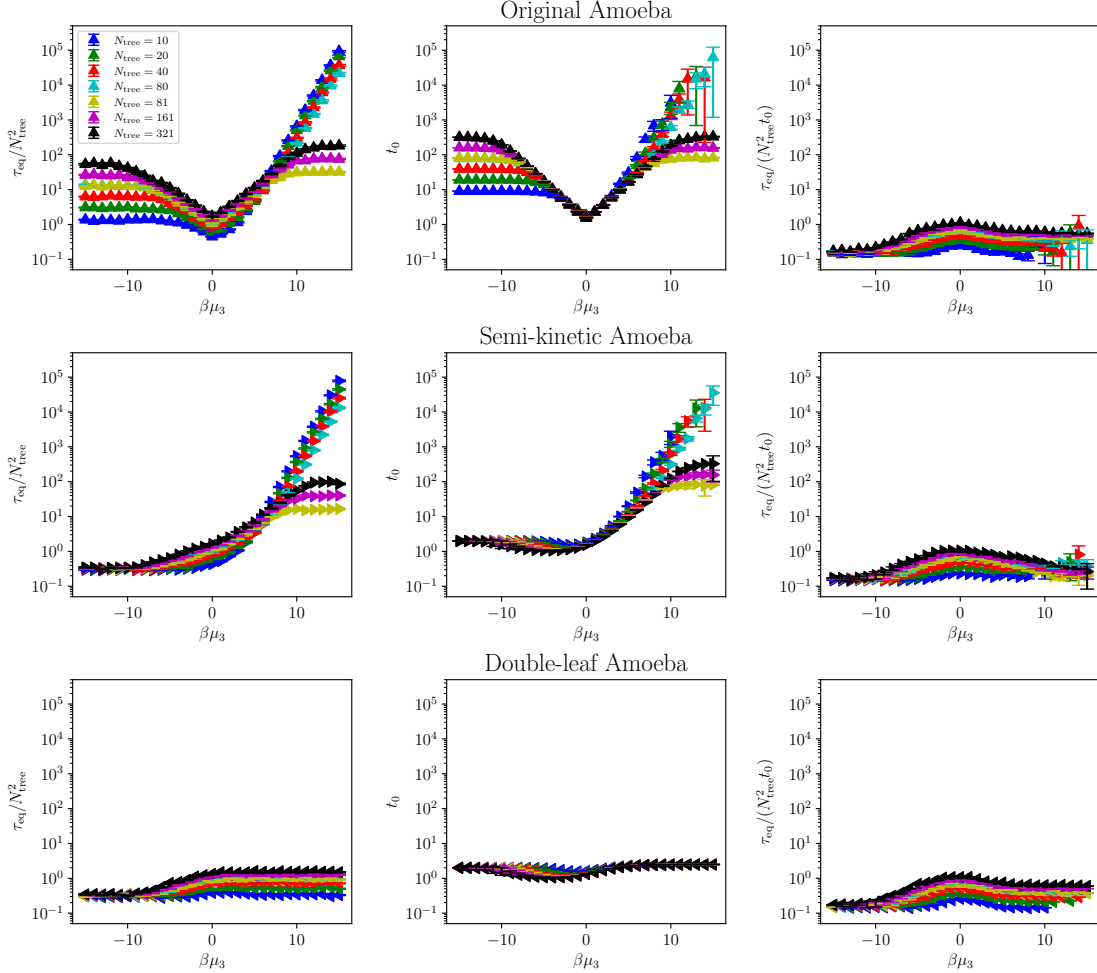


Figure 7.1: Performance of the original (top), semi-kinetic (middle), and double-leaf (bottom) Amoeba algorithms as a function of branching chemical potential. Left: the measured equilibration time, Eq. (7.5), divided by the naive N_{tree}^2 factor. Middle: the effective-move time t_0 estimated from the probabilities of effective connectivity-changing moves, Eq. (7.9). Right: the residual factor $\tau_{\text{eq}}/(N_{\text{tree}}^2 t_0)$. Different colours correspond to different tree sizes.

In the left panels of Figure 7.1 we observe that the original Amoeba algorithm is most efficient around $\beta\mu_3 = 0$, where branching is neither promoted nor suppressed. Moving away from this point in either direction causes a strong increase in the equilibration time. The semi-kinetic algorithm removes the severe slowdown in the weakly branched regime but does not, by itself, cure the highly branched limit. The double-leaf extension was designed precisely for that remaining bottleneck and produces an algorithm whose elementary connectivity-changing time is nearly independent of branching activity.

7.2.1 Weakly branched trees: rejection of side-branching moves

For $\beta\mu_3 \ll 0$, the equilibrium ensemble is dominated by essentially linear chains. In this limit a standard reptation move is effective: one removes an endpoint and places it at an end of the remaining chain. The original Amoeba algorithm, however, chooses its new attachment position without first accounting for the strong penalty against branch point creation. For a nearly linear tree, most nodes are two-functional and attaching the removed leaf to one of them would create a statistically disfavoured branch point. The latter is therefore accepted only with an exponentially small probability as the branching penalty grows.

For a strictly linear tree the original algorithm consequently rejects approximately a fraction $(N_{\text{tree}} - 2)/N_{\text{tree}}$ of the attempted attachments. Only an $\mathcal{O}(1/N_{\text{tree}})$ fraction of proposed moves produces a reptation-like end relocation. This adds an extra factor $\mathcal{O}(N_{\text{tree}})$ to the otherwise $\mathcal{O}(N_{\text{tree}}^2)$ relaxation, explaining the strong growth visible in the upper-left panel of Fig. 7.1.

The semi-kinetic construction avoids this by incorporating the branching weight already at the proposal stage. Branch-forming attachments are proposed only with the small frequency required by equilibrium, so proposed Monte Carlo steps are overwhelmingly useful reptation moves and acceptance probabilities are always of order unity. As a result the semi-kinetic and double-leaf algorithms recover the $\mathcal{O}(N_{\text{tree}}^2)$ reference scaling in the linear-chain limit. Double-leaf proposals bring essentially no additional advantage there because double leaves are themselves absent from a nearly linear connectivity.

7.2.2 Strongly branched trees: mobility of the $f = 2$ defect

The opposite limit is more subtle. Consider first a maximally branched tree without two-functional nodes. Removing a leaf necessarily converts its adjacent $f = 3$ node into an $f = 2$ node and therefore creates a transition state with a single defect. In the original Amoeba algorithm, reattaching the leaf to its original position restores the initial conformation. Reattaching it to a different leaf creates a second $f = 2$ node and reduces N_3 by one; for large positive $\beta\mu_3$ this final state has a strongly suppressed statistical weight and the move is rejected.

The semi-kinetic algorithm is more efficient at selecting statistically favoured attachments, but in this particular transition state that feature is not enough: the only strongly favoured node is the $f = 2$ site that was created by cutting the leaf and reattachment restores the original connectivity. This differs qualitatively from the linear-chain limit, where a cut chain possesses two statistically equivalent attachment choices and the connectivity of the polymers is changed in a finite fraction of the accepted moves.

The role of an existing $f = 2$ site becomes clearer for a maximally branched tree with one such defect. In the original algorithm, a detached leaf reaches this site only with probability of order $1/N_1$, while in the semi-kinetic algorithm the defect can be displaced much more efficiently. Nevertheless, displacement of the $f = 2$ defect alone is not sufficient

for fast equilibration. A branch point becomes mobile only after it is converted into a leaf, which for a peripheral $f = 3$ node requires removing its two leaves in succession. The probability that the second accepted single-leaf move removes the second leaf rather than merely displacing the defect remains of order $1/N_1$. The resulting probability for a connectivity rearrangement that mobilizes a branch point is therefore approximately $1/(2N_1) \sim 1/N_{\text{tree}}$ for both single-leaf algorithms.

This argument also explains the even-odd effect at large positive branching activity. For odd N_{tree} , the functionality constraints require at least one $f = 2$ node even in the maximally branched state. For even N_{tree} , no such node is required; creating the pair of $f = 2$ defects needed for a single-leaf rearrangement costs a free-energy penalty proportional to μ_3 , and the associated waiting time for an accepted and effective move grows exponentially with $\beta\mu_3$.

Double-leaf moves remove this bottleneck directly. A peripheral branch point carrying two leaves can be converted into a leaf in one move and transplanted elsewhere without relying on a pre-existing $f = 2$ defect. The double-leaf algorithm therefore remains effective throughout the strongly branched regime and has no analogous even-odd effect. For this reason, the double-leaf Amoeba algorithm is most robust across the complete range of branching activities.

7.3 Effective trial moves and the effective-move time t_0

The qualitative arguments above can be made quantitative by classifying Amoeba moves according to the functionality of the node from which a leaf is pruned and the functionality of the chosen node in the transition state. In addition to the total numbers N_1 and N_2 of one- and two-functional nodes, we denote by $N_{1,2}$ and $N_{1,3}$ the numbers of leaves attached to $f = 2$ and $f = 3$ nodes, respectively, and by N_{DL} the number of double leaves.

Table 7.1: Move classes used to estimate the rate of effective connectivity changes in the three fixed-size Amoeba algorithms. f_B is the functionality of the node from which the selected leaf is pruned and f_C the functionality of the node gaining the leaf, both evaluated in the transition state. An asterisk indicates $C = B$. The columns α_O , α_{SK} , and α_{DL} are proposal probabilities for the original, semi-kinetic, and double-leaf algorithms; the following three columns are the corresponding approximate acceptance probabilities. The last column assigns an effectiveness σ to the accepted move.

Class	$f(B)$	$f(C)$	α_O	α_{SK}	α_{DL}	acc_O	acc_{SK}	acc_{DL}	σ
I	1	1	$\frac{N_{1,2}}{N_1} \frac{N_1-1}{N_1+N_2-1}$	$\frac{N_{1,2}}{N_1} \frac{N_1-1}{N_1+(N_2-1)e^{\beta\mu_3}}$	$\frac{N_{1,2}}{N_1} \frac{N_1-1}{N_1+(N_2-1)e^{\beta\mu_3}} \frac{N_1}{N_1+N_{\text{DL}}}$	1	1	1	1
II	1	2	$\frac{N_{1,2}}{N_1} \frac{N_2-1}{N_2+N_1-1}$	$\frac{N_{1,2}}{N_1} \frac{(N_2-1)e^{\beta\mu_3}}{N_1+(N_2-1)e^{\beta\mu_3}}$	$\frac{N_{1,2}}{N_1} \frac{(N_2-1)e^{\beta\mu_3}}{N_1+(N_2-1)e^{\beta\mu_3}} \frac{N_1}{N_1+N_{\text{DL}}}$	$\min\left(1, \frac{N_1}{N_1+1} e^{\beta\mu_3}\right)$	$\frac{N_1}{N_1+1}$	$\frac{N_1+N_{\text{DL}}}{N_1+1+N_{\text{DL}}}$	1
III	2	1	$\frac{N_{1,3}}{N_1} \frac{N_1-1}{N_1+N_2}$	$\frac{N_{1,3}}{N_1} \frac{N_1-1}{N_1-1+(N_2+1)e^{\beta\mu_3}}$	$\frac{N_{1,3}}{N_1} \frac{N_1-1}{N_1-1+(N_2+1)e^{\beta\mu_3}} \frac{N_1}{N_1+N_{\text{DL}}}$	$\min\left(1, \frac{N_1}{N_1-1} e^{-\beta\mu_3}\right)$	1	1	1
IV	2	2	$\frac{N_{1,3}}{N_1} \frac{N_2}{N_1+N_2}$	$\frac{N_{1,3}}{N_1} \frac{N_2 e^{\beta\mu_3}}{N_1-1+(N_2+1)e^{\beta\mu_3}}$	$\frac{N_{1,3}}{N_1} \frac{N_2 e^{\beta\mu_3}}{N_1-1+(N_2+1)e^{\beta\mu_3}} \frac{N_1}{N_1+N_{\text{DL}}}$	1	1	1	0
V	1	1*	$\frac{N_{1,2}}{N_1} \frac{1}{N_1+N_2-1}$	$\frac{N_{1,2}}{N_1} \frac{1}{N_1+(N_2-1)e^{\beta\mu_3}}$	$\frac{N_{1,2}}{N_1} \frac{1}{N_1+(N_2-1)e^{\beta\mu_3}} \frac{N_1}{N_1+N_{\text{DL}}}$	1	1	1	0
VI	2	2*	$\frac{N_{1,3}}{N_1} \frac{1}{N_1+N_2}$	$\frac{N_{1,3}}{N_1} \frac{e^{\beta\mu_3}}{N_1-1+(N_2+1)e^{\beta\mu_3}}$	$\frac{N_{1,3}}{N_1} \frac{e^{\beta\mu_3}}{N_1-1+(N_2+1)e^{\beta\mu_3}} \frac{N_1}{N_1+N_{\text{DL}}}$	1	1	1	0
VII	1	1	—	—	$\frac{N_{\text{DL}}}{N_1+N_{\text{DL}}} \frac{N_1-2}{N_1-1}$	—	—	1	2
VIII	1	1*	—	—	$\frac{N_{\text{DL}}}{N_1+N_{\text{DL}}} \frac{1}{N_1-1}$	—	—	1	0

Classes I and V are the familiar reptation-like cases in which linear segments exchange

an endpoint or the detached leaf returns to its original attachment point. Classes II and III create or remove a branch point. Classes IV and VI move a leaf between a branch point and a two-functional site or return it to its previous position. Classes VII and VIII are the double-leaf counterparts, in which a branch point is either transported to a new leaf or restored to its original location.

Not every accepted move contributes equally to the equilibration of the tree. For this reason, we construct the parameter σ to categorize move classes based on their effectiveness. We assign $\sigma = 0$ to the identity moves V, VI, and VIII. Class IV also receives $\sigma = 0$ in this coarse estimate: although a leaf changes position, the move transports the $f = 2$ defect without mobilizing a branch point. Classes I–III have $\sigma = 1$, while a double-leaf transport in class VII is assigned $\sigma = 2$ because two leaves are moved in one accepted event. This simple bookkeeping converts the full proposal table into an estimate of the effective connectivity-change rate,

$$t_0^{-1} = \sum_{\nu=1}^{\text{VIII}} \alpha_{\nu} \text{acc}_{\nu} \sigma_{\nu} \Big|_{\langle N_1 \rangle, \langle N_2 \rangle, \langle N_{1,2} \rangle, \langle N_{1,3} \rangle, \langle N_{\text{DL}} \rangle}, \quad (7.9)$$

such that t_0 estimates the number of attempted Monte Carlo steps required to produce one effective change of the connectivity.

The middle column of Fig. 7.1 shows that this simple estimate accounts for most of the algorithm- and branching-dependent variation in the raw equilibration times. Across the full data set the estimates of t_0 span roughly five orders of magnitude. For the double-leaf algorithm, t_0 nevertheless remains of the order of only a few Monte Carlo steps and is nearly independent of branching activity. The remaining ratio $\tau_{\text{eq}}/(N_{\text{tree}}^2 t_0)$, shown in the right column, is much less algorithm dependent. That residual factor is the starting point for the universal relaxation analysis below.

7.4 A universal tree regime in Amoeba dynamics

Dividing the equilibration time by the effective time t_0 does not reduce the result exactly to the hypothesized N_{tree}^2 scaling behaviour of reptation algorithms. Instead, the ratios in the right column of Fig. 7.1 reveal a weaker but systematic dependence on the internal tree structure. To identify its origin, we examine the centre-of-mass motion before the terminal diffusive regime.

Figure 7.2 displays the double-leaf data in a form that divides out the simplest reptation prediction. At the earliest times most systems exhibit a regime in which

$$\frac{g_3(t)/\langle R_g^2 \rangle}{t/N_{\text{tree}}^2}$$

is approximately time independent. At late times the same quantity again approaches a constant, corresponding to terminal diffusion. Between those limits, however, branched

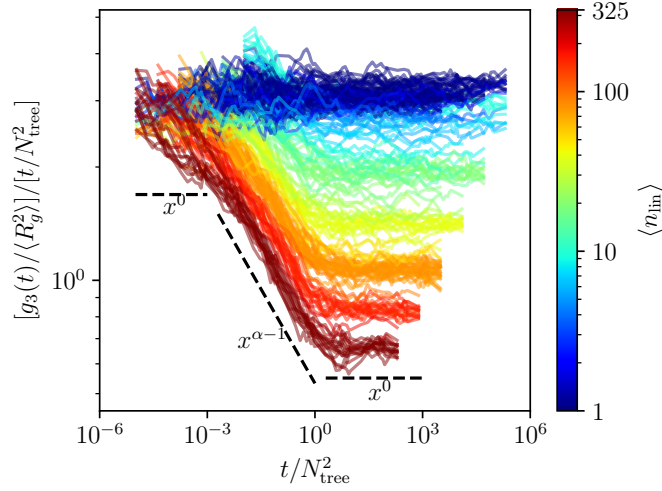


Figure 7.2: Centre-of-mass diffusion for the double-leaf Amoeba algorithm. The quantity $[g_3(t)/\langle R_g^2 \rangle]/[t/N_{\text{tree}}^2]$ is plotted against t/N_{tree}^2 and colour-coded by the mean number of linear segments, $\langle n_{\text{lin}} \rangle = 2\langle N_3 \rangle + 1$. Linear chains remain diffusive, whereas branched trees exhibit an extended intermediate subdiffusive regime.

trees display a broad anomalous regime with

$$g_3(t) \sim t^\alpha, \quad \alpha < 1. \quad (7.10)$$

The polymer centre of mass therefore diffuses more slowly on intermediate time scales than predicted by the simple uncorrelated-move argument.

The deviations are controlled by the number of linear segments in the tree,

$$\langle n_{\text{lin}} \rangle = 2\langle N_3 \rangle + 1. \quad (7.11)$$

A linear chain has $\langle n_{\text{lin}} \rangle = 1$ and shows no extended anomalous regime. As branching increases, the number of linear segments grows, the onset and duration of the subdiffusive regime become more prominent, and the asymptotic centre-of-mass diffusion is slowed.

A scaling form that captures these observations is

$$g_3(t) \simeq \langle R_g^2 \rangle \begin{cases} \frac{1}{N_{\text{tree}}^2} \frac{t}{\tau_0}, & t \lesssim \tau_1, \\ \langle n_{\text{lin}} \rangle^x \left(\frac{t}{\tau_1} \right)^\alpha, & \tau_1 \lesssim t \lesssim \tau_{\text{eq}}, \\ \frac{t}{\tau_{\text{eq}}}, & t \gtrsim \tau_{\text{eq}}, \end{cases} \quad (7.12)$$

with characteristic times

$$\tau_1 = N_{\text{tree}}^2 \langle n_{\text{lin}} \rangle^x \tau_0, \quad (7.13)$$

$$\tau_{\text{eq}} = N_{\text{tree}}^2 \langle n_{\text{lin}} \rangle^\Delta \tau_0. \quad (7.14)$$

Here τ_0 denotes the microscopic time of the scaling description. The move-class quantity t_0 in Eq. (7.9) is its directly calculable algorithm-specific proxy: it estimates how many attempted Monte Carlo steps are needed for one elementary effective connectivity change. The distinction is useful because t_0 can be computed from proposal and acceptance statistics, whereas τ_0 is the short-time scale entering the coarse-grained relaxation ansatz. In the comparisons below, rescaling by t_0 is used to remove the dominant algorithm-dependent microscopic slowdown.

The scale τ_1 separates an early regime dominated by mass exchange among the outer linear segments from an intermediate “tree” regime in which the destruction and creation of branch points controls the relaxation. Both times reduce to the standard reptation form $N_{\text{tree}}^2 \tau_0$ when $\langle n_{\text{lin}} \rangle \rightarrow 1$.

Continuity of Eq. (7.12) at the terminal crossover gives a relation between the three exponents,

$$\Delta = -\frac{1-\alpha}{\alpha} x. \quad (7.15)$$

The five-row comparison in Fig. 7.3 is useful because the three algorithms look very different before the short-time inefficiency is removed. Linear-chain data for the original Amoeba algorithm are shifted by the $\mathcal{O}(N_{\text{tree}})$ rejected branch-forming attempts described in Sec. 7.2.1. Highly branched data for the original and semi-kinetic algorithms are shifted by their difficulty in mobilizing branch points, with maximally branched even trees displaying extended plateaus while they wait for the rare $f = 2$ defects needed by single-leaf rearrangements.

Once time is expressed in units of t_0 , these algorithm-specific offsets largely disappear. This gives two separate conclusions. First, the dramatic performance differences between the three move sets arise primarily from short-time proposal and acceptance statistics. Second, after those differences are divided out, all three algorithms display the same collective tree relaxation. The anomalous regime is therefore not a peculiarity of the double-leaf proposal but an intrinsic feature of Amoeba-like pruning and regrowth dynamics.

Fits to the third row of Fig. 7.3 give

$$\Delta = 0.40 \pm 0.05 \quad \text{original Amoeba,} \quad (7.16)$$

$$\Delta = 0.39 \pm 0.05 \quad \text{semi-kinetic Amoeba,} \quad (7.17)$$

$$\Delta = 0.41 \pm 0.04 \quad \text{double-leaf Amoeba.} \quad (7.18)$$

Within the estimated statistical and systematic errors these values are indistinguishable. The crossover from subdiffusion to terminal diffusion gives similarly compatible estimates

$$\alpha = 0.85 \pm 0.01 \quad \text{original Amoeba,} \quad (7.19)$$

$$\alpha = 0.84 \pm 0.01 \quad \text{semi-kinetic Amoeba,} \quad (7.20)$$

$$\alpha = 0.84 \pm 0.01 \quad \text{double-leaf Amoeba.} \quad (7.21)$$

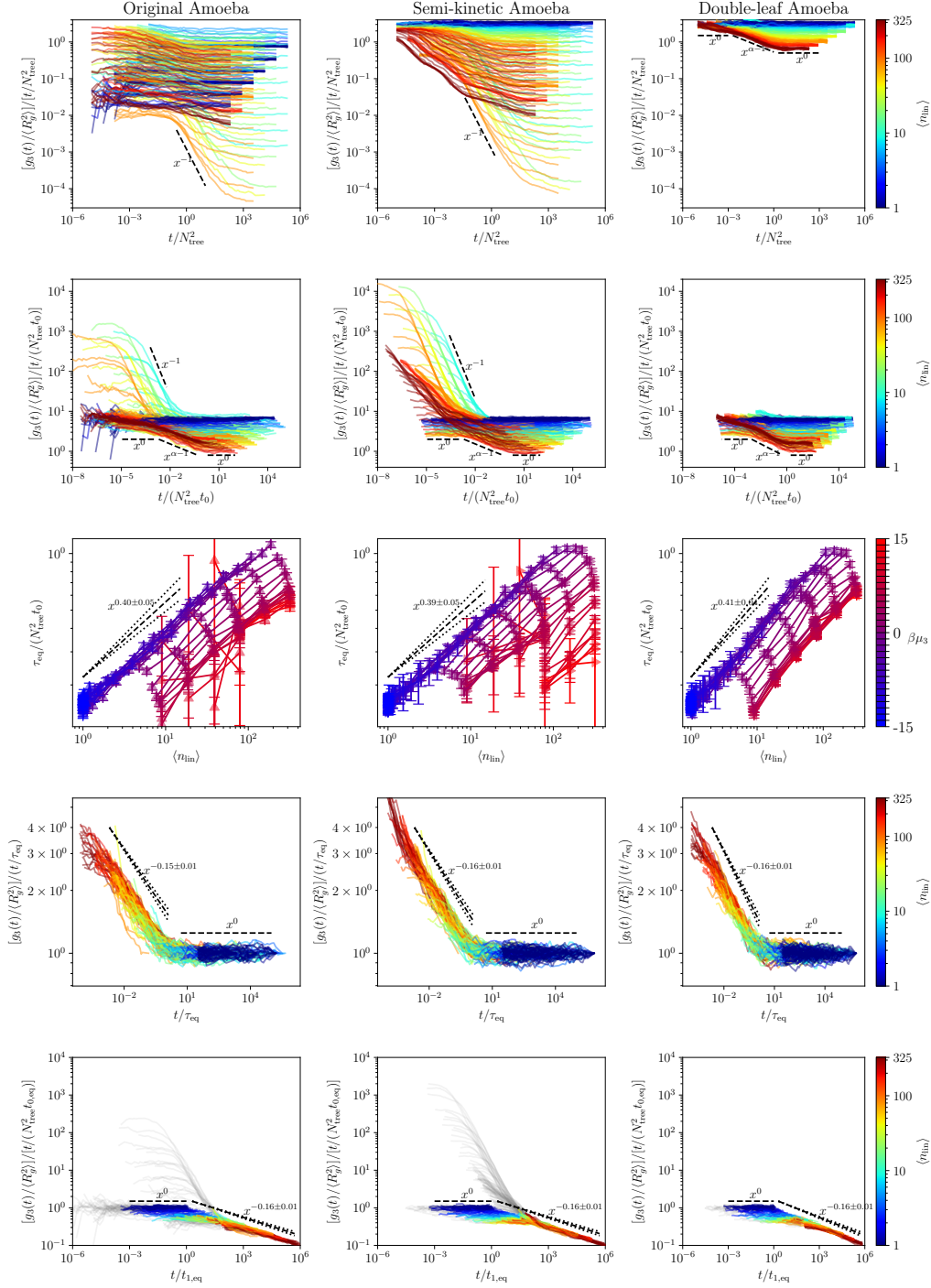


Figure 7.3: Scaling analysis of the centre-of-mass diffusion for the original (left column), semi-kinetic (middle column), and double-leaf (right column) Amoeba algorithms. Row 1 shows the raw diffusion after division by the naive N_{tree}^2 scaling. Row 2 additionally measures time in units of the algorithm-specific elementary connectivity-change time t_0 . Row 3 shows $\tau_{\text{eq}}/(N_{\text{tree}}^2 t_0)$ as a function of $\langle n_{\text{lin}} \rangle$. Row 4 aligns the crossover from the subdiffusive tree regime to terminal diffusion using t/τ_{eq} . Row 5 aligns the onset of the tree regime. The collapse after rescaling by t_0 demonstrates that the long-time relaxation is common to all three Amoeba move sets once their short-time proposal inefficiencies are removed.

Using Eq. (7.15) then yields the common estimate

$$x = -2.2 \pm 0.2. \quad (7.22)$$

The data are therefore summarized, to the accuracy relevant here, by

$$\boxed{\tau_{\text{eq}} \sim N_{\text{tree}}^2 \langle n_{\text{lin}} \rangle^{0.4} \tau_0} \quad (7.23)$$

with an intermediate onset time close to

$$\frac{\tau_1}{\tau_0} \sim \frac{N_{\text{tree}}^2}{\langle n_{\text{lin}} \rangle^2} \equiv l_{\text{lin}}^2, \quad (7.24)$$

where $l_{\text{lin}} = N_{\text{tree}}/\langle n_{\text{lin}} \rangle$ is the characteristic number of nodes per linear segment. For fixed branching chemical potential this length is asymptotically independent of the total tree size, up to finite-size corrections.

For comparison with the measured data one can invert Eq. (7.23) and the observed separation of time scales to define

$$t_{0,\text{eq}} = \frac{\tau_{\text{eq}}}{N_{\text{tree}}^2 \langle n_{\text{lin}} \rangle^{0.4}}, \quad (7.25)$$

$$t_{1,\text{eq}} = \frac{\tau_{\text{eq}}}{\langle n_{\text{lin}} \rangle^{2.4}}, \quad (7.26)$$

as estimators of τ_0 and τ_1 , respectively. The final row of Fig. 7.3 uses these estimates to align the onset of the anomalous regime. The collapse is less precise than at the terminal crossover, because t_0 itself is only a rough move-class estimate, but the expected trend is visible: trees with long linear segments exhibit an extended initial reptation-like regime, whereas highly branched trees cross almost immediately into the collective tree regime.

The exponent $\Delta \simeq 0.4$ remains phenomenological. The existence of the intermediate scale τ_1 shows that pruning and regrowing complete branches is more correlated than the naive independent-leaf picture underlying Eq. (7.8). One plausible interpretation is that, over intermediate times, the algorithm repeatedly draws mobile material from a restricted subset of peripheral nodes before the rearrangements penetrate to the central parts of the tree. This would make renewal of the deepest connectivity systematically harder than renewal of a linear chain. Regardless of the microscopic explanation, the collapse across all three algorithms indicates that a correction of the form in Eq. (7.23) should be expected generally for Amoeba-like leaf movers rather than only for one particular proposal scheme.

7.5 Direct Prüfer sampling bypasses equilibration

The analysis above concerns a Markov-chain problem: even an optimally proposed Amoeba move must be repeated many times before the tree loses memory of its previous connectivity. For ideal trees, the exact combinatorial representation of Chapter 3 changes the

problem qualitatively. A tree can be generated by independently drawing its functionality composition and a compatible Prüfer code, so no relaxation trajectory is required.

For $f_{\max} = 3$, once the probability distribution of N_3 has been precomputed, it can be sampled. For a single tree, the functionality assignment of its nodes, constructing a code and corresponding randomly branched polymer as in Table A.3, each require $\mathcal{O}(N_{\text{tree}})$ computational steps. This linear cost is asymptotically optimal because merely writing an N_{tree} -node connectivity requires $\mathcal{O}(N_{\text{tree}})$ information.

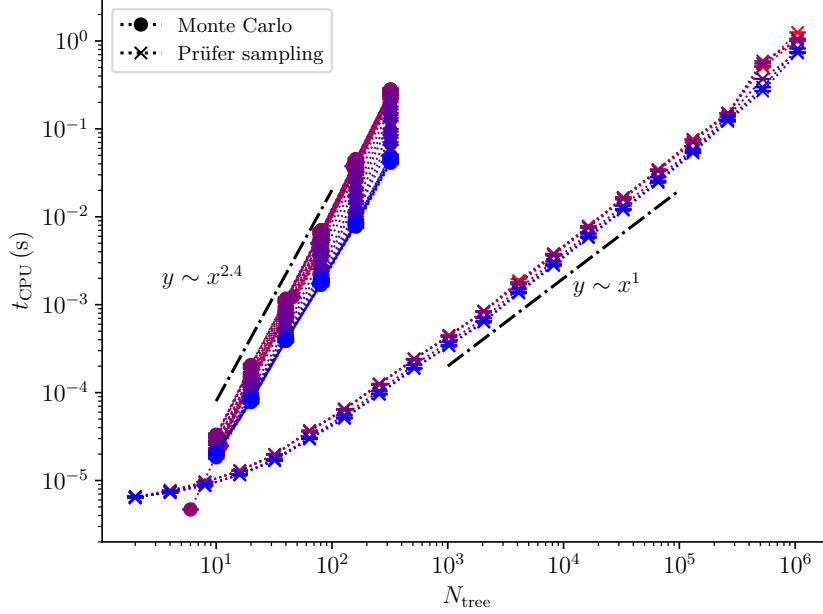


Figure 7.4: CPU time to obtain an independent ideal-tree configuration by direct Prüfer sampling (crosses) compared with the equilibration time of the optimized Amoeba Monte Carlo scheme (filled circles), as a function of tree size. The direct sampler follows approximately linear scaling, whereas the Markov-chain cost grows approximately as $N_{\text{tree}}^{2.4}$ over the measured range. Different colours correspond to different branching chemical potentials as in Fig. 6.1.

The difference is illustrated in Fig. 7.4. The optimized Amoeba data are consistent with a CPU cost close to $N_{\text{tree}}^{2.4}$ for producing decorrelated ideal-tree samples, in agreement with the universal relaxation correction discussed above. The direct Prüfer procedure remains approximately linear. In the implementation used for the corresponding study, an ideal tree of $N_{\text{tree}} = 2^{20} \simeq 10^6$ nodes could be generated in about one CPU second, whereas extrapolating the Monte Carlo timing gave a cost of order 10^8 s, or several years, for a decorrelated configuration of the same size.

The advantage of using Prüfer sampling has clear limits. The method relies on the separation between connectivity statistics and spatial embedding of the ideal ensemble. Once excluded-volume or other position-dependent interactions couple connectivity to conformation, a randomly drawn graph cannot simply be embedded with independent bond orientations and accepted as an equilibrium interacting polymer. In addition, for arbitrary

functionality constraints the preprocessing required to construct the exact composition probabilities can be more expensive than the $f \leq 3$ case, even though the subsequent generation of each Prüfer code and connectivity remains linear. Thus Prüfer sampling is the preferred reference method for independent ideal-tree statics, while Amoeba methods remain necessary when connectivity and spatial interactions must equilibrate together.

7.6 Equilibration of tightly double-folded rings

The tree–ring correspondence developed in Chapter 4 creates an opportunity to obtain equilibrium statistics of tightly double-folded rings faster than with the local ring dynamics of Sec. 5.3. Where the local algorithm moves ring monomers locally, the semi-grand-canonical Amoeba sampler modifies the underlying tree by non-local insertion and deletion of nodes and then restores the ring degrees of freedom by random wrapping and placement of stored length. Chapter 6 established that these routes generate the same equilibrium observables when their statistical weights are matched. We now compare how rapidly they reach those ensembles.

7.6.1 Systems, initial states, and observables

The tree–ring equivalence was validated for ideal, self-avoiding, and melt systems in Chapter 6; the performance comparison here focuses on melts with $N_{\text{ring}} \in \{64, 216, 512, 4096, 32768\}$ and branching chemical potentials $\beta\tilde{\mu}_3$ in the interval $[-6, 0]$. The melts consisted of 12 polymers in periodic three-dimensional FCC boxes with $N_{\text{ring}}^{1/3} \times N_{\text{ring}}^{1/3} \times N_{\text{ring}}^{1/3}$ elementary cells. Two different initial conditions were used. In the first, every ring starts from an extremely compact dimer state whose underlying tree contains only two neighbouring nodes. In the second, the explicit ring dynamics is initialized from equilibrated wrapped trees generated by the semi-grand-canonical Amoeba sampler.

The time-dependent observables are the mean tree size $\langle N_{\text{tree}}(t) \rangle$, the mean number of branch nodes $\langle N_3(t) \rangle$, the ring mean-square gyration radius $\langle R_g^2(t) \rangle$, and the centre-of-mass mean-square displacement $g_3(t)$.

The same criterion as Eq. (7.3) is used to judge equilibration in the ring and tree representations.

7.6.2 Relaxation from compact configurations

Figure 7.5 illustrates the relaxation for an interacting ring melt at $N_{\text{ring}} = 512$ and $\beta\tilde{\mu}_3 = 0$. Several features are worth keeping separate. First, $\langle N_{\text{tree}} \rangle$, $\langle N_3 \rangle$, and $\langle R_g^2 \rangle$ become stationary on approximately the same scale on which $g_3(t)$ crosses into terminal diffusion. Second, the stationary values are the same for the ring and tree representations, as independently demonstrated in Chapter 6. Third, the approach to equilibrium need not be monotonic: a structural observable can overshoot or pass through an apparently stationary regime

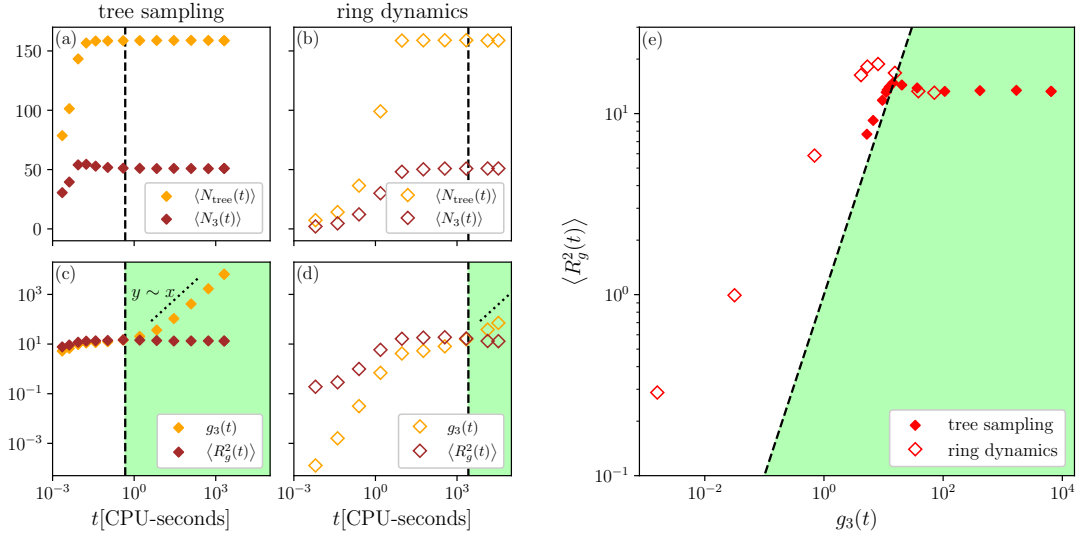


Figure 7.5: Representative equilibration of a melt of tightly double-folded rings with $N_{\text{ring}} = 512$ and $\beta\tilde{\mu}_3 = 0$, starting from compact dimer configurations. Filled symbols correspond to tree sampling and open symbols to local ring dynamics. The panels follow $\langle N_{\text{tree}}(t) \rangle$, $\langle N_3(t) \rangle$, $\langle R_g^2(t) \rangle$, and $g_3(t)$ as functions of CPU time. In the right panel, plotting $\langle R_g^2(t) \rangle$ against $g_3(t)$ makes the equilibration criterion visible: trajectories entering the region $g_3(t) \gtrsim \langle R_g^2(t) \rangle$ have a saturated mean squared gyration radius.

before the centre of mass has diffused over the polymer size. Fourth, the tree representation reaches the common equilibrium state orders of magnitude faster than the explicit local ring dynamics.

The last point can be heuristically explained by the nonlocality of leaf-mover algorithms with respect to local ring dynamics: A local ring move transports stored length or alters a local hairpin and therefore propagates information only gradually along the contour and through the underlying tree. By contrast, a tree-level cut or attachment can transfer connectivity and polymer mass between distant regions in a single accepted update. The corresponding ring mass would require $\mathcal{O}(N_{\text{tree}})$ local steps to arrive at the same point.

The comparison in Fig. 7.6 makes the difference in performance more explicit. Starting from compact dimers, the semi-grand-canonical Amoeba algorithm reaches equilibrium even for $N_{\text{ring}} = 32768$. Under the same criterion, the local ring dynamics reaches equilibrium from those compact states only for $N_{\text{ring}} \leq 512$ within the available simulation time. Already after the earliest recorded tree-sampling sweeps, $\langle R_g^2(t) \rangle$ has moved close to its final scale, whereas the local ring dynamics remains much closer to the compact starting state.

More importantly, the middle panel demonstrates a complementary use of the mapping between trees and rings. Rings initialized from equilibrated tree configurations show no drift under the subsequent local dynamics. The leaf-mover algorithm can therefore prepare equilibrated tightly double-folded rings that subsequently serve as initial states for the physically relevant local ring dynamics.

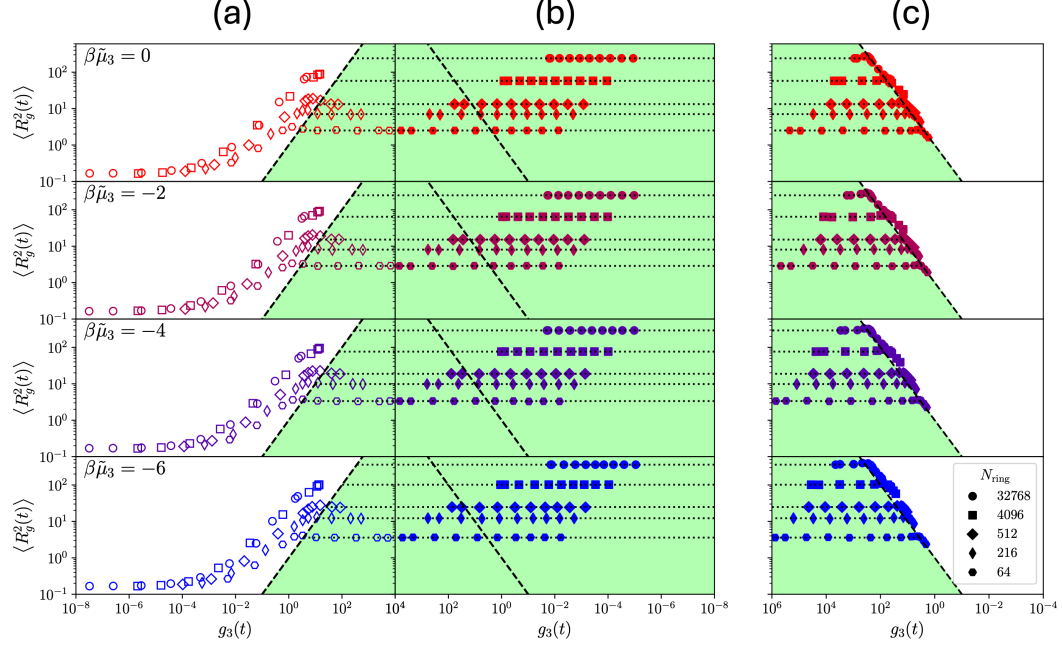


Figure 7.6: Time evolution of ring mean-square gyration radius ($\langle R_g^2(t) \rangle$), against mean-square displacement of the ring centre of mass $g_3(t)$. Panels (a,b) represent ring dynamics simulations, with the following initial conformations: (a) compact dimers (open symbols, time increases from left to right), (b) equilibrated rings generated using the Amoeba tree sampling (filled symbols, time increases from right to left). Panel (c) represents Amoeba tree sampling, with compact dimers for initial conformations. Symbols of different colours are for different $\beta \tilde{\mu}_3$ (values are in the figure), while different symbols are for different N_{ring} (see legend in the figure). Dashed lines correspond to $y = x$: crossing this line marks the regime $g_3(t) \gtrsim \langle R_g^2(t) \rangle$ where polymers reach equilibrium (see Fig. 7.5(e)). The green shaded areas in all panels correspond to the portions of the simulation trajectories from which data can be reliably sampled, because the corresponding ensembles are equilibrated. Notice in particular that the trajectories in panel (b) are entirely shaded green, owing to their prior equilibration using Amoeba tree sampling. This is also reflected in the stationary behaviour of $\langle R_g^2(t) \rangle$ in these plots. Note: because both algorithms only start writing data after one MC “sweep” (here, one MC sweep corresponds to the number of MC moves equal to the total number of monomers of the system), data for very short time scales are absent in Amoeba simulations.

7.6.3 CPU-time scaling

The CPU equilibration time is approximated as in Sec. 7.1.1. We used only the equilibrated portion of each trajectory for this estimate.

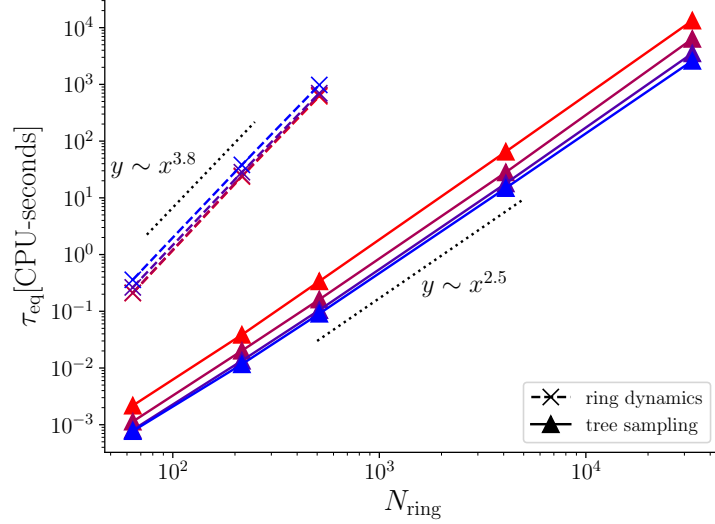


Figure 7.7: CPU equilibration time for melt simulations using explicit local ring dynamics and semi-grand-canonical Amoeba tree sampling. Over the explored range the tree sampler scales approximately as $\tau_{\text{eq}} \sim N_{\text{ring}}^{2.5}$, whereas local ring dynamics scales approximately as $N_{\text{ring}}^{3.8}$. The colour variation corresponds to different branching chemical potentials as in Fig. 6.11.

The result is shown in Fig. 7.7. Apart from a branching-dependent prefactor, the semi-grand-canonical tree sampler displays an empirical CPU scaling close to

$$\tau_{\text{eq}}^{\text{tree}} \sim N_{\text{ring}}^{2.5}, \quad (7.27)$$

whereas the explicit dynamic algorithm follows approximately

$$\tau_{\text{eq}}^{\text{ring}} \sim N_{\text{ring}}^{3.8}. \quad (7.28)$$

Thus the tree-level route is faster by at least an additional factor of order N_{ring} over the studied range. In practical terms, the tree representation allowed melts with $N_{\text{ring}} = 32768$ to be equilibrated where direct dynamic preparation had already become difficult around $N_{\text{ring}} = 512$.

7.7 Extensions, alternative move sets, and limitations

In the following, we discuss extensions and alternatives for the move choices of the Amoeba algorithms that were introduced in Chapter 5.

7.7.1 Alternative transition-state constructions

The transition state used for the semi-kinetic algorithm is not unique. One alternative that we implemented in early approaches was to reverse the order of operations: first attach an additional bond, creating a temporary $N_{\text{tree}} + 1$ state, and then remove an $f = 1$ node using a weighted rule. This proved inefficient in both extreme branching regimes. For nearly linear trees, the first attachment often creates a branch point that is immediately removed again, leaving the tree unchanged. For strongly branched trees there are more one-functional than two-functional targets, so the first attachment typically creates a linear segment that is then disproportionately likely to be removed in the second part of the same update.

A more ambitious possibility is a fully kinetic construction in which the cut is also statistically weighted and the intermediate state is skipped. This requires evaluating all combinations of cut and attachment classes in the initial and final trees. For $f \leq 3$ the bookkeeping already produces sixteen classes of moves. The additional data-structure updates increase the CPU cost per step, while the proposal normalization no longer cancels cleanly between the forward and reverse moves. If the proposals are weighted so as to absorb the local statistical factors, the acceptance probability retains a ratio of proposal partition sums of the schematic form

$$\text{acc}_{i \rightarrow f} = \min\left(1, \frac{z_{|i\rangle}}{z_{|f\rangle}}\right). \quad (7.29)$$

In the strongly branched regime this is particularly problematic because a final state containing a linear defect can still carry an exponentially small acceptance probability. The increased complexity therefore does not remove the essential bottleneck as cleanly as the explicit double-leaf proposal.

7.7.2 Higher functionality

The same philosophy of the generalized Amoeba algorithms presented here extends beyond $f_{\text{max}} = 3$. A semi-kinetic attachment rule can analogously be defined for a set of functionality-dependent chemical potentials $\{\mu_f\}$. The double-leaf update then generalizes naturally to multiple-leaf moves in which a branch point carrying several peripheral leaves is moved as a compound object. The principal cost is implementation complexity: more functionality classes require more dynamically maintained lists and more proposal classes. The anomalous scaling exponent $\Delta \simeq 0.4$ may also depend on the set $\{\mu_f\}$.

7.7.3 Spatial interactions

The ideal-tree study of the Amoeba algorithms initially left open how much of the acceleration would survive excluded-volume interactions. There are good reasons to expect some loss: a connectivity change that is statistically favourable at the graph level may be rejected because the proposed spatial embedding overlaps another node, and moving

two leaves at once can create more excluded-volume conflicts than moving one. The later tree–ring study provides a concrete answer for the related semi-grand-canonical problem. When excluded-volume interactions are evaluated consistently at the common tree-node level, the non-local tree representation remains dramatically more efficient than local ring dynamics for dense melts, while Chapter 6 confirms that the equilibrium ensembles remain equivalent. This does not constitute a complete scaling theory for interacting fixed-size double-leaf Amoeba dynamics, but it demonstrates that the central strategy—equilibrating static structure through non-local tree moves and switching representations only when needed—survives the introduction of strong spatial constraints.

7.8 Practical synthesis and chapter conclusion

The performance results suggest a simple hierarchy of sampling strategies.

For ideal static observables, direct combinatorial sampling using Prüfer codes is clearly the optimal approach. When interactions prevent such factorization, Amoeba moves provide a non-local Markov-chain alternative. Their short-time performance can be optimized by matching proposal frequencies to the equilibrium weights and by moving double-leaves when single-leaf updates encounter bottlenecks. After those local inefficiencies are removed, however, an intrinsic tree relaxation remains. The fixed-size data identify this collective contribution through the mean number of linear segments and the scaling $\tau_{\text{eq}} \sim N_{\text{tree}}^2 \langle n_{\text{lin}} \rangle^{0.4}$.

The tree–ring correspondence adds a final level to this hierarchy. Instead of optimizing a local ring algorithm, one can equilibrate the statistically equivalent coarse-grained tree with non-local moves and reconstruct the explicit ring afterwards. For static properties this reduces the accessible computational cost by more than one power of N_{ring} in the measured melt simulations. For dynamics, the same strategy provides equilibrated starting states for systems that local ring dynamics could not realistically prepare from compact initial conditions.

Chapter 8

Concluding Remarks

This thesis develops a statistical and computational framework for polymers with annealed branching connectivity, focusing on labelled trees and tightly double-folded rings. The ideal ensembles expose the combinatorial structure needed to separate connectivity entropy from conformational weights, while the interacting models test how far the resulting tree–ring correspondence survives beyond the ideal limit. The same exact statistical weights also guide efficient sampling algorithms. Together, the results connect the statistical mechanics of annealed branching to practical strategies for accelerating equilibrium simulations.

8.1 Main Findings

- In Chapter 3, we use Prüfer codes to derive exact finite-size configurational multiplicities and partition functions for ideal labelled randomly branched polymers with controlled branching activity. The resulting framework recovers and quantitatively connects historical results of Zimm and Stockmayer [5, 6], De Gennes [67], and Daoud and Joanny [68].
- The main finding of Chapter 4 is the one-to-one mapping between wrapping codes and tightly double-folded rings. This mapping allows switching between tree and ring ensembles, which is central to this thesis.
- Chapter 5 proposes Prüfer sampling as an optimal way to sample ideal randomly branched polymers, while we construct Amoeba algorithms to be efficient at arbitrary branching activity.
- We find in Chapter 6 that Amoeba algorithms sample labelled trees, whereas local dynamics of ring monomers on tree backbones sample wrapping codes. We show that the mapping between rings and trees carries over to ensembles with hard excluded volume as well.
- Finally, Chapter 7 establishes a hierarchy of algorithms based on their equilibration costs. For ideal monodisperse trees, direct Prüfer sampling generates independent configurations in $\mathcal{O}(N_{\text{tree}})$ work per tree. When excluded volume couples connectivity and spatial embedding, Amoeba algorithms remain applicable to trees and tightly

double-folded rings. After proposal-dependent short-time bottlenecks are removed, the Amoeba variants share an additional collective equilibration scaling with exponent $\Delta \simeq 0.4$.

8.2 Discussion

This thesis focuses on the statistics of randomly branched polymers and tightly double-folded rings, and much of the research has been performed using lattice models. Here we discuss the strengths and limitations of our models and how they could be used and extended to answer currently open questions.

8.2.1 Application to biological polymers

An important feature we used for relating the ensembles discussed here is the equivalence of each monomer or node in a polymer. As a consequence, double-folded rings are not sequence-specific and an averaged contact map over tightly double-folded ring structures would not reveal any specific structure. The biological polymers we aim to study, however, have a specified sequence of base pairs, and, as both secondary structures of RNA and contact maps in DNA reveal, their structures are not random. A possible shortcoming of our models is thus that no one-to-one mapping exists between a single biological molecule and a particular tightly double-folded ring. In other words, our models might appear too coarse-grained or too simple.

However, this simplicity is advantageous when we use the models here as a controlled reference ensemble. The parameter space is low-dimensional, consisting of a chemical potential and a tree size, or the combined parameter λN_{tree} (see Secs. 3.4 and 3.5). This makes it possible to establish the baseline statistics before adding sequence dependence or other biologically motivated features. In this sense, the double-folded ring is a null model against which more specific hypotheses can be tested.

For example, active processes and other biologically inspired out-of-equilibrium effects could be added to the tightly double-folded ring monomers to obtain a more realistic setup. In these scenarios it is unclear, though, how the exact tree–ring mapping that we derived in this thesis should be performed. A more modest approach would therefore be to start from a properly equilibrated ensemble and add the active processes only at the start of a simulation with local ring dynamics to compare their effects on the structure and organization of the polymers.

8.2.2 Application to a model of melts of rings

The tightly double-folded ring polymers in this thesis are always confined to a tree backbone. Section 2.4 discussed why this is a useful reference model for ring melts. In other such models, such as the decorated-loop [54] or fractal-loopy-globule [55] descriptions, the ring

can open away from a strictly double-folded backbone. Such openings can permit threadings and other relevant microscopic structures that are absent from the present model.

This limitation can be useful in comparison with other models, experiments and simulations: When systematic deviations are observed from the tightly double-folded ring model only in certain regimes, this can identify microscopic features such as threadings as the source of those deviations. An absence of arrested dynamics as in [59] at comparable scales for the double-folded ring model would hint at threadings as the underlying mechanism. A subsequent controlled inclusion of threadings, by relaxing the excluded volume constraints in Sec. 3.1.3 for example, would then test this behaviour more directly.

A more stringent issue concerns the nature of our mapping between rings and trees and how to carry over between the two ensembles when interactions are introduced. This was possible mainly by omitting interactions at the ring level and introducing them only at the tree level. Empirically [20, 22], this assumption has proven effective enough for comparison with ring statistics. However, with ring monomers as the relevant degrees of freedom, this assumption is not necessarily physical. Once interactions at the ring level are introduced, Eq. (4.19) is not sufficient to account for the degrees of freedom of the ring monomers, and the central mapping result between the tree and ring ensembles would break down. To obtain a similar, exact, mapping, the distributions for placing ring monomers on the tree nodes need to be determined first. For sufficiently simple interactions, this might be possible, and the repton distributions in Sec. 4.4.2 might offer a suitable starting point. For more complicated interactions, however, it is possible that only effective solutions remain.

Finally, most of this thesis restricts the functionality to $f \leq 3$. This considerably simplifies the discussion through Eqs. (3.6) and (3.7), but physically there is nothing that prevents higher-order branch points for generic rings. Additionally, the multiplicity associated with wrapping an f -functional node (Eq. (4.26)) cancels the corresponding labelling degeneracy in a Prüfer code, making higher-order branch nodes particularly relevant for tightly double-folded rings. Figure 6.8 shows that higher-order branch points indeed occur when the functionality restriction is relaxed, although their probability decreases rapidly with increasing functionality. Depending on the ensemble of interest, it might therefore be useful to adopt the algorithmic generalizations proposed in Secs. 5.1.3 and 7.7.2.

8.2.3 Labelled versus unlabelled trees

Section 6.1.3 showed that the Amoeba algorithms used here sample labelled trees rather than unlabelled trees. However, this does not imply that ensembles of branching polymers in experiments necessarily represent labelled trees as well.

The nature of the tree node labels does not have a direct physical interpretation. However, for the distributions that we computed (Eqs. (3.31), (3.42), (3.46)), we showed that the labelling factor always drops out, making the labelling more of a “gauge” than an actual observable. In the same way, it can be argued that physical nodes have an identity,

such as a position in space at a certain time. A thought experiment in which such nodes are combined into trees then requires prior labelling of the nodes, whose entropy should likely be accounted for.

On the other hand, geometry and topology play a significant role in biology, as certain interactions rely solely on the occurrence of very specific shapes. When trees are synthesized based on this shape specificity, it could be that an ensemble of unlabelled trees is more relevant to model the situation at hand.

Section 3.3 argued that the distinction between the two ensembles cannot be reduced to the overall labelling entropy: after normalization by the number of labelled linear chains, the labelled and unlabelled total multiplicities share the same $N_{\text{tree}}^{-5/2}$ prefactor. The different exponential growth constants can in practice be accounted for by a suitable set of chemical potentials $\{\mu_f\}$.

Whether nature samples labelled rather than unlabelled therefore remains an open question in many scenarios. Extreme cases, such as the linear-versus-star example in Sec. 3.3 and the fraction of linear chains in the ensemble in Sec. 6.1.4 might, however, be exceptions to this.

8.3 Outlook

The coherent framework for branching polymers developed in this thesis can be applied to probe several open questions. Moreover, the data generated for the comparison between different algorithms in Chapters 6 and 7 can be directly analysed and interpreted in the context of one or more of the following topics.

- The connection between double-folded rings and bacterial DNA explored in [21] should be investigated further. In the latter work, contact probabilities of sparsely branching double-folded rings could be mapped onto contact probabilities of bacterial DNA under the assumption that the small systems studied were representative. With the current algorithms, we are in a position to explore the statistics of entire bacterial genomes efficiently under the mapping proposed in the mentioned works.
- The dynamics in the equilibrated tightly double-folded ring melt should be studied in more detail. If a transition to a topological glass state is absent, comparison with molecular dynamics simulations would suggest that threadings can indeed play a major role in slowing down the ring dynamics. However, it should be noted that, with the efficiency of the current best available local ring dynamics algorithm, we can presumably test only small time or length scales.
- A better understanding of the observed exponent $\Delta = 0.4$ in Sec. 7.4 can be advantageous for several reasons: First, this might allow the Amoeba algorithms to be sped up even further. Second, the cutting and pasting of leaves could presumably be related to the local dynamics in Sec. 5.3, which could allow the current mapping of statistics to be extended to dynamics as well and enable the study of large time scales of tightly double-folded ring polymers more efficiently.

- An extension of this thesis would be to compare our ensembles with those obtained using different algorithms, such as the ones discussed in Sec. 5.5. In particular, growth algorithms might offer another avenue for efficiently sampling melts of trees.
- More generally, the concepts in this thesis should inspire a broader look for analogous exact correspondences between polymer configurations and combinatorial objects. In the present thesis, the graph-theoretical description of ideal ensembles made both exact counting and efficient sampling possible; similar correspondences may provide useful coarse-grained descriptions beyond trees and tightly double-folded rings.

The algorithms used for sampling ideal monodisperse trees are openly available [86], as are the analysed simulation results [87]. The implementation of the semi-grand-canonical Amoeba algorithm used to efficiently equilibrate double-folded elastic rings is now openly archived in Ref. [88].

The framework developed here therefore provides both a controlled reference for branched and double-folded polymer statistics and a practical starting point for more detailed biological and ring-polymer models.

Acknowledgements

In July 2022, I decided to cycle southwards from Amsterdam. I arrived in Trieste, not knowing that I would call this city my home only two months later. Now, four years later, I carry this experience with me as a reminder to myself that life might surprise you and that planning does not always make sense. I stuck to that idea right up until my submission deadline.

Much like a polymer in a melt, a person cannot be described without incorporating the effects of their surroundings. Leaving my own ideal description aside, I would therefore like to acknowledge and thank the following people.

A great part of my gratitude goes out to **Angelo Rosa**. From my very first failed exam in polymer physics to my defence, you have always been supportive and given me confidence in my own abilities. In the first year of the PhD, I chose to do a project with you, mainly because I wanted to do “analytical stuff”, but I also felt that with you I would have some freedom to explore questions of my own. I have not been disappointed in either regard, but along the way I found out that your support extended beyond those aspects: you were always available to discuss research, you went with me on missions, and you were helpful in finding a postdoctoral position. These are all things I never asked for, but that you nevertheless did, and I really appreciate that! Perhaps one of your greatest qualities is your ability to navigate this crazy world of academia by establishing, maintaining, and constantly looking for new connections, and you have always encouraged me to do the same. Indeed, I have heard many anecdotes about you and I am happy that I now have a few of my own. I am sure this is not the end of it, as we have enough open projects to work on. Otherwise, you might pass me on the bike or while trail running in the mountains one day. And if none of that works out, I will certainly visit your vineyard that I sincerely hope you will have one day.

This thesis would not have been possible without **Ralf Everaers**. I should of course start by thanking you for hosting me in Lyon. In the past three years, I have seen the city in winter, spring, summer and autumn, and last time I had the pleasure of meeting your family and visiting your beautiful house in the countryside. My stays in Lyon and your visits to Trieste have been incredibly fruitful as they have always ignited new ideas and more often than not spawned a new draft (or two). The endless blackboard sessions and your persistence in fully understanding a problem yourself have forced me time and again to sharpen and clarify our ideas to the point where I am happy to defend them to

any audience. I am sure that this level of scrutiny will benefit me in whatever is to come beyond this thesis. But as rigorous as your physics is, your interests outside the office are equally broad. Over the past years, we have discussed politics, AI, cultural differences, bread and of course also some football; be assured that if Germany and the Netherlands ever both make it far enough in the tournament to play each other, you will hear from me. But before that, and before Italy participates again, I hope we can have another beer and discuss what else can be done with branching polymers.

A special thank-you goes to **Elham Ghobadpour**. My PhD project quite literally started where yours ended and your data and input on the papers have been pivotal for much of the research. Apart from that, you have always provided me with useful advice about working with a supervisor, navigating an academic career and organising and publishing scientific code. All lunches, dinners and drinks with you and **Ali Farnudi** added a lot to my Lyon experience and I am super grateful that you always had something organised. I couldn't have wished for a better collaboration!

I also thank **Ivan Junier** and **Gabin Laurent** for the good collaboration we have had over the years. **Ivan**, thanks a lot for introducing me to the supercoiling community every year and allowing me to present multiple times to this specialised audience. I cannot say I understand the biology now, but looking back, I really learnt a lot. **Gabin**, the ease with which you adopted my messy code—and you even found a bug!—saved me a lot of time. I am very curious what you guys will do with the code, and I look forward to the draft!

In the past year, I also had the pleasure of meeting **Sergei Nechaev** and starting a collaboration with you. Much of this thesis is built on the ideas that you sparked more than 40 years ago on the connection between tree- and ring polymers. Our current work on braids goes back to these ideas and I am very eager to see what these basic principles can still tell us. You visited us last year and I am very grateful for your hospitality during my visit to your lab in Orsay. On both occasions, I was very impressed by your knowledge and your ability to switch between seemingly distant topics. Thank you for our collaboration so far, and I am sure I can still learn a lot from you.

When I applied, I was lucky that the “SBP-” or “MSB-”, or whatever group had some extra scholarships available for me and my colleagues. Otherwise, I might never have got in and become part of the notorious curriculum with **Anna-Maria**, **Debarshi**, **Giuseppe**, **Matteo**, **Salvo** and **Samuel**. I can't say I have ever seen office 323 as a very productive place with you guys, but we certainly had fun. Whether it was **Matteo's** notes, **Giuseppe's** jokes, or **Debarshi's** codes, we all passed the exams and could leave **Mauro** and **Satvik** with a new bunch of biophysicists at the end of the year.

I would particularly like to thank **Matteo** for inviting me to your house for dinner in the very first week of the PhD and later for going swimming and running with me. I always enjoyed the talks we had during our runs. Thank you for taking me out and for being a true friend these past few years.

If there is anyone I should thank for taking me out, it is definitely you, **Samuel**. Coming to Trieste, you actually managed to buy a real Wilier, and boy, we had fun with

it. Whether it was the pussy-climb, lion-climb or tiger-climb, we always came out on top, not without a healthy dose of toxic competition. You always designed the routes, which included some accidental gravel roads, trails or non-existent roads; it was always a surprise. In the same way, you took me on an unforgettable cycling trip through Croatia, and on a stupidly dangerous bivouac-sleepover in your home region in Piedmont. Along the way you introduced me to many amazing people, and I hope there are still some memories to be made with **Sara**, **Matti**, **Davide D.** and **Ade**. During the first year, you were definitely my introduction to Italy, to Piedmont and to the people at SISSA. Whether it was half a kilo of pasta aglio e olio or a Testina Bolita, I could always come over for a bite or two. It's a pity that we can't jump on the bikes together anymore, but sometimes I still see you cycling here on the hills of the Karst. If you had cared, I would of course have dedicated this PhD thesis to you. But let's instead leave that for a future trip to the North Cape.

After the first year, I arrived in a new office with **Davide M.**, and considering that not much later we also shared the same house, I know that I tested your patience quite a bit. But you got through it all, and I still find it cool that we arrived at the same partition function for branched polymers. Through you, I also met **Giada**, who miraculously turned out to be a Dutch translator and a super friendly person. Soon, we will come to Lisbon to visit you guys for real!

In Lyon, I shared the office with a bit more seniority. I thank **Max** for interesting conversations and **Tatiana** for all the advice that you gave. Who knows, once I learn to ski properly, I might meet you on the slopes one day.

To my last office mates in Trieste I should first say: Sorry! Sorry for the mess that is my workplace, but don't worry, I will clean it all and finally put your names on the door! I am proud to be part of your vlogs, **Ashish**, and I hope you manage everything, from your family at home to the papers here and the annoying mission forms in Trieste. To **Valentina B.** I want to say thank you for all the small talk. We could both work super focused on our stuff, but venting our frustrations about the system was a very good way to lighten the mood a bit. It is a pity that I will leave our little in-office rebellion behind, but I am sure it is in good hands.

Outside the office, I am happy to have met a bunch of people from SISSA with whom I shared many good times. Of the PIs, I thank **Alessandro L.** for taking us to the mountains on several occasions. I thank **Giovanni** for being a proper coordinator, and for organising group meetings with desserts that I could often taste as well. Being a "+1" to that group, I often had lunch, coffees and some dinners with **Olivier**, **Salvo** and **Tomás**, with whom there was always something to discuss. It was mostly about food, but never without passion. **Salvo**, you were crucial in keeping me updated about SISSA politics when I had no clue what was happening. We also spent last summer at SISSA, writing our theses more or less together. I hope you had an *awesome* time as well!

In many regards, SISSA felt like a second home during my stay in Trieste. However, work cannot replace a real house and I have been lucky to live with quite a few special people, of whom I should first thank **Thomas**: Without you, I would have left the PhD

rather soon, because there was no roof over my head. It was cool to spend the first year together, and I am curious where you will end up. I would also like to thank **Giulio** and **Jaime** for all the good times. I hope you carry “pizza Monday” with you wherever you might go!

En tot zover in het Engels, want hoeveel mensen ik de afgelopen jaren ook heb ontmoet, minstens zoveel mensen heb ik vier jaar geleden plots achtergelaten in Nederland. Ik wil daarom eerst de families **Van der Hoek** en **Van Schaik** bedanken: hoe ver weg ik nu ook woon, jullie zijn en blijven voor mij familie. Ik kom nog altijd graag langs op de boerderij en ik vind het heel bijzonder dat zoveel van jullie gezamenlijk naar mijn verdediging willen komen. Voor **Henk** en **Dinie** wordt dit zelfs al de tweede keer!

Ik mag me ook gelukkig prijzen met een stel goede vrienden die nog altijd moeite doen om langs te komen. **Michiel** en **Werner**, het is niet normaal hoeveel kilometers jullie hebben gevreten om mij op te zoeken en een rondje te fietsen. Ondanks dat jullie zo ver weg wonen, hebben jullie misschien wel het beste beeld van mijn leventje hier. Door jullie voelt Amsterdam nog altijd een beetje als thuiskomen en ik hoop dat we snel weer een pijltje kunnen prikken, een pilsje kunnen drinken en nog even niet “verliezen van het leven” of van de weddenschap.

Ik wil ook graag **Anouk** en **Aranka** bedanken omdat jullie me zijn komen opzoeken en omdat ik in bij jullie kon slapen in Amsterdam. Samen in Lyon was goed eten, goed wijnen, maar vooral echt fijn gezelschap. Laten we ook in de toekomst altijd een beetje Skøllies blijven, waar we dan ook zijn.

Ook wil ik graag **Alex** bedanken voor de maandjes dat je hier was. Het was heel fijn om in je moerstaal te kunnen praten en ons samen te verwonderen over Italië in het algemeen en SISSA in het bijzonder.

Ik denk dat ik hier nooit zou zijn geweest zonder **Maarten** en **Louise**. Bij puur toeval hebben we vier jaar geleden samen met **Fred** en **Linda** Trieste verkend en ik weet nog goed hoe we speculeerden over een PhD precies hier en kijk me nu! Ik heb het altijd fijn met jullie, of het nou komt door het uitmuntende eten, door de eindeloze podcasts met Maarten of door de lichtelijke “insauzing” met wijn. Laten we na Venetië en Sardinië snel een nieuw paradijsje verkennen!

Lieve **Janneke**, jij bent ook ontzettend belangrijk geweest. Ik weet dat ik altijd open met je kan praten en dat we elkaar begrijpen zoals alleen broer en zus dat kunnen. Dat was niet alleen fijn in de afgelopen paar jaar, maar ook in alle jaren daarvoor. Bij jou en **Tijmen** kan ik altijd terecht en nu met **Merel** is het alleen maar gezelliger geworden. Ik kijk er ontzettend naar uit om mijn rol als “oom pizza” te vervullen!

Ik kan deze thesis niet verdedigen zonder stil te staan bij al jullie hulp, **papa** en **mama**. Of het nu was omdat ik in Nederland altijd bij jullie kon slapen, omdat jullie me vaak hebben opgezocht in Trieste en Lyon, of omdat jullie me financieel vaak een duwtje in de rug gaven, ik kon altijd op jullie rekenen. Maar eerlijk is eerlijk, jullie hebben eigenlijk nooit anders gedaan. Ik heb altijd mijn eigen weg kunnen gaan en jullie hebben er altijd

alles aan gedaan om mijn keuzes te ondersteunen. Hoe meer ik op mijn eigen benen ga staan, hoe dankbaarder ik daarvoor ben. Bedankt voor jullie onvoorwaardelijke vertrouwen in mij gedurende de afgelopen dertig jaar.

Finally, I would like to thank my *mori*, **Elisa**. When I hopped on my bike four years ago, I could not have imagined what we have now. We have a relationship that is supportive and surprising, and that I care about more than anything. We also have a dog now—and thank you, **Rudi**, for taking care of my sleep schedule—which only adds to the fun of being with you. You have made my life so much richer in many ways: you invited me to Sardinia several times, tried to teach me to ski and climb, took me on my first scuba dive, and together we explored at least three circles of hell in India. But you also taught me the secrets of making an incredible tomato sauce and now we are exploring the world of making bread and pizza.

Mi hai anche fatto conoscere un sacco di persone a cui voglio molto bene: la tua famiglia e specialmente **Francesca** con cui mi sento sempre benvenuto e che mi porta anche in vacanza in Sardegna, **Valentina P.**-CuCumiao! Non ho rubato niente, signora!-e **Simone**, perché sono un granchio. Poi ci sono i nostri amici di Trieste, **Morgi**, **Cisco** e **Maddi** con cui abbiamo vinto non solo THE MIND, ma anche THE GAME; gli amici di Roma, **Sara**, **Giachi**, **Marghi** ed **Ema** con cui ci sentiamo sempre a casa; e il tuo gruppo musicale, i leggendari Bassiconici, con **Alessandro B.**, **Davide M.**, **Luca** e **Vittorio**, con cui è sempre stato bello fare aperitivo durante (con gli altri “+1”, **Nika** e **Giada**) e dopo i concerti.

I would not have known any of these people like I do now if it wasn’t for you. You have been amazingly supportive not just over the past few months, but throughout our years together. Four years ago, I had no clue where I would be today. Now, I cannot wait to explore together whatever the future has in store for us.

Ti amo (di più).

Appendix A

Algorithm visualisations and generalizations

A.1 Constructing a Prüfer code from a connectivity graph

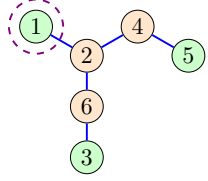
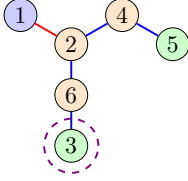
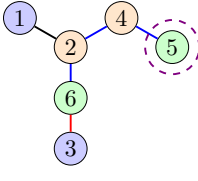
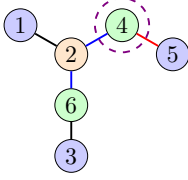
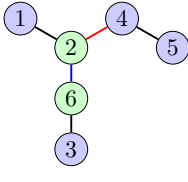
Encoding step	Partial Prüfer code	Procedure
	[]	For the given labelled tree, assign to every node i a residual degree d_i , defined as the number of incident bonds remaining in the current reduced tree. Initially, $d_i = f_i$. Nodes with $d_i = 1$ are leaves (green), whereas nodes with $d_i > 1$ are internal nodes (orange). Identify the leaf j with the smallest label ($j = 1$ here, encircled).
	[2]	Record the label i of the unique neighbour of j as the first entry of the Prüfer code ($i = 2$ here). Remove j and its incident bond from the reduced tree by setting $d_j = 0$, and decrease d_i by one. Then identify the remaining leaf with the smallest label ($j = 3$ here, encircled).
	[2,6]	Record the label of the unique neighbour of the current leaf as the next code entry ($i = 6$ here), remove the leaf and its incident bond, and update the residual degrees. In this step, d_6 decreases to one, so node 6 becomes a leaf in the reduced tree.
	[2,6,4]	Repeat the same procedure for the current smallest-labelled leaf.
	[2,6,4,2]	Stop after $N_{\text{tree}} - 2$ entries have been recorded. At that point, only two nodes with residual degree one remain (nodes 2 and 6 here), and their final connecting bond is not entered in the code. In the completed Prüfer code, the label of node i appears exactly $f_i - 1$ times.

Table A.1: Construction of a Prüfer code from a labelled tree. At each step, the leaf j with the smallest label is selected, the label i of its unique neighbour is appended to the code, and j and its incident bond are removed from the reduced tree. Green nodes are current leaves ($d_i = 1$), orange nodes have residual degree $d_i > 1$, and blue nodes have already been removed. The bond and code entry processed in the current step are shown in red; previously processed bonds are black and unprocessed bonds are blue. The procedure terminates after $N_{\text{tree}} - 2$ entries, when two leaves remain. In the completed code, label i occurs exactly $f_i - 1$ times.

A.2 Constructing a connectivity graph from a Prüfer code

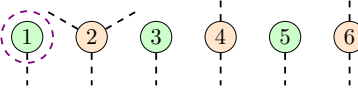
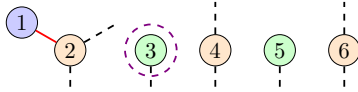
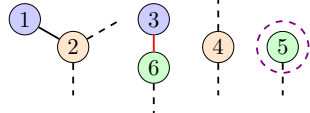
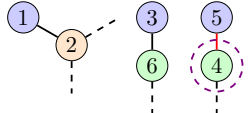
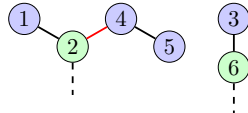
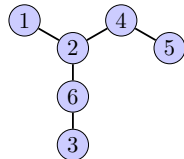
Decoding step	Prüfer code	Procedure
	[2,6,4,2]	For the code [2, 6, 4, 2], let n_i be the number of occurrences of label i in the code and initialize the <i>residual degree</i> as $d_i = n_i + 1$. Thus, labels absent from the code have $d_i = 1$ and are leaves (green), whereas nodes with $d_i > 1$ are internal nodes (orange). Identify the leaf j with the smallest label ($j = 1$ here, encircled).
	[2,6,4,2]	Connect the current leaf j to the node i specified by the first remaining entry of the code ($i = 2$ here). Remove that entry from the code, set $d_j = 0$, and decrease d_i by one. Then identify the remaining leaf with the smallest label ($j = 3$ here, encircled).
	[2,6,4,2]	Connect the current smallest-labelled leaf to the node specified by the next remaining code entry ($i = 6$ here), and update the residual degrees as before. Because d_6 decreases to one, node 6 becomes a leaf in the remaining construction.
	[2,6,4,2]	Repeat the same decoding step with the next remaining code entry.
	[2,6,4,2]	Continue until all $N_{\text{tree}} - 2$ entries of the Prüfer code have been read. At that point, exactly two unconnected nodes with residual degree one remain.
	[2,6,4,2]	Finally, connect the two remaining leaves to complete the unique labelled tree represented by the Prüfer code.

Table A.2: Decoding a Prüfer code to reconstruct its unique labelled connectivity graph. If n_i is the number of occurrences of label i in the code, the residual degrees are initialized as $d_i = n_i + 1$. At each step, the leaf j with the smallest label is connected to the node i specified by the first remaining code entry; that entry is then removed and the residual degrees are updated. Dashed bonds represent incident bonds that have not yet been assigned. The current bond and code entry are shown in red, completed bonds in black, current leaves in green, internal nodes in orange, and processed nodes in blue. After all $N_{\text{tree}} - 2$ entries have been read, the two remaining leaves are connected.

A.3 Sampling Prüfer codes

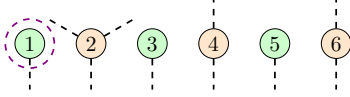
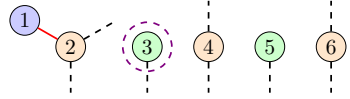
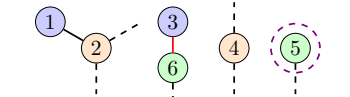
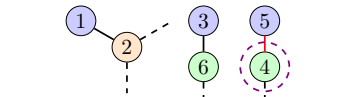
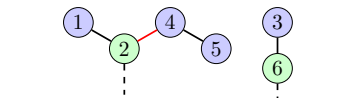
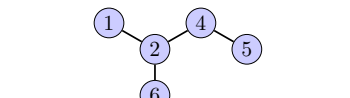
Sampling step	Partial Prüfer code	Procedure
	$[]$	Given a prescribed labelled functionality sequence $\{f_i\}$, define the remaining code multiplicity $m_i = f_i - 1$ and the residual degree $d_i = m_i + 1 = f_i$. Nodes with $d_i = 1$ are leaves (green), whereas nodes with $d_i > 1$ are internal nodes (orange). Identify the leaf j with the smallest label ($j = 1$ here, encircled).
	$[2]$	Draw the first code entry i from the labels with $m_i > 0$, with probability $m_i / \sum_{\ell} m_{\ell}$. Initially, a node with $f_i = 3$ therefore has twice the probability of being selected as a node with $f_i = 2$. Append i to the Prüfer code, connect the current leaf j to i , set $d_j = 0$, and decrease both m_i and d_i by one. Then identify the remaining leaf with the smallest label.
	$[2, 6]$	Repeat the weighted draw using the updated multiplicities m_i , append the selected label to the code, and connect the current smallest-labelled leaf to the selected node. If m_i becomes zero, then d_i becomes one and node i joins the set of leaves. Identify the new smallest-labelled leaf.
	$[2, 6, 4]$	Repeat the same sampling-and-decoding step using the updated multiplicities.
	$[2, 6, 4, 2]$	Continue until the code contains $N_{\text{tree}} - 2$ entries. All remaining multiplicities are then zero, and exactly two unconnected nodes with residual degree one remain.
	$[2, 6, 4, 2]$	Finally, connect the two remaining leaves. The resulting connectivity graph is a uniformly sampled labelled tree with the prescribed functionality sequence $\{f_i\}$.

Table A.3: Uniform sampling of labelled trees with a prescribed functionality sequence $\{f_i\}$ using Prüfer codes. Initially, label i has remaining code multiplicity $m_i = f_i - 1$. At each step, the next code entry is sampled with probability $m_i / \sum_{\ell} m_{\ell}$, appended to the code, and used to connect the current smallest-labelled leaf to node i ; the multiplicities and residual degrees are then updated. This procedure generates a uniformly random permutation of admissible Prüfer codes and therefore a uniformly sampled labelled tree with the specified functionality sequence. The colour convention is the same as in Table A.2.

A.4 Constructing a tree wrapping and its corresponding code

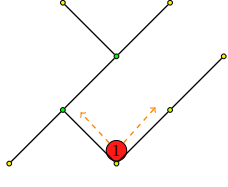
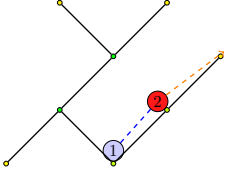
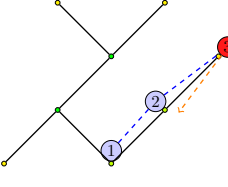
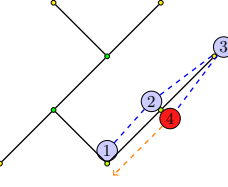
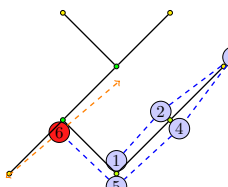
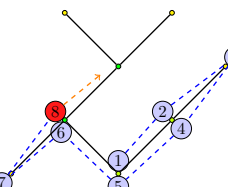
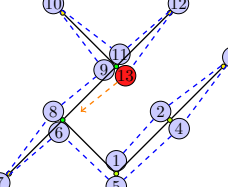
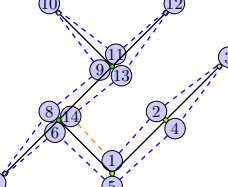
1. Begin by randomly placing a ring monomer on a node on the tree. Randomly choose one of the attached nodes, to decide the direction of wrapping.	2. When a linear node (a node with $f = 2$) is reached for the first time, place a ring monomer on it and proceed in the direction of unvisited tree nodes.	3. When a leaf is reached, place a ring monomer on it. Turn back and follow the connectivity of the tree in the “visited” direction.	4. When a linear node is visited for the second time, place a ring monomer on it, and proceed in the direction of the neighbor i that has not been visited f_i times.
 [2, ...]	 [2, 2, ...]	 [2, 2, 1, ...]	 [2, 2, 1, ...]
5. When a branch point (a node with $f = 3$) is reached for the first time, place a ring monomer on it and randomly choose one of the nodes' two unvisited bonds to continue.	6. When a branch point is reached for the second time, place a ring monomer on it, and proceed along the only remaining unvisited branch.	7. When a branch point is reached for the third time, place a ring monomer on it and continue along the direction of tree neighbor t that has not been visited f_t times.	8. The procedure concludes after N_{ring} ring monomers have been placed on the tree. In the last step, close the ring by connecting the first and last ring monomer.
 [2, 2, 1, 3, ...]	 [2, 2, 1, 3, 1, ...]	 [2, 2, 1, 3, 1, 3, 1, 1,]	 [2, 2, 1, 3, 1, 3, 1, 1, 1]

Table A.4: Rules to construct a tightly double-folded ring polymer (violet circles) wrapped around a tree and how to translate this into a corresponding wrapping code. At each step, the last placed ring monomer is coloured red. The possible directions of the wrapping procedure are indicated by the orange arrows. In every frame, the evolving construction of the wrapping code is shown in correspondence with the placed ring monomers. When a ring monomer is placed that does not correspond to a new entry in the wrapping code, this is indicated by the symbol “||”.

A.5 Decoding a wrapping code

1. Read the first label in the wrapping sequence. This is the functionality of the tree node on which the first ($i = 1$) ring monomer is placed.	2. For index t in the wrapping code, connect tree nodes with $f = f_t$ to tree nodes with $f = f_{t-1}$ and place ring monomers with increasing labels on each node t , up until $f_t = 1$.	3. After $f_t = 1$, follow the connectivity of the tree constructed so far, and place ring monomers on each encountered tree node, up until the first tree node is visited $f_{t=1}$ times (for $f_{t=1} > 1$), or up until a ring monomer is placed for the second time on a tree node t with $f_t = 3$.	4. For $f_{t=1} > 1$: After the first tree node is visited $f_{t=1}$ times, read the wrapping code from after the last encountered endpoint and repeat step 2, starting by connecting the new tree node, to the first placed tree node.
5. After a branch point (a node with $f = 3$) has been reached for the second time, read the wrapping code from after the last encountered endpoint and follow step 2, starting from a new tree bond connected to the relevant branch point.	6. Repeat Steps 2 - 5 up until one arrives at the N_{tree} -th entry of the wrapping code. At this step, the tree connectivity has been constructed.	7. After the tree connectivity has been constructed, follow the connectivity of the tree and place ring monomers on each tree node t that has been visited $f_t - 1$ times.	8. The procedure concludes after N_{ring} ring monomers have been placed on the tree. In the last step, close the ring by connecting the first and last ring monomer.

Table A.5: Reverse procedure to construct a tightly double-folded ring polymer from a given wrapping code. The colour convention is the same as in Table A.4.

A.6 Decoding invalid and valid wrapping codes

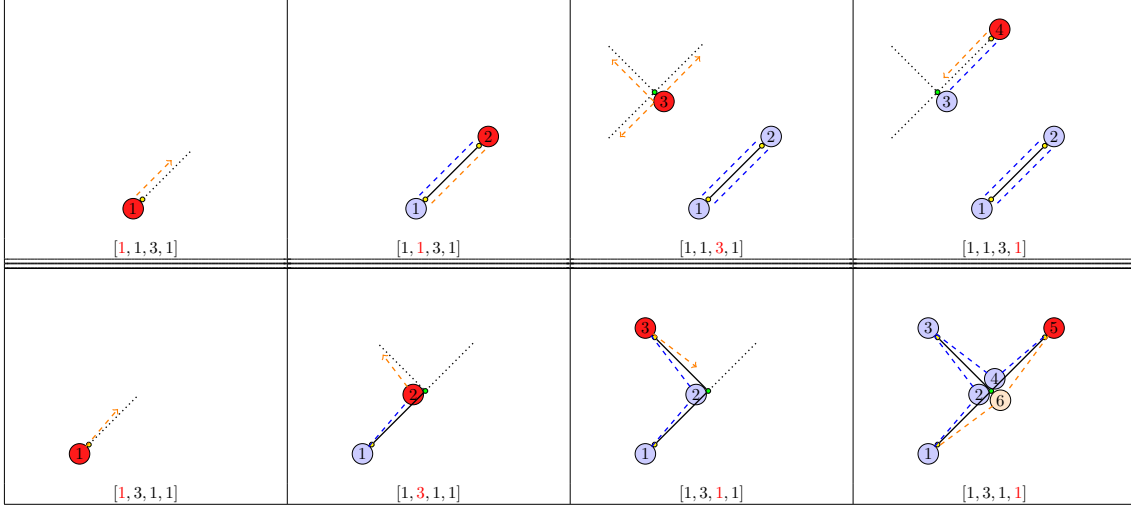


Table A.6: Decoding an invalid (top) wrapping code and a valid (bottom) wrapping code for the tree composed of $N_1 = 3$ $f = 1$ nodes and $N_3 = 1$ $f = 3$ nodes. Following the steps in Table A.5, the invalid wrapping code encodes a ring closure after just two steps, resulting in two disconnected tightly double-folded rings, of which only one is completed. The valid wrapping code closes the ring at the very last step, leading to a single tightly double-folded ring with $N_{\text{ring}} = 2N_{\text{tree}} - 2 = 6$ ring monomers.

A.7 Using recursions for sampling trees with arbitrary functionality constraints

For a prescribed labelled functionality sequence $\{f_i\}$, label i occurs exactly $f_i - 1$ times in the Prüfer code. Using Eq. (3.22), the statistical weight of the functionality sequence is therefore

$$p(\{f_i\} \mid N_{\text{tree}}, \{\mu_f\}) \propto \prod_{i=1}^{N_{\text{tree}}} \frac{\exp(\beta\mu_{f_i})}{(f_i - 1)!}, \quad (\text{A.1})$$

subject to

$$\sum_{i=1}^{N_{\text{tree}}} (f_i - 1) = N_{\text{tree}} - 2. \quad (\text{A.2})$$

The factor $(N_{\text{tree}} - 2)!$ omitted in Eq. (A.1) counts the permutations of the resulting Prüfer code and is independent of the functionality sequence. To sample Eq. (A.1) without explicitly enumerating all compositions, we define

$$F(n, N_{\text{code}}, \{\mu_f\}) = \sum_{\substack{f_1, \dots, f_n \text{ allowed} \\ \sum_{i=1}^n (f_i - 1) = N_{\text{code}}}} \prod_{i=1}^n \frac{\exp(\beta\mu_{f_i})}{(f_i - 1)!}. \quad (\text{A.3})$$

Here, n denotes the number of labels whose functionalities remain to be assigned, while N_{code} denotes the number of Prüfer-code entries that remain to be distributed among those labels.

For the complete tree,

$$\mathcal{Z}_{\text{tree}}(N_{\text{tree}}, \{\mu_f\}) = (N_{\text{tree}} - 2)! F(N_{\text{tree}}, N_{\text{tree}} - 2, \{\mu_f\}), \quad (\text{A.4})$$

where $\mathcal{Z}_{\text{tree}}$ denotes the configurational partition function of the labelled-tree ensemble.

Eq. (A.3) can be evaluated recursively. Consider assigning functionality f to one of the n remaining labels. This label occupies $f - 1$ positions in the Prüfer code and contributes a factor $\exp(\beta\mu_f)/(f - 1)!$ to the statistical weight. The remaining $n - 1$ labels must therefore occupy $N_{\text{code}} - f + 1$ code positions. Hence,

$$F(n, N_{\text{code}}, \{\mu_f\}) = \sum_{\substack{f \text{ allowed} \\ f-1 \leq N_{\text{code}}}} \frac{\exp(\beta\mu_f)}{(f - 1)!} F(n - 1, N_{\text{code}} - f + 1, \{\mu_f\}). \quad (\text{A.5})$$

The recursion relation immediately gives the conditional probability for assigning functionality f' to the next label,

$$p(f = f' \mid n, N_{\text{code}}, \{\mu_f\}) = \frac{\frac{\exp(\beta\mu_{f'})}{(f' - 1)!} F(n - 1, N_{\text{code}} - f' + 1, \{\mu_f\})}{F(n, N_{\text{code}}, \{\mu_f\})}, \quad (\text{A.6})$$

where f' is restricted to the allowed functionalities satisfying $f' - 1 \leq N_{\text{code}}$.

The recursion is initialized by

$$F(0, 0, \{\mu_f\}) = 1, \quad (\text{A.7})$$

$$F(0, N_{\text{code}}, \{\mu_f\}) = 0, \quad N_{\text{code}} > 0, \quad (\text{A.8})$$

$$F(n, N_{\text{code}}, \{\mu_f\}) = 0, \quad N_{\text{code}} < 0. \quad (\text{A.9})$$

States for which the remaining code length cannot be realized by the remaining labels may likewise be assigned zero weight.

For a tree of size N_{tree} , both n and N_{code} can span intervals of order N_{tree} during the recursion. The number of distinct recursion states is therefore bounded by $\mathcal{O}(N_{\text{tree}}^2)$. Evaluating each state requires a sum over at most f_{max} allowed functionalities, giving

$$\mathcal{O}(N_{\text{tree}}^2 f_{\text{max}}) \quad (\text{A.10})$$

operations for a direct implementation. For a fixed maximum functionality this reduces to $\mathcal{O}(N_{\text{tree}}^2)$, whereas $f_{\text{max}} = \mathcal{O}(N_{\text{tree}})$ gives the upper bound $\mathcal{O}(N_{\text{tree}}^3)$.

Once the table of F has been computed, a functionality sequence is generated through N_{tree} successive draws. For fixed f_{max} , this requires $\mathcal{O}(N_{\text{tree}})$ operations per sample, in addition to the linear-time generation and decoding of the Prüfer code.

Appendix B

Combinatorial result for wrapping codes

B.1 Bertrand's ballot theorem

Bertrand's ballot theorem has a long history and several classical proofs [69, 89, 90, 91, 92]. It concerns the ordering of votes in an election between two candidates. Let $V_A > V_B$ denote the final numbers of votes for candidates A and B . Among all orderings of these ballots, the fraction for which A is strictly ahead after every ballot is

$$\frac{V_A - V_B}{V_A + V_B}. \quad (\text{B.1})$$

The theorem is useful here because the viability condition for a wrapping code can be translated into an equivalent one-dimensional path constraint.

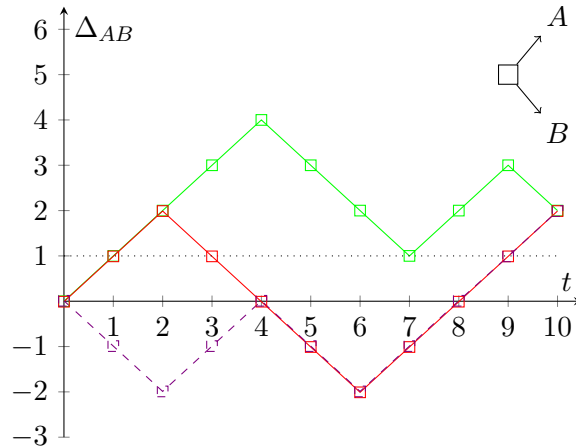


Figure B.1: Random-walk representation of ballot counting for $V_A = 6$ and $V_B = 4$. A vote for A is an upward step and a vote for B a downward step. The green path keeps A strictly ahead; the red path first ties at $t = 4$. Reflecting the red path about the horizontal axis up to its first return to zero produces the dashed violet path that starts with a vote for B .

Reflection-principle proof

We use the reflection construction associated with André's proof [90] and represent the ballot count as a one-dimensional walk [91], with

$$\Delta_{AB}(t) = V_A(t) - V_B(t),$$

where a vote for A produces a step $+1$ and a vote for B a step -1 . Every path begins at $(0, 0)$ and ends at $(V_A + V_B, V_A - V_B)$. A strictly valid path must start with an upward step and may never return to $\Delta_{AB} = 0$ thereafter.

The fraction of all ballot orderings that start with a vote for A is

$$\frac{V_A}{V_A + V_B}, \tag{B.2}$$

whereas the fraction that start with a vote for B is

$$\frac{V_B}{V_A + V_B}. \tag{B.3}$$

Consider a path that starts upward but later returns to zero for the first time at $t = p$. Reflecting the segment from $t = 0$ through $t = p$ maps it uniquely onto a path that starts downward and has the same final endpoint. Conversely, because $V_A > V_B$, every path that starts downward must cross zero before reaching the positive final endpoint and can be reflected back. The invalid start-up paths are therefore in one-to-one correspondence with all start-down paths. Subtracting their fraction from Eq. (B.2) gives

$$\frac{V_A}{V_A + V_B} - \frac{V_B}{V_A + V_B} = \frac{V_A - V_B}{V_A + V_B},$$

which proves Eq. (B.1).

Ties allowed

For its application to wrapping codes we need a version of the theorem in which candidate A is allowed to tie with B but may never fall behind. Let $V'_A \geq V'_B$ be the final vote counts in this problem. If we add one compulsory initial vote for A , we find $V_A = V'_A + 1$ and $V_B = V'_B$, which is exactly the situation discussed above. Dividing the strict-ballot fraction by the probability of the compulsory initial up-step gives

$$\frac{V'_A - V'_B + 1}{V'_A + 1}. \tag{B.4}$$

This is the fraction of orderings in which A is never behind B .

Blank votes and application to wrapping codes

Suppose that V'_C additional blank votes leave Δ_{AB} unchanged. Because horizontal steps can be inserted anywhere without changing whether the vertical path violates the boundary, the fraction in Eq. (B.4) is unchanged.

To apply this result, read a wrapping code backwards. The first entry is fixed to be a 1. Among the remaining entries there are $N_1 - 1 = N_3 + 1$ symbols 1, N_3 symbols 3, and N_2 symbols 2. A viable code is obtained exactly when the running number of 3's never exceeds the running number of 1's; the 2's act as blank votes. Thus

$$V'_A = N_3 + 1, \quad V'_B = N_3, \quad V'_C = N_2.$$

Equation (B.4) then gives

$$\frac{(N_3 + 1) - N_3 + 1}{(N_3 + 1) + 1} = \frac{2}{N_3 + 2},$$

which is precisely the viability fraction used in Eq. (4.6).

Appendix C

Asymptotic behaviour of tightly double-folded rings

C.1 Partition function for tightly double-folded rings, Z_{ring}

The partition function Z_{ring} in Eq. (4.9) is exact and can, in principle, be used to compute the expectation value of any observable $\langle \mathcal{A} \rangle$, provided one knows how $\mathcal{A}$ depends on the number of branch points N_3 and the tree size N_{tree} . However, for large N_{tree} , factorial terms become hard to interpret. To address this, we derive analytical expressions in the limit $N_{\text{tree}} \rightarrow \infty$, which are not only easier to evaluate but may also be more intuitive for applications to realistic physical scenarios.

The full conformational partition function $Z_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3)$ in Eq. (4.10) can be approximated using a scaling relation. To do so, we first take the continuum limit and replace the sum over N_3 by an integral:

$$\frac{Z_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3)}{c^{N_{\text{tree}}-1}} \approx 2 \int_0^{N_{3,\text{max}}} \frac{(N_{\text{tree}} - 1)!}{(N_3 + 2)! (N_{\text{tree}} - 2N_3 - 2)! N_3!} \exp[\beta \tilde{\mu}_3 N_3] dN_3, \quad (\text{C.1})$$

which, after introducing the scaled variable $x = (N_3 + 1)/N_{\text{tree}}$, becomes

$$\begin{aligned} \frac{Z_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3)}{c^{N_{\text{tree}}-1}} &\approx 2 \exp[-\beta \tilde{\mu}_3] \int_{\frac{1}{N_{\text{tree}}}}^{\frac{N_{3,\text{max}}+1}{N_{\text{tree}}}} dx \\ &\times \frac{N_{\text{tree}}! \exp[\beta \tilde{\mu}_3 x N_{\text{tree}}]}{\left[N_{\text{tree}} \left(x + \frac{1}{N_{\text{tree}}} \right) \right]! [N_{\text{tree}}(1 - 2x)]! \left[N_{\text{tree}} \left(x - \frac{1}{N_{\text{tree}}} \right) \right]!}. \end{aligned} \quad (\text{C.2})$$

Applying Stirling's approximation ($n! \sim \sqrt{2\pi n}(n/e)^n$) and subsequently neglecting terms

of order $\mathcal{O}(1/N_{\text{tree}})$, we obtain

$$\begin{aligned}
\frac{Z_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3)}{c^{N_{\text{tree}}-1}} &\approx \frac{\exp[-\beta\tilde{\mu}_3]}{\pi N_{\text{tree}}} \int_{\frac{1}{N_{\text{tree}}}}^{\frac{N_{3,\text{max}}+1}{N_{\text{tree}}}} dx \frac{\exp[\beta\tilde{\mu}_3 x N_{\text{tree}}]}{\sqrt{(1-2x) \left(x^2 - \frac{1}{N_{\text{tree}}^2}\right)}} \\
&\quad \times \frac{\left(\frac{1}{1-2x}\right)^{N_{\text{tree}}} [N_{\text{tree}}(1-2x)]^{2xN_{\text{tree}}}}{\left[N_{\text{tree}} \left(x + \frac{1}{N_{\text{tree}}}\right)\right]^{(x+\frac{1}{N_{\text{tree}}})N_{\text{tree}}} \left[N_{\text{tree}} \left(x - \frac{1}{N_{\text{tree}}}\right)\right]^{(x-\frac{1}{N_{\text{tree}}})N_{\text{tree}}}} \\
&\approx \frac{\exp[-\beta\tilde{\mu}_3]}{\pi N_{\text{tree}}} \int_{\frac{1}{N_{\text{tree}}}}^{\frac{N_{3,\text{max}}+1}{N_{\text{tree}}}} dx \frac{1}{x\sqrt{1-2x}} \left(\frac{1}{1-2x}\right)^{N_{\text{tree}}} \\
&\quad \times \left(\frac{1-2x}{x}\right)^{2xN_{\text{tree}}} \exp[\beta\tilde{\mu}_3 x N_{\text{tree}}]. \tag{C.3}
\end{aligned}$$

In the limit $N_{\text{tree}} \rightarrow \infty$, this integral can be evaluated using the Laplace's method:

$$\int_a^b h(x) e^{Ng(x)} dx \approx \sqrt{\frac{2\pi}{N|g''(x_0)|}} h(x_0) e^{Ng(x_0)}, \quad \text{for } N \rightarrow \infty \tag{C.4}$$

where x_0 is the value for which $g'(x_0) = 0$. From Eq. (C.3), we identify:

$$g(x) = -\ln(1-2x) + 2x \ln\left(\frac{1}{x} - 2\right) + \beta\tilde{\mu}_3 x, \tag{C.5}$$

$$g'(x) = 2 \ln\left(\frac{1}{x} - 2\right) + \beta\tilde{\mu}_3, \tag{C.6}$$

$$g''(x) = \frac{2}{x(2x-1)}, \tag{C.7}$$

$$h(x) = \frac{1}{x\sqrt{1-2x}}. \tag{C.8}$$

Solving $g'(x_0) = 0$ gives:

$$x_0 = \frac{1}{2 + \exp(-\beta\tilde{\mu}_3/2)}, \tag{C.9}$$

which is equal to the asymptotic mean fraction of branch points $\lambda(\tilde{\mu}_3)$ (Eqs. (4.31) and (4.35)). Note that this result is consistent with the branching probability λ of Eq. (3.62) after applying the tree-ring chemical-potential shift in Eq. (4.24). Finally, after simple manipulations, the resulting asymptotic form of the partition function is given, up to unimportant prefactors, by the simple expression:

$$Z_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3) \approx \frac{\exp(-\beta\tilde{\mu}_3)}{\sqrt{\pi x_0}} \frac{\exp[N_{\text{tree}}g(x_0)] c^{N_{\text{tree}}-1}}{N_{\text{tree}}^{3/2}} \tag{C.10}$$

$$\sim \frac{\left(\frac{c}{1-2\lambda}\right)^{N_{\text{tree}}}}{N_{\text{tree}}^{3/2}}. \tag{C.11}$$

whose exponential factor coincides with Eq. (4.30).

C.2 Approximate mean tree size with reptons, $\langle N_{\text{tree}} \rangle$

When N_{tree} is not conserved, we can use the scaling relation Eq. (C.10) and a saddle-point approximation to estimate the expected tree size $\langle N_{\text{tree}} \rangle$ by extremizing the logarithm of the N_{tree} -dependent statistical weight in Eq. (4.12).

Applying Stirling's approximation, we write to leading order:

$$\begin{aligned} \ln[\Omega_{\text{rep}}(N_{\text{ring}}, N_{\text{tree}}) Z_{\text{ring}}(N_{\text{tree}}, \tilde{\mu}_3)] &\propto \kappa(2(N_{\text{tree}} - 1) + N_{\text{rep}} - N_{\text{ring}}) \\ &\quad + N_{\text{ring}} \ln(N_{\text{ring}}) - 2(N_{\text{tree}} - 1) \ln(2(N_{\text{tree}} - 1)) \\ &\quad - N_{\text{rep}} \ln(N_{\text{rep}}) + N_{\text{tree}} \ln\left(\frac{c}{1 - 2\lambda}\right). \end{aligned} \quad (\text{C.12})$$

with the Lagrange multiplier κ enforcing the constraint in Eq. (4.2). Extremizing Eq. (C.12) with respect to N_{rep} , N_{tree} and κ yields

$$\kappa = 1 + \ln(N_{\text{rep}}), \quad (\text{C.13})$$

$$2\kappa = 2 \ln(2(N_{\text{tree}} - 1)) + 2 - \ln\left(\frac{c}{1 - 2\lambda}\right), \quad (\text{C.14})$$

$$N_{\text{ring}} = 2(N_{\text{tree}} - 1) + N_{\text{rep}}. \quad (\text{C.15})$$

Solving Eqs. (C.13)-(C.15) gives the asymptotic expression:

$$\frac{\langle N_{\text{tree}} \rangle - 1}{N_{\text{ring}}/2} = 1 - \frac{\langle N_{\text{rep}} \rangle/2}{N_{\text{ring}}/2} \approx \left(\sqrt{\frac{1 - 2\lambda}{c}} + 1 \right)^{-1}, \quad (\text{C.16})$$

which coincides with Eq. (4.32). By a similar procedure, one obtains Eqs. (4.33), (4.34) and (4.35) for $\langle N_1 \rangle$, $\langle N_2 \rangle$ and $\langle N_3 \rangle$.

C.3 Tightly double-folded rings of arbitrary functionality

To test the arbitrary-functionality multiplicity in Eq. (4.27), we use the simulation ensembles reported in the Supporting Information of Ref. [22]. There, several ensembles of tightly double-folded rings were simulated at fixed ring size $N_{\text{ring}} = 216$ and with no chemical potential applied, but with different imposed maximum functionalities $f_{\text{max}} \in \{3, 6, 9, 12\}$. For arbitrary maximum functionality, the tree constraints generalize to

$$\sum_{f=1}^{f=f_{\text{max}}} N_f = N_{\text{tree}}, \quad (\text{C.17})$$

$$\sum_{f=3}^{f=f_{\text{max}}} (f - 2)N_f = N_1 - 2. \quad (\text{C.18})$$

For generality, in the following we associate a functionality-specific chemical potential $\tilde{\mu}_f$ to each node of functionality f .

The asymptotic expression for the mean total number, $\langle N_f \rangle$, of nodes of functionality f can be computed using the same methods introduced in Sec. C.2. As before, we use a saddle-point approximation and extremize $\ln \left[\Omega_{\text{ring}}(N_{\text{tree}}, \{N_f\}) e^{\sum_{f=1}^{f_{\text{max}}} \beta \tilde{\mu}_f N_f} \right]$, with $\Omega_{\text{ring}}(N_{\text{tree}}, \{N_f\})$ given by Eq. (4.27). Incorporating the constraints (C.17) and (C.18) using Lagrange multipliers κ_1 and κ_2 , we obtain to leading order the expression:

$$\begin{aligned} & \ln(\Omega_{\text{ring}}(N_{\text{tree}}, \{N_f\})) + \sum_{f=1}^{f_{\text{max}}} \beta \tilde{\mu}_f N_f \\ & \approx \sum_{f=1}^{f_{\text{max}}} \beta \tilde{\mu}_f N_f + (N_{\text{tree}} - 1) \ln(N_{\text{tree}} - 1) - \sum_{f=1}^{f_{\text{max}}} N_f \ln(N_f) \\ & \quad + \kappa_1 \left(N_{\text{tree}} - \sum_f N_f \right) + \kappa_2 \left(N_1 - 2 - \sum_{f=3}^{f_{\text{max}}} (f-2) N_f \right). \end{aligned} \quad (\text{C.19})$$

Extremizing with respect to N_f , κ_1 and κ_2 yields the system

$$N_f = \exp(-(f-2)\kappa_2 - \kappa_1 - 1 + \beta \tilde{\mu}_f), \quad (\text{C.20})$$

$$N_1 = \sum_{f=3}^{f_{\text{max}}} (f-2) N_f + 2, \quad (\text{C.21})$$

$$N_{\text{tree}} = \sum_{f=1}^{f_{\text{max}}} N_f, \quad (\text{C.22})$$

which, after neglecting terms $\mathcal{O}(1/N_{\text{tree}})$ and smaller, can be written in the compact form:

$$\frac{N_f}{N_{\text{tree}}} = \frac{\exp(-(f-2)\kappa_2 + \beta \tilde{\mu}_f)}{\sum_{f'=1}^{f_{\text{max}}} \exp(-(f'-2)\kappa_2 + \beta \tilde{\mu}_{f'})}, \quad (\text{C.23})$$

$$\frac{N_1}{N_{\text{tree}}} = \sum_{f=3}^{f_{\text{max}}} (f-2) \frac{N_f}{N_{\text{tree}}}. \quad (\text{C.24})$$

Then, introducing $\alpha = e^{-\kappa_2}$, Eq. (C.24) is reduced to the simple polynomial equation:

$$1 = \sum_{f=3}^{f_{\text{max}}} (f-2) \alpha^{f-1} \exp[-\beta(\tilde{\mu}_1 - \tilde{\mu}_f)], \quad (\text{C.25})$$

which must be solved numerically to find the optimal α . Substitution into Eq. (C.23) gives

$$\frac{N_f}{N_{\text{tree}}} = \frac{\alpha^f \exp(\beta \tilde{\mu}_f)}{\sum_{f'=1}^{f_{\text{max}}} \alpha^{f'} \exp(\beta \tilde{\mu}_{f'})}. \quad (\text{C.26})$$

Notice that a closely related expression in the context of a mean-field approximation for

interacting (*i.e.*, non-ideal) trees was derived recently in Ref. [79]. For branching chemical potentials $\{\tilde{\mu}_f = 0\}$, Eqs. (C.25) and (C.26) recover the expected limiting cases $f_{\max} = 3$ and $f_{\max} \rightarrow \infty$.

Bibliography

- [1] Luca Tubiana, Gareth P. Alexander, Agnese Barbensi, Dorothy Buck, Julyan H.E. Cartwright, Mateusz Chwastyk, Marek Cieplak, Ivan Coluzza, Simon Čopar, David J. Craik, Marco Di Stefano, Ralf Everaers, Patrícia F.N. Faísca, Franco Ferrari, Achille Giacometti, Dimos Goundaroulis, Ellinor Haglund, Ya-Ming Hou, Nevena Ilieva, Sophie E. Jackson, Aleksandre Japaridze, Noam Kaplan, Alexander R. Klotz, Hongbin Li, Christos N. Likos, Emanuele Locatelli, Teresa López-León, Thomas Machon, Cristian Micheletti, Davide Michieletto, Antti Niemi, Wanda Niemyska, Szymon Niewieczerzal, Francesco Nitti, Enzo Orlandini, Samuela Pasquali, Agata P. Perlinska, Rudolf Podgornik, Raffaello Potestio, Nicola M. Pugno, Miha Ravnik, Renzo Ricca, Christian M. Rohwer, Angelo Rosa, Jan Smrek, Anton Souslov, Andrzej Stasiak, Danièle Steer, Joanna Sułkowska, Piotr Sułkowski, De Witt L. Sumners, Carsten Svaneborg, Piotr Szymczak, Thomas Tarenzi, Rui Travasso, Peter Virnau, Dimitris Vlassopoulos, Primož Ziherl, and Slobodan Žumer. Topology in soft and biological matter. *Phys. Rep.*, 1075:1–137, 2024.
- [2] Paul J. Flory. Molecular size distribution in three dimensional polymers. I. Gelation. *J. Am. Chem. Soc.*, 63(11):3083–3090, 1941.
- [3] Paul J. Flory. Molecular size distribution in three dimensional polymers. II. Trifunctional branching units. *J. Am. Chem. Soc.*, 63(11):3091–3096, 1941.
- [4] Paul J. Flory. Molecular size distribution in three dimensional polymers. III. Tetrafunctional branching units. *J. Am. Chem. Soc.*, 63(11):3096–3100, 1941.
- [5] Walter H. Stockmayer. Theory of molecular size distribution and gel formation in branched-chain polymers. *J. Chem. Phys.*, 11(2):45–55, 1943.
- [6] B. H. Zimm and W. H. Stockmayer. The dimensions of chain molecules containing branches and rings. *J. Chem. Phys.*, 17:1301–1314, 1949.
- [7] A. M. Gutin, A. Y. Grosberg, and E. I. Shakhnovich. Polymers with annealed and quenched branchings belong to different universality classes. *Macromolecules*, 26:1293–1295, 1993.
- [8] Ralf Everaers, Alexander Y. Grosberg, Michael Rubinstein, and Angelo Rosa. Flory theory of randomly branched polymers. *Soft Matter*, 13:1223–1234, 2017.

- [9] Charles M. Schroeder, Ralf Everaers, Kurt Kremer, Margarita Kruteva, Christos N. Likos, Gregory B. McKenna, Thomas O'Connor, J. Ravi Prakash, Dieter Richter, Rae Robertson-Anderson, Michael Rubinstein, Kenneth S. Schweizer, and Dimitris Vlassopoulos. Ring polymer physics and rheology: Challenges and opportunities. *J. Rheol.*, 70(1):183–216, 2026.
- [10] A. Y. Grosberg. Annealed lattice animal model and Flory theory for the melt of non-concatenated rings: towards the physics of crumpling. *Soft Matter*, 10:560–565, 2014.
- [11] A. Rosa and R. Everaers. Ring polymers in the melt state: the physics of crumpling. *Phys. Rev. Lett.*, 112:118302, 2014.
- [12] Ivan Junier, Elham Ghobadpour, Olivier Espeli, and Ralf Everaers. DNA supercoiling in bacteria: state of play and challenges from a viewpoint of physics based modeling. *Front. Microbiol.*, 14:1192831, 2023.
- [13] F. Brock Fuller. Decomposition of the linking number of a closed ribbon: A problem from molecular biology. *Proc. Natl. Acad. Sci. USA*, 75(8):3557–3561, 1978.
- [14] L. F. Liu and J. C. Wang. Supercoiling of the DNA template during transcription. *Proc. Natl. Acad. Sci. USA*, 84(20):7024–7027, 1987.
- [15] James C. Wang. *Untangling the double helix: DNA entanglement and the action of the DNA topoisomerases*. Cold Spring Harbor Laboratory Press, 2009.
- [16] John F. Marko and Eric D. Siggia. Fluctuations and supercoiling of DNA. *Science*, 265(5171):506–508, 1994.
- [17] J. F. Marko and E. D. Siggia. Statistical mechanics of supercoiled DNA. *Phys. Rev. E*, 52:2912–2938, 1995.
- [18] John F. Marko. The internal ‘slithering’ dynamics of supercoiled DNA. *Physica A*, 244(1):263–277, 1997.
- [19] John F. Marko. Short note on the scaling behavior of communication by ‘slithering’ on a supercoiled DNA. *Physica A*, 296:289–292, 2001.
- [20] Elham Ghobadpour, Max Kolb, Mohammad Reza Ejtehadi, and Ralf Everaers. Monte Carlo simulation of a lattice model for the dynamics of randomly branching double-folded ring polymers. *Phys. Rev. E*, 104:014501, 2021.
- [21] Elham Ghobadpour. *Multiple Scale Modeling of Bacterial Chromosome*. PhD thesis, Université Grenoble Alpes, 2023.
- [22] Elham Ghobadpour, Max Kolb, Ivan Junier, and Ralf Everaers. The emergent dynamics of double-folded randomly branching ring polymers. *Macromolecules*, 58(19):10632–10646, 2025.

- [23] A. Yu. Grosberg. How two meters of DNA fit into a cell nucleus: Polymer models with topological constraints and experimental data. *Polym. Sci. Ser. C*, 54:1–10, 2012.
- [24] Mattia Conte, Ehsan Irani, Andrea M. Chiariello, Alex Abraham, Simona Bianco, Andrea Esposito, and Mario Nicodemi. Loop-extrusion and polymer phase-separation can co-exist at the single-molecule level to shape chromatin folding. *Nat. Commun.*, 13(1):4070, 2022.
- [25] Kirill E. Polovnikov, Hugo B. Brandão, Sergey Belan, Bogdan Slavov, Maxim Imakaev, and Leonid A. Mirny. Crumpled polymer with loops recapitulates key features of chromosome organization. *Phys. Rev. X*, 13:041029, 2023.
- [26] Brian Chan and Michael Rubinstein. Activity-driven chromatin organization during interphase: Compaction, segregation, and entanglement suppression. *Proc. Natl. Acad. Sci. USA*, 121(21):e2401494121, 2024.
- [27] Job Dekker and Leonid A. Mirny. The chromosome folding problem and how cells solve it. *Cell*, 187(23):6424–6450, 2024.
- [28] Erez Lieberman-Aiden, Nynke L. van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R. Lajoie, Peter J. Sabo, Michael O. Dorschner, Richard Sandstrom, Bradley Bernstein, M. A. Bender, Mark Groudine, Andreas Gnirke, John Stamatoyannopoulos, Leonid A. Mirny, Eric S. Lander, and Job Dekker. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, 326(5950):289–293, 2009.
- [29] Takashi Nagano, Yaniv Lubling, Tim J. Stevens, Stefan Schoenfelder, Eitan Yaffe, Wendy Dean, Ernest D. Laue, Amos Tanay, and Peter Fraser. Single-cell Hi-C reveals cell-to-cell variability in chromosome structure. *Nature*, 502(7469):59–64, 2013.
- [30] Tatsuya Hirano. Condensin-based chromosome organization from bacteria to vertebrates. *Cell*, 164(5):847–857, 2016.
- [31] Iain F. Davidson, Roman Barth, Maciej Zaczek, Jaco van der Torre, Wen Tang, Kota Nagasaka, Richard Janissen, Jacob Kerssemakers, Gordana Wutz, Cees Dekker, and Jan-Michael Peters. CTCF is a DNA-tension-dependent barrier to cohesin-mediated loop extrusion. *Nature*, 616(7958):822–827, 2023.
- [32] A. Rosa and R. Everaers. Structure and dynamics of interphase chromosomes. *PLoS Comput. Biol.*, 4:e1000153, 2008.
- [33] Jonathan Halverson, Jan Smrek, Kurt Kremer, and Alexander Y. Grosberg. From a melt of rings to chromosome territories: the role of topological constraints in genome folding. *Rep. Prog. Phys.*, 77(2):022601, 2014.
- [34] Domen Vaupotič, Angelo Rosa, Luca Tubiana, and Anže Božič. Scaling properties of RNA as a randomly branching polymer. *J. Chem. Phys.*, 158(23):234901, 2023.

- [35] M. Doi and S. F. Edwards. *The Theory of Polymer Dynamics*. Oxford University Press, New York, 1986.
- [36] P.-G. De Gennes. *Scaling Concepts in Polymer Physics*. Cornell University Press, Ithaca, 1979.
- [37] M. Rubinstein and R. H. Colby. *Polymer Physics*. Oxford University Press, New York, 2003.
- [38] H. A. Kramers. The behavior of macromolecules in inhomogeneous flow. *J. Chem. Phys.*, 14(7):415–424, 1946.
- [39] P. J. Flory. *Statistical Mechanics of Chain Molecules*. Interscience, New York, 1969.
- [40] J. C. Le Guillou and J. Zinn-Justin. Critical exponents for the n -vector model in three dimensions from field theory. *Phys. Rev. Lett.*, 39:95–98, 1977.
- [41] A. Rosa and R. Everaers. Computer simulations of melts of randomly branching polymers. *J. Chem. Phys.*, 145:164906, 2016.
- [42] A. Rosa and R. Everaers. Beyond Flory theory: Distribution functions for interacting lattice trees. *Phys. Rev. E*, 95:012117, 2017.
- [43] Angelo Rosa and Ralf Everaers. Conformational statistics of randomly branching double-folded ring polymers. *Eur. Phys. J. E*, 42(1):7, 2019.
- [44] M. Müller, J. P. Wittmer, and M. E. Cates. Topological effects in ring polymers: A computer simulation study. *Phys. Rev. E*, 53:5063–5074, 1996.
- [45] M. Müller, J. P. Wittmer, and M. E. Cates. Topological effects in ring polymers. II. Influence of persistence length. *Phys. Rev. E*, 61:4078–4089, 2000.
- [46] T. Vettorel, A. Y. Grosberg, and K. Kremer. Territorial polymers. *Phys. Today*, 62:72, 2009.
- [47] J. D. Halverson, W. B. Lee, G. S. Grest, A. Y. Grosberg, and K. Kremer. Molecular dynamics simulation study of nonconcatenated ring polymers in a melt: I. Statics. *J. Chem. Phys.*, 134:204904, 2011.
- [48] J. D. Halverson, G. S. Grest, A. Y. Grosberg, and K. Kremer. Rheology of ring polymer melts: From linear contaminants to ring-linear blends. *Phys. Rev. Lett.*, 108:038301, 2012.
- [49] Raoul D. Schram, Angelo Rosa, and Ralf Everaers. Local loop opening in untangled ring polymer melts: a detailed “Feynman test” of models for the large scale structure. *Soft Matter*, 15:2418–2429, 2019.

- [50] M. E. Cates and J. M. Deutsch. Conjectures on the statistics of ring polymers. *J. Phys. (Paris)*, 47:2121–2128, 1986.
- [51] T. Sakaue. Ring polymers in melts and solutions: Scaling and crossover. *Phys. Rev. Lett.*, 106:167802, 2012.
- [52] A. R. Khokhlov and S. K. Nechaev. Polymer chain in an array of obstacles. *Phys. Lett.*, 112A:156–160, 1985.
- [53] M. Rubinstein. Dynamics of ring polymers in the presence of fixed obstacles. *Phys. Rev. Lett.*, 57:3023–3026, 1986.
- [54] S. Obukhov, A. Johnner, J. Baschnagel, H. Meyer, and J. P. Wittmer. Melt of polymer rings: The decorated loop model. *EPL*, 105(4):48005, 2014.
- [55] Ting Ge, Sergey Panyukov, and Michael Rubinstein. Self-similar conformations and dynamics in entangled melts and solutions of nonconcatenated ring polymers. *Macromolecules*, 49(2):708–722, 2016.
- [56] S. F. Edwards. Statistical mechanics with topological constraints: I. *Proc. Phys. Soc.*, 91:513–519, 1967.
- [57] Mattia A. Ubertyini and Angelo Rosa. Ring polymers in two-dimensional melts double-fold around randomly branching “primitive shapes”. *Soft Matter*, 21(45):8711–8723, 2025.
- [58] Mattia Alberto Ubertyini, Jan Smrek, and Angelo Rosa. Entanglement length scale separates threading from branching of unknotted and non-concatenated ring polymers in melts. *Macromolecules*, 55(23):10723–10736, 2022.
- [59] Davide Michieletto and Matthew S. Turner. A topologically driven glass in ring polymers. *Proc. Natl. Acad. Sci. USA*, 113:5195–5200, 2016.
- [60] A. Rosa and R. Everaers. Computer simulations of randomly branching polymers: Annealed *vs.* quenched branching structures. *J. Phys. A: Math. Theor.*, 49:345001, 2016.
- [61] A. Cayley. A theorem on trees. *Q. J. Math.*, 23:376–378, 1889.
- [62] H. Prüfer. Neuer Beweis eines Satzes über Permutationen. *Arch. Math. Phys.*, 27:742–744, 1918.
- [63] Surendra W. Singaram, Ajaykumar Gopal, and Avinoam Ben-Shaul. A Prüfer-sequence based algorithm for calculating the size of ideal randomly branched polymers. *J. Phys. Chem. B*, 120:6231–6237, 2016.
- [64] J. W. Moon. The second moment of the complexity of a graph. *Mathematika*, 11(2):95–98, 1964.

- [65] OEIS Foundation Inc. A000055: Number of trees with n unlabeled nodes. Published electronically at <https://oeis.org>, 2026. Part of The On-Line Encyclopedia of Integer Sequences, Accessed: September 18, 2026.
- [66] Richard Otter. The number of trees. *Ann. Math.*, 49(3):583–599, 1948.
- [67] P.-G. De Gennes. Statistics of branching and hairpin helices for the dAT copolymer. *Biopolymers*, 6:715, 1968.
- [68] M. Daoud and J. F. Joanny. Conformation of branched polymers. *J. Physique*, 42:1359–1371, 1981.
- [69] J. Bertrand. Solution d’un problème. *C. R. Acad. Sci. Paris*, 105:369, 1887.
- [70] F. Eggenberger and G. Pólya. Über die Statistik verketteter Vorgänge. *Z. Angew. Math. Mech.*, 3(4):279–289, 1923.
- [71] Ally Picciotto. *How to encode a tree*. PhD thesis, University of California, San Diego, 1999.
- [72] Evan Thompson, Tim Paulden, and David K. Smith. The dandelion code: A new coding of spanning trees for genetic algorithms. *IEEE Trans. Evol. Comput.*, 11(1):91–100, 2007.
- [73] W. A. Seitz and D. J. Klein. Excluded volume effects for branched polymers. *J. Chem. Phys.*, 75:5190–5193, 1981.
- [74] Frederick T. Wall and Frederic Mandel. Macromolecular dimensions obtained by an efficient Monte Carlo method without sample attrition. *J. Chem. Phys.*, 63(11):4592–4595, 1975.
- [75] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.
- [76] W. M. Young and E. W. Elcock. Monte Carlo studies of vacancy migration in binary ordered alloys: I. *Proc. Phys. Soc.*, 89(3):735, 1966.
- [77] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. A new algorithm for Monte Carlo simulation of Ising spin systems. *J. Comput. Phys.*, 17(1):10–18, 1975.
- [78] Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22(4):403–434, 1976.
- [79] Davide Marcato, Achille Giacometti, Amos Maritan, and Angelo Rosa. Entropy of self-avoiding branching polymers: Mean-field theory and Monte Carlo simulations. *J. Chem. Phys.*, 163(14):141101, 2025.

- [80] M. Rubinstein. Discretized model of entangled-polymer dynamics. *Phys. Rev. Lett.*, 59:1946–1949, 1987.
- [81] E. J. Janse van Rensburg and N. Madras. A nonlocal Monte Carlo algorithm for lattice trees. *J. Phys. A: Math. Gen.*, 25:303–333, 1992.
- [82] Hsiao-Ping Hsu, Walter Nadler, and Peter Grassberger. Simulations of lattice animals and trees. *J. Phys. A: Math. Gen.*, 38(4):775, 2005.
- [83] Peter Grassberger. Pruned-enriched Rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000. *Phys. Rev. E*, 56:3682–3693, 1997.
- [84] David Bruce Wilson. Generating random spanning trees more quickly than the cover time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 296–303, New York, NY, USA, 1996. Association for Computing Machinery.
- [85] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, San Diego, second edition, 2002.
- [86] Pieter H. W. van der Hoek. Sampling ideal trees [Software]. <https://github.com/PHWvdH/Sampling-Ideal-trees/>, 2026. GitHub repository.
- [87] Pieter H. W. van der Hoek. The configurational entropy of random trees [Data set]. <https://github.com/PHWvdH/Configurational-Entropy-of-Random-Trees/>, 2026. GitHub repository.
- [88] Elham Ghobadpour and Pieter H. W. van der Hoek. Coarse-Grain Bacterial DNA Simulator (CGBD). <https://gitlab.com/e.ghobadpour/CGBD>, 2026. GitLab repository.
- [89] É. Barbier. Généralisation du problème résolu par M. J. Bertrand. *C. R. Acad. Sci. Paris*, 105:409, 1887.
- [90] D. André. Solution directe du problème résolu par M. Bertrand. *C. R. Acad. Sci. Paris*, 105:436–437, 1887.
- [91] A. Aeppli. A propos de l’interprétation géométrique du problème du scrutin. *Enseign. Math.*, 23:328–329, 1923.
- [92] Marc Renault. Four proofs of the ballot theorem. *Math. Mag.*, 80:345–347, 2007.