Enumeration of rooted constellations and hypermaps through quantum matrix integrals

Jacob P. Dyer

PhD

University of York
Mathematics

September 2015

Abstract

We present a new method for enumeration of rooted constellations and other objects these can represent, specifically rooted hypermaps and maps. We derive a closed-form generating function enumerating rooted hypermaps with one face and a fixed number of darts, partitioned by number of edges, and vertices. We derive an algorithmic procedure for calculating generating functions enumerating all rooted hypermaps for fixed number of darts, partitioning by number of edges, vertices and faces, as well as an analogous procedure for enumerating rooted maps for fixed edge count. We also look at the enumeration problem for general rooted constellations, but do not calculate generating functions. Using these results we find recursion relations for calculating the total number of rooted hypermaps, maps and constellations of any given degree.

This method is based on matrix integration tools originally developed in the study of bipartite quantum systems, specifically in calculating mean properties of their subsystems, where the averaging is over all possible pure states of the overall system. We present this work first, studying the mean von Neumann entropy of entanglement between the quantum system's two subsystems. We look at an unproven entropy approximation proposed by Lubkin (1978), derived from an infinite series expansion of the entropy which was not known to be convergent. We prove that this series is convergent if and only if the subsystem being studied is of dimension two, by deriving closed-form expressions for the series terms and finding their limiting behaviour. In light of this we examine the validity of Lubkin's approximation rigorously, confirming the limit in which it *is* valid, but deriving a more accurate approximation in the process.

Contents

Acknowledgements

I wish to thank my supervisor, Bernard Kay, for providing, and helping me with, the initial problem my work started with, and for staying with me when I then took the work off in a completely different direction. I particularly thank him for his assistance during the process of writing and submitting my papers. I am grateful to Christopher Hughes, also from my department, for introducing me to Zeilberger's algorithm and the fascinating book $A{=}B$. I also thank EPSRC for funding my research. I am grateful to all of my family and friends, at home, at university and around the world, for providing me with support and encouragement and helping keep me on track.

Declaration

I declare that the work presented in this thesis, unless otherwise stated, is based on my own research and has not been submitted previously for a degree at this or any other university. Parts of this work have been submitted for publication, and can be found at:

- J. P. Dyer, Divergence of Lubkin's series for a quantum subsystem's mean entropy, arXiv:1406.5776 (2014)
- J. P. Dyer, Matrix integrals and generating functions for permutations and one-face rooted hypermaps, arxiv:1407.7774 (2014)
- J. P. Dyer, Matrix integrals and generating functions for enumerating rooted hypermaps by vertices, edges and faces for a given number of darts, arxiv:1411.3534 (2014)

Part I

Introduction

Chapter 1

Quantum theory and enumeration

Over the years a number of interesting, sometimes surprising, mathematical results have been obtained through the discovery of overlaps between otherwise disconnected fields of mathematics, and one context in which this is particularly the case is the use of combinatorics in quantum theory. A good example of this is the well-known Feynman diagram method for modelling particle interactions in the Standard Model and other related models [33, 14, 13, 25].

This example nicely demonstrates the reason why this link exists. When quantum particles meet, e.g. in a particle accelerator, the likelihood of them interacting is given by a *scattering amplitude*, a function of the particles' momenta, which is expressed as a Gaussian-weighted integral. In general these integrals cannot be solved exactly, but they can be expanded as an infinite series where each of the terms is a simpler Gaussian integral. Without going into the technical details of actual quantum scattering amplitudes, a simple example of this principle is the integral

$$\int_0^\infty e^{-x^2+kx}dx = \sum_{n=0}^{\infty} \frac{k^n}{n!} \int_0^\infty x^n e^{-x^2} dx = \sum_{n=0}^{\infty} \frac{\Gamma[(n+1)/2]}{2n!} k^n,$$

where

$$\Gamma[(n+1)/2] = \begin{cases} [(n-1)/2]!, & n \text{ odd} \\ \frac{\sqrt{\pi}(n-1)!!}{2^{n/2}}, & n \text{ even} \end{cases}.$$

So, apart from the linear dependence of every other term on $\sqrt{\pi}$ (which we can calculate numerically to high accuracy), we can evaluate all of the terms in this series exactly in finite time, despite the original integral having no closed-form solution[1]. This is useful in the numerical calculation of scattering amplitudes, as it allows for approximations of true amplitudes to be found by taking truncations of these series expansions.

Feynman diagrams then enter into the problem because each term in such a series turns out to have a direct correspondence to a combinatorial diagram. These diagrams

[1] Note that the lower bound of integration is zero. The integral would have a much simpler solution if the integral were unbounded in both directions.

all satisfy a well-defined set of rules governing their construction, and their properties are determined by the nature of the interaction being modelled. As the correspondence between series term and diagram is one-to-one, the task of generating the list of terms in the series expansion is equivalent to the task of constructing all valid diagrams with the required properties. From these diagrams the series terms can then be reconstructed and summed together.

Feynman diagrams in particular also have the useful property that they can be interpreted physically as representing classical trajectories of interacting particles. This leads to the now-common interpretation of quantum interactions as an average over all possible classical interactions matching the observed outcome.

What is less often studied, however, is the reverse of this sort of connection. Combinatorics can be used to calculate approximate results in quantum theory, but at the same time tools from quantum theory can in some cases be used to derive results in combinatorics. What's more, these results are often *exact*, not approximate.

This is particularly the case in enumerative combinatorics, where for instance quantum procedures have been used to enumerate maps [39]. The parallel between the two problems is clear, again as demonstrated by the Feynman diagram problem. Combinatorial enumeration involves summing over all possible objects (e.g. maps) with a specified set of properties (e.g. a given number of edges), while computation of scattering amplitudes involves summation over all possible classical paths with a specified initial and final condition, as represented by Feynman diagrams.

Mathematically, the reason is also clear. As has already been noted, Feynman diagrams arise from expansion of Gaussian integral expressions, while the map enumeration example also relies on Gaussian integrals [39]. It is through this shared dependence on this one mathematical tool that the connection arises.

Our topic here is a new example of such a connection. Specifically we will look at matrix integrals whose forms arise in the study of finite-dimensional bipartite quantum systems, and we will show that they can be used in the enumeration of objects called *rooted constellations* [20]. Due to links between constellations and a number of more familiar combinatorial objects we will use this to derive results for *rooted maps* and *hypermaps [20]* as well.

This task has many similarities with the Feynman diagram example. As was the case there we will start by computing a property of a quantum system which is defined as a mean over all possible configurations of the system. Specifically we will look at a bipartite quantum system of finite dimension, and at the mean of the *von Neumann entropy of entanglement* over all possible pure states of this system. As with Feynman diagrams, the method we will use involves taking the integral which defines the mean, expanding it as an infinite series of simpler integrals, each of which we can evaluate explicitly.

Our initial motivation for studying this problem was purely quantum – we wished to know if and when this series was convergent, and if a truncation of it could be used as an approximation of the mean entropy. While we did solve this problem, showing

that the series only converged when one of the systems was two-dimensional, a much more intriguing result ended up arising from the individual series terms themselves. Each of these terms was a multi-dimensional Gaussian integral, and they evaluated to rational functions of the dimensions of the quantum subsystems. Upon closer inspection, the numerators of these rational functions turned out to be generating functions (polynomials whose coefficients are equal to counts of combinatorial objects), which enumerate rooted hypermaps with one face.

This link was discovered just by searching for other known occurrences of the same set of integers, but the process of proving the connection revealed an underlying combinatorial structure in the class of integrals we had been using, and more importantly a way in which this structure could be further exploited in other cases. Staying within the field of diagrams on two-dimensional surfaces initially, we found first that the method could be extended to all rooted hypermaps, not just those with one face, and then to rooted maps as well.

What's more, the proofs of these facts relied on a particular representation of maps and hypermaps – the aforementioned constellations. Constellations are in a sense a purer form of combinatorial object, not dependent on any geometrical interpretation (maps are defined as graphs embedded on two-dimensional orientable surfaces, and, as a generalisation of maps, hypermaps have a similar interpretation), defined instead as sets of permutations. Constellations can be generalised significantly beyond those used to represent hypermaps, and sure enough, our matrix-integral methods can also be applied to these more general objects.

Ultimately the reason why all of these methods work lies in the fact that all these objects (hypermaps, maps) are specialised types of constellation, and conversely it is the fact that constellations find use in a number of different topics that makes these methods so versatile. It must be noted however that the power of these methods varies considerably from problem to problem, with the more specialised cases in general producing the most results. For instance, we are only able to explicitly calculate generating functions in the cases of maps and hypermaps, and not for general constellations. We are able to find some information in all cases, such as overall counts for each type of object at each degree[2], but still more information remains hidden within intractable integral expressions.

Nonetheless, the generating functions we are able to calculate are themselves very powerful. Prior work exists on rooted hypermap enumeration, but this has often been restricted to specific genera of hypermap, such as planar [4] and toroidal [3] hypermaps. A method for generating individual rooted hypermaps has been found, allowing for them to be counted directly [35] (it is from this work that we originally discovered the link), but as this relies on generating all of the hypermaps individually it cannot tell us anything about the properties of the generating functions, such as symmetries, recursions and other patterns. Our method allows for production of generating functions for any degree of rooted hypermap, and, as was the case in the

[2]By *degree* we mean the number of edges in a map, and analogously the number of *darts* in a hypermap. The meaning of these terms will be given in more detail in Chapter 4.

above references, the hypermaps are then partitioned by number of vertices, edges and faces. Furthermore, as well as the benefits generating functions bring in allowing for proofs of patterns and symmetries, this method is also much faster at computing counts than Walsh's method, simply because it doesn't require listing the individual hypermaps.

As our method arises from constellations, however, these generating functions are not partitioned by the geometric genus of the maps and hypermaps, as the above methods were. This is in fact beneficial as it means that a fundamental symmetry of the set of hypermaps (under exchange of edges, vertices and faces) becomes immediately apparent.

This is not to say that any information is lost, of course. While genus information is not naturally apparent in the resulting generating functions, the Euler characteristic formula directly links the genus of hypermaps to their edge/vertex/face counts, so this information can still easily be extracted.

Ultimately, while we will explicitly go through the derivation of a number of specific results in this thesis, for hypermaps, maps, and constellations, the main aim is to provide a demonstration of the methods themselves and how they can be adapted to suit a wide range of different enumeration problems. The hope is that these tools may then find further use in new contexts, within enumerative combinatorics and beyond.

Chapter 2

Structure of this thesis

While the primary results of this thesis are almost entirely within combinatorics, the story of their derivation is very much in two distinct parts, and this thesis is structured to reflect this.

In Part II we begin with the the original quantum problem which motivated our study. This work is stand-alone, with its own aims and conclusions, and we will treat it as such. This problem, as we have briefly mentioned, is that of determining the convergence properties of an infinite series expansion for the mean von Neumann entropy of a bipartite quantum system. This question is tackled, and answered, in Chapter 6.

This work makes considerable use of matrix integration, so before we can tackle this problem, we first need to introduce the specific types of integrals we will be using, and the methods we will use to manipulate and evaluate them. After a brief introduction and summary of the quantum information concepts from which the integrals derive (Chapters 3 and 4), we look at these methods in Chapter 5. It is in these integrals that we begin to see combinatorial information appearing, so as well as looking at the integral methods required to solve the entropy problem (Section 5.2) we also briefly introduce a second method in Section 5.1, based on Gaussian integration, which we will use extensively later on to provide the connection between these integrals and combinatorial generating functions.

We then move on to combinatorial enumeration proper in Part III. We start with a fairly qualitative look at the expressions derived in Part II, and describe how their relevance to combinatorics was discovered, in Chapter 8, then give a general introduction of the combinatorial concepts used throughout this part in Chapter 9, including hypermaps and constellations. This provides us with an opportunity to highlight which properties and interpretations of these various objects will be of most importance to us, setting the ground-work for the rest of the part.

We then begin our enumeration work in Chapter 10 with our simplest example, where we show that the expressions derived in Part II are in fact generating functions for enumerating rooted hypermaps with one face. It is here that we first show a definite connection between quantum theory and combinatorics by proving these functions' nature as generating functions, and the methods introduced in the proof of

this provide the basis for proving later analogous results for rooted hypermaps, maps and constellations.

We start the process of generalising this work in Chapter 11, taking the methods from the one-face case and applying them first to rooted hypermaps with two faces in Section 11.1. Then, from here, we move straight to enumerating rooted hypermaps with any number of faces in the rest of the chapter. This generalisation introduces a number of new complexities, and we demonstrate how to overcome these and unlock the full potential of our methods. We also look at ways in which the generating functions being produced may be manipulated to extract further information, such as overall counts of rooted hypermaps of any given degree.

At this point we have most of our enumeration tools in place, so we move on to demonstrations of how to apply these tools to further problems in Chapter 12. We first look at rooted maps in Section 12.1, and the shared properties between maps and hypermaps become apparent in that the two methods and results end up resembling each other very closely. Again we find a method of computing generating functions, as well as a way of counting all rooted maps with a given number of edges.

Following from this case, which is essentially a specialisation of the hypermap method, we then move in the other direction and show how to generalise the method to all rooted constellations in Section 12.2. This is where we find ourselves furthest from our quantum starting point, and a number of the tools we have been using up to this point no longer apply, leaving us unable to compute the generating functions themselves as we have been able to before. We are still able to produce counts for the total number of rooted constellations of a given degree, however.

Part II

Random states of bipartite quantum systems

Chapter 3

Introduction

In this part we look at a question from quantum information theory, concerning entanglement in finite-dimensional bipartite quantum systems. Mathematically, a quantum system is described by a complex Hilbert space $\mathscr{H}$, with a finite-dimensional quantum system then having a Hilbert space isomorphic to $\mathbb{C}^m$ for some positive integer m. In these terms, a bipartite quantum system is one whose Hilbert space has some specified decomposition as a non-trivial tensor product

$$\mathscr{H} = \mathbb{C}^{m_1 m_2} = \mathbb{C}^{m_1} \otimes \mathbb{C}^{m_2}.$$

The study of quantum systems is generally concerned with examining properties of *states* of these systems. A state is described by a *density operator* $\hat{\rho}^{12}$, a positive semidefinite Hermitian operator with unit trace acting on $\mathscr{H}$, and this operator encodes all of the physical properties of the overall system in the given state. In a bipartite system $\hat{\rho}^{12}$ describes the state of the overall system, but we can also then give denisity operators which just describe the states of the two subsystems; these are given by the partial traces of $\hat{\rho}^{12}$,

$$\begin{aligned}\hat{\rho}_1^{12} &= \mathrm{Tr}_{\mathbb{C}^{m_2}}[\hat{\rho}^{12}] \\ \hat{\rho}_2^{12} &= \mathrm{Tr}_{\mathbb{C}^{m_1}}[\hat{\rho}^{12}],\end{aligned}$$

and are referred to as *reduced density operators*. Using these we can compute properties of the subsystems in isolation, just as we would use $\hat{\rho}^{12}$ to compute properties of the overall system.

We are interested in one specific subsystem property, the *von Neumann entropy*[1], which for the first subsystem is defined as

$$S_1^{12} = -\mathrm{Tr}_{\mathbb{C}^{m_1}}[\hat{\rho}_1^{12} \ln \hat{\rho}_1^{12}].$$

We can also define the entropy S_2^{12} of the second subsystem using the same formula, and in general these two entropies will have different values. However, in the cases we

[1] As the von Neumann entropy is the only form of entropy we consider here, we will often refer to it as just the entropy for short.

will be studying here – *pure states,* where the associated $\hat{\rho}^{12}$ is a projection operator – the two reduced density operators have the same eigenvalue spectrum, and as a result the two entropies S_1^{12} and S_2^{12} are necessarily equal. This allows us to view the entropy as a property of the overall system, and we can then use it as a measure of one of the characteristic traits of quantum systems: *entanglement.*

Roughly speaking, entanglement describes how dependent the states of the two subsystems of the system are on each other. In a completely disentangled state the two subsystems would be independent, and $\hat{\rho}^{12}$ would just be the tensor product of $\hat{\rho}_1^{12}$ and $\hat{\rho}_2^{12}$. In more general states, however, the two subsystems' states become intermixed, and $\hat{\rho}^{12}$ stops being separable in this way.

The von Neumann entropy is then a measure of the level of this intermixing. As we are taking $\hat{\rho}^{12}$ to be a projection operator, when there is no entanglement $\hat{\rho}_1^{12}$ and $\hat{\rho}_2^{12}$ must be projection operators as well, and the resulting entropy is zero. For entangled states the entropy has a non-zero positive value, with higher values indicating greater entanglement, up to a maximum value equal to the logarithm of the smallest of the two dimensions m_1 and m_2. A state with this entropy is said to be *maximally entangled.*

Our work here begins with a question raised by Lubkin [22] while studying the value of the entropy averaged over all possible pure states of a bipartite system. Unable to find an exact expression for this mean, he was attempting to find a way of approximating its value. He did so by writing a Taylor series expansion for the mean entropy (based on the Taylor expansion of the logarithm), truncating after the first few terms and evaluating it explicitly. His motivation for doing so was of course a physical one; he wanted to show that a random pure state of a large bipartite quantum system could with reasonable likelihood have subsystems with high entropy. His argument was that, although the universe we observe has very high entropy, it would be much more satisfying for the overall state of the universe to be pure, with the entropy we see resulting from entanglement with some other subsystem beyond our observation.

While this physical problem provided the original basis for Lubkin's work, the details which we will be concerned with here are purely mathematical in nature; we wish to determine if and when Lubkin's series expansion for the mean entropy converges. Lubkin acknowledged that he didn't know if the series converged, and apart from some qualitative arguments to say that it would, he left the question of whether or not it did unanswered. His later conclusions are therefore based on an assumption, which we will now address. We will find the condition for the series' convergence by finding a general closed-form expression for the series terms and looking at their asymptotic behaviour, and we will in fact find that the series only converges when one of the subsystems is two-dimensional, diverging rapidly otherwise (Theorem 6.1.1).

This is, however, a surprise, due to the fact that Lubkin's approximation in fact compares favourably with other expressions for the mean entropy which have been found since [24, 16, 27, 30]. Given this, we will finish by comparing the relative accuracy of these various approximations, validating Lubkin's original conclusions, but also showing that other available entropy approximations are nonetheless more accurate.

Chapter 4

Preliminaries

In this chapter we will give an overview of the basic principles of finite-dimensional quantum systems, describing the features which we will be making use of in the rest of this part and establishing various conventions of notation. We will look at the physical interpretation of such systems only in passing, and put our focus mainly on their mathematical definitions. This will be beneficial when we move on to combinatorial applications in Part III, as in most of the enumeration problems we will look at there will be no meaningful physical interpretation of the expressions being used, and attempting to keep track of such things will only confuse matters.

4.1 Quantum states

A quantum system A is described mathematically by an associated Hilbert space, which we will denote $\mathscr{H}_A$. This is a vector space, the elements of which correspond (up to scaling and phase rotation) to *pure states* of A. We will denote these *state vectors* using the bra-ket notation e.g. $|\psi\rangle$, with the corresponding Hermitian conjugate co-vector being $\langle\psi|$. By convention vectors corresponding to physical states are normalised (i.e. satisfying $\langle\psi|\psi\rangle = 1$).

Our focus here will be restricted to quantum systems of finite dimension, i.e. systems where the Hilbert space is a finite-dimensional complex vector space. For given dimension m, any such Hilbert space is necessarily isomorphic to $\mathbb{C}^m$, so without loss of generality we will equate any m-dimensional quantum system with an abstract system A_m, and will denote its Hilbert space $\mathscr{H}_m \equiv \mathbb{C}^m$. As a simple example, consider a system consisting of two q-bits, each of which can either be in a state $|0\rangle$ or $|1\rangle$ (or superpositions thereof). The Hilbert space associated with this system is spanned by the Bell states

$$\begin{aligned} |\Phi^+\rangle &= \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\Phi^-\rangle &= \tfrac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\Psi^+\rangle &= \tfrac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\Psi^-\rangle &= \tfrac{1}{\sqrt{2}}(|01\rangle - |10\rangle), \end{aligned} \tag{4.1.1}$$

so the system is equivalent to A_4.

It is not satisfactory to represent states purely in terms of state vectors, however, for two reasons. The first relates to how observable properties of quantum states are measured: an observable O is represented by a Hermitian operator $\hat{O}$, and the expectation value for a measurement of this observable for a state vector $|\psi\rangle$ is given by the inner product $\langle O\rangle_\psi = \langle\psi|\hat{O}|\psi\rangle$. Such inner products are invariant under phase rotations of the state vector, however, meaning that any two state vectors which are equivalent up to a change in phase are physically indistinguishable from each other. Ideally, representations of quantum states should correspond uniquely to their physical states, so it is for this reason that we represent pure states instead by *density operators* $\hat{\rho} = |\psi\rangle\langle\psi|$, which are invariant under phase rotation of the state vector and still contain all of the same observable information through the identity $\mathrm{Tr}[\hat{\rho}\hat{O}] = \langle\psi|\hat{O}|\psi\rangle = \langle O\rangle_\psi$.

The second reason is because a state vector can only represent pure states, and cannot represent what are called *mixed states*. What this means is best demonstrated in conjunction with the discussion of bipartite quantum systems, however, so we will introduce these next.

4.2 Bipartite systems

Consider the finite-dimensional quantum system $A_{m_1m_2}$ where[1][2] $m_2 \geq m_1 \geq 2$. The Hilbert space associated with $A_{m_1m_2}$ is $\mathscr{H}_{m_1m_2} \equiv \mathbb{C}^{m_1m_2}$, which is equivalent to the tensor product space $\mathbb{C}^{m_1} \otimes \mathbb{C}^{m_2} \equiv \mathscr{H}_{m_1} \otimes \mathscr{H}_{m_2}$. Such a system is called a *bipartite quantum system* i.e. a quantum system which can be decomposed into two distinct parts.

Physically this can be understood as follows: if the state vector $|\psi_1\rangle \in \mathscr{H}_{m_1}$ represents some pure state of the system A_{m_1}, and the state vector $|\psi_2\rangle \in \mathscr{H}_{m_2}$ represents a pure state of A_{m_2}, then the tensor product $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ is a vector in $\mathscr{H}_{m_1m_2}$, and therefore represents a state of $A_{m_1m_2}$. So a bipartite quantum system can be thought of as a physical union of the two smaller systems, with a state of the overall system including the states of both subsystems.

Not all states of $A_{m_1m_2}$ can be directly decomposed into the product of states of A_{m_1} and A_{m_2}, however. If we return to the example used in Section 4.1, the system of two q-bits is a bipartite system which can be factored into two single q-bit systems (both equivalent to A_2). But none of the four Bell states given in (4.1.1) can be factorised into a product of states of the two subsystems. Non-separable states like this are known as *entangled states*.

How do we represent the state of a subsystem in an entangled state, then? We have already said that a pure state of $A_{m_1m_2}$ may be represented by a density operator

[1] We can take $m_2 \geq m_1$ without loss of generality, as the subsystem A_{m_1} will have the same characteristics whether it's thought of as the first subsystem of $A_{m_1m_2}$ or the second subsystem of $A_{m_2m_1}$. It is therefore not important which way round the two numbers are ordered, and we choose the convention $m_2 \geq m_1$.

[2] We do not consider systems where either of the subsystems are one-dimensional. This is because a one-dimensional subsystem can always be trivially factored out of a superposition, only having one state it can exist in. As a result no entanglement is able to occur, so there is no reason to consider such a system as bipartite.

acting on $\mathscr{H}_{m_1 m_2}$ (which we will label $\hat{\rho}^{12}$). We can then modify this density operator to produce a new *reduced density operator* (RDO) which acts only on e.g. $\mathscr{H}_{m_1}$ by taking the partial trace of $\hat{\rho}^{12}$ over $\mathscr{H}_{m_2}$.

The partial trace is best understood in terms of vector bases: for the two subsystem Hilbert spaces $\mathscr{H}_{m_1}$ and $\mathscr{H}_{m_2}$, we define orthonormal bases $\{|\phi_1^1\rangle, \ldots, |\phi_{m_1}^1\rangle\}$ and $\{|\phi_1^2\rangle, \ldots, |\phi_{m_2}^2\rangle\}$. Using these, we construct an orthonormal basis set $\{|\phi_a^1\rangle \otimes |\phi_b^2\rangle : 1 \leq a \leq m_1, 1 \leq b \leq m_2\}$ spanning $\mathscr{H}_{m_1 m_2}$. In terms of this basis our density operator can be written

$$\hat{\rho}^{12} = \rho_{a_1 b_1 a_2 b_2} |\phi_{a_1}^1\rangle\langle\phi_{a_2}^1| \otimes |\phi_{b_1}^2\rangle\langle\phi_{b_2}^2|,$$

where $\rho_{a_1 b_1 a_2 b_2}$ are complex numbers, and the convention of summation over repeated indices is used (as it will be throughout). Then, taking the trace over $\mathscr{H}_{m_2}$, we get the RDO

$$\hat{\rho}_1^{12} = \rho_{a_1 b a_2 b} |\phi_{a_1}^1\rangle\langle\phi_{a_2}^1|.$$

The partial trace can be applied to completely general density operators, but, as we will only be looking at cases where the bipartite system is in a pure state, we can make some simplifications. A pure state, as stated in Section 4.1, can be represented by a normalised state vector. Using the basis given above, we can write such a vector as[3]

$$|\psi\rangle = x_{ab} |\phi_a^1\rangle \otimes |\phi_b^2\rangle,$$

where x_{ab} are $m_1 m_2$ complex numbers, normalised in the sense that $\bar{x}_{ab} x_{ab} = 1$ (where $\bar{x}_{ab}$ is the complex conjugate of x_{ab}). The density operator is once more given by $\hat{\rho}^{12} = |\psi\rangle\langle\psi|$, and if we take the partial trace we get the RDO

$$\hat{\rho}_1^{12} = x_{a_1 b} \bar{x}_{a_2 b} |\phi_{a_1}^1\rangle\langle\phi_{a_2}^1|.$$

So far we have only defined the RDO for the subsystem A_{m_1}, but we could also look at A_{m_2} if we wanted to. In this case the RDO would be $\hat{\rho}_2^{12} = x_{a b_1} \bar{x}_{a b_2} |\phi_{b_1}^2\rangle\langle\phi_{b_2}^2|$. However, we will see in Section 4.2.1 that $\hat{\rho}_1^{12}$ and $\hat{\rho}_2^{12}$ always have the same spectrum of eigenvalues when $A_{m_1 m_2}$ is in a pure state. All of the results we study here will depend only on the eigenvalues of the RDO, so it in fact doesn't matter which subsystem we study as they will give the same results. By convention we will always choose to study A_{m_1}, with $m_1 \leq m_2$ still.

4.2.1 The Schmidt decomposition

In Section 4.2 we looked at decompositions of state vectors for a bipartite quantum system, using an arbitrary orthonormal basis. While we are free to choose any such basis to work with, some bases reveal a lot more information about the nature of the states then others. One particularly informative choice of basis is the Schmidt

[3] Again there is a phase degeneracy in the state vector, as a vector with components x_{ab} will give the same density operator as one with components $e^{i\theta} x_{ab}$ for some real constant θ. This does not affect our ability to construct states in this manner, however. We will discuss the significance of this degeneracy further, and its effect (or lack thereof) on taking means over pure states, in Section 5.1.

decomposition [12, p 47], which in our notation can be described as follows:

Fact 4.2.1. *For any given state vector $|\psi\rangle$ in $\mathscr{H}_{m_1 m_2}$ (with $m_1 \leq m_2$), there exist orthonormal bases $\{|\phi_1^1\rangle, \ldots, |\phi_{m_1}^1\rangle\}$ and $\{|\phi_1^2\rangle, \ldots, |\phi_{m_2}^2\rangle\}$ such that*

$$|\psi\rangle = \sum_{a=1}^{m_1} \sqrt{p_a} |\phi_a^1\rangle \otimes |\phi_a^2\rangle, \tag{4.2.1}$$

where the $\sqrt{p_a}$ are non-negative real numbers [12].

Proof. We construct such a basis by first choosing the basis $|\phi_a^1\rangle$ such that the RDO $\hat{\rho}_1^{12}$ is diagonal (this is always possible as $\hat{\rho}_1^{12}$ is Hermitian), and then setting $|\phi_a^2\rangle$ to be the relative states of $|\phi_a^1\rangle$ i.e.

$$|\phi_a^2\rangle \propto \langle \phi_a^1 | \psi \rangle_1$$

(where $\langle \phi_a^1 | \psi \rangle_1$ represents the inner product over $\mathscr{H}_{m_1}$ only). We then choose the remaining $|\phi_a^2\rangle$ for $m_1 < a \leq m_2$ such that the entire basis is orthonormal. The coefficients $\sqrt{p_a}$ can always be made real and non-negative because we are free to choose the relative phases of $|\phi_a^1\rangle$ and $|\phi_a^2\rangle$. □

The main consequence of the Schmidt decomposition is that the two RDOs $\hat{\rho}_1^{12}$ and $\hat{\rho}_2^{12}$ will always have the same eigenvalue spectrum if the overall state is pure. If we take $\hat{\rho}^{12} = |\psi\rangle\langle\psi|$ with $|\psi\rangle$ expressed as in (4.2.1), then

$$\hat{\rho}_1^{12} = \sum_{a=1}^{m_1} p_a |\phi_a^1\rangle\langle\phi_a^1|$$
$$\hat{\rho}_2^{12} = \sum_{a=1}^{m_1} p_a |\phi_a^2\rangle\langle\phi_a^2|.$$

Each p_a is necessarily non-negative, and the normalisation of $|\psi\rangle$ ensures that their sum is unity (because $\Sigma p_a = \mathrm{Tr}[\hat{\rho}_1^{12}] = \mathrm{Tr}[\hat{\rho}^{12}] = \langle\psi|\psi\rangle = 1$). This reflects the standard interpretation of the eigenvalues of the RDO as probabilities associated with an ensemble of orthogonal states $|\phi_a^1\rangle$. In physical terms, if $A_{m_1 m_2}$ were prepared in the state $|\psi\rangle$ and the subsystem A_{m_1} subsequently observed, the probability of it being observed in the state $|\phi_a^1\rangle$ would be p_a.

The symmetry of the eigenvalues can be expressed in another way, which is particularly relevant for our work in the following chapters:

Corollary 4.2.1. *Let $\hat{\rho}_1^{12}$ and $\hat{\rho}_2^{12}$ be the reduced density matrices of the subsystems of a system $A_{m_1 m_2}$ in a pure state. Let f be a function of one variable, analytic in the neighbourhood of zero. Then*

$$Tr[f(\hat{\rho}_1^{12})] = Tr[f(\hat{\rho}_2^{12})] = \sum_{a=1}^{m_1} f(p_a).$$

Proof. Consider the monomial case $f(x) = x^d$ for some non-negative integer d. We have

$$(\hat{\rho}_1^{12})^d = \sum_{a=1}^{m_1} p_a^d |\phi_a^1\rangle\langle\phi_a^1|$$
$$(\hat{\rho}_2^{12})^d = \sum_{a=1}^{m_1} p_a^d |\phi_a^2\rangle\langle\phi_a^2|$$

due to the orthonormality of the two bases, so

$$\mathrm{Tr}[(\hat{\rho}_1^{12})^d] = \mathrm{Tr}[(\hat{\rho}_2^{12})^d] = \sum_{a=1}^{m_1} p_a^d.$$

We can then write more general f as a Maclaurin series

$$f(x) = \sum_{d=0}^{\infty} f_d x^d,$$

so

$$\mathrm{Tr}[f(\hat{\rho}_1^{12})] = \sum_{d=0}^{\infty} f_d \mathrm{Tr}[(\hat{\rho}_1^{12})^d] = \sum_{d=0}^{\infty} f_d \sum_{a=1}^{m_1} p_a^d = \sum_{a=1}^{m_1} f(p_a),$$

and the same holds for $\mathrm{Tr}[f(\hat{\rho}_2^{12})]$. □

This is a significant benefit to us, as all quantities we will be computing in the remainder of this part depend only on traces of functions of the reduced density matrices (monomial functions in most cases, in fact). Therefore, when we want to find the mean value of such a quantity, we can do so by integrating over the space of possible sets of eigenvalues, rather than over the entire matrix space. We will consider these matrix integrals more carefully in Chapter 5, looking in particular at the integral over eigenvalues in Section 5.2.

Chapter 5

Matrix integrals

Now that we have introduced all of the necessary fundamental concepts for studying bipartite quantum systems, we will in this look at expressions of the form $\langle \mathrm{Tr}[f(\hat{\rho}_1^{12})]\rangle$ i.e. mean values of the trace of the matrix expression $f(\hat{\rho}_1^{12})$, where $\hat{\rho}_1^{12}$ is the RDO for the subsystem A_{m_1} of the bipartite quantum system $A_{m_1 m_2}$, and the mean is over all pure states of $A_{m_1 m_2}$ (as before we will assume without loss of generality that $m_1 \leq m_2$).

Such means have been studied before by numerous authors [22, 24, 16, 27, 30]. Our ultimate aim in this part is to look specifically at the von Neumann entropy of entanglement, given by $\langle S_{m_1 m_2}\rangle = \langle \mathrm{Tr}[-\hat{\rho}_1^{12} \ln \hat{\rho}_1^{12}]\rangle$, first considered by Lubkin [22] (we will look at this expression specifically in Chapter 6). However, in both this part and the next we will need to be able to evaluate much more general means, so we will use this chapter to investigate the general properties of these means and the methods for evaluating them, focussing on the case when f is a monomial.

In Section 5.1 we will look at their interpretation as integrals over the space of state vectors, building on the brief discussion given by Lubkin in his paper [22]. Then, in Section 5.2, we will look at how the same integral can be given as an integral over the space of eigenvalues of $\hat{\rho}_1^{12}$, making use of work by Lloyd and Pagels [21], Page [24] and others [16, 27, 30]. In the case of the eigenvalue integral, we will get as far as giving a closed-form expression for the integral when f is a monomial.

5.1 The state vector integral

The mean $\langle \mathrm{Tr}[f(\hat{\rho}_1^{12})]\rangle$ is naturally expressed as an integral, as the space of pure states of $A_{m_1 m_2}$ is continuous. We need to take care when choosing exactly what space to integrate over, and how to weight the mean, however. Lubkin chose to integrate over the sphere of unit state vectors $|\psi\rangle$ using the invariant volume element on this sphere, and he gave a brief justification for his choice in his paper [22]. Here we will give a more thorough explanation for the validity of his choice.

Our mean $\langle \ldots \rangle$ can essentially be defined by three properties:

- *Linearity* – $\langle A + B\rangle = \langle A\rangle + \langle B\rangle$.

- *Normalisation* – $\langle 1 \rangle = 1$.
- *Unitary invariance* – This reflects an underlying symmetry of the space of states: if $\hat{\rho}^{12}$ is a the normalised density operator for a pure state of $A_{m_1 m_2}$ and $\hat{U}$ is a unitary operator acting on $\mathscr{H}_{m_1 m_2}$, then $\hat{U}\hat{\rho}^{12}\hat{U}^\dagger$ is also a normalised density operator representing a pure state with the same RDO eigenvalue spectrum. Thus we require that the mean itself be invariant under unitary transformations.

The requirement of invariance has a much clearer interpretation if the states are thought of in terms of state vectors: if we choose a vector $|\psi\rangle$ such that $\hat{\rho}^{12} = |\psi\rangle\langle\psi|$, then the transformation $\hat{\rho}^{12} \to \hat{U}\hat{\rho}^{12}\hat{U}^\dagger$ is equivalent to $|\psi\rangle \to \hat{U}|\psi\rangle$. In other words, the space of pure states is invariant under rotations of the unit sphere in $\mathscr{H}_{m_1 m_2}$.

This unit sphere is isometric with $S^{2m_1 m_2 - 1}$, the unit sphere in $\mathbb{R}^{2m_1 m_2}$; in particular, they share the same invariant volume element, which we denote $d\Omega$. The two spheres are not equivalent in all respects as their symmetry groups ($\mathrm{U}(m_1 m_2)$ and $\mathrm{O}(2m_1 m_2)$ respectively) are different, but we only require the volume element in order to construct our integral (as Lubkin argued also [22][1]), so for convenience we will refer to both as $S^{2m_1 m_2 - 1}$ from now on.

We thus find that the mean which satisfies our three properties is

$$\langle \mathrm{Tr}[f(\hat{\rho}_1^{12})] \rangle = \frac{1}{Z_{m_1 m_2}} \int_{S^{2m_1 m_2 - 1}} d\Omega \mathrm{Tr}[f(\hat{\rho}_1^{12})], \tag{5.1.1}$$

where the normalising factor

$$Z_{m_1 m_2} = \int_{S^{2m_1 m_2 - 1}} d\Omega = \frac{2\pi^{m_1 m_2}}{\Gamma(m_1 m_2)}$$

is the total "volume" of $S^{2m_1 m_2 - 1}$.

In Section 4.1 we argued that it was incorrect to represent pure states as state vectors instead of density matrices, so it may seem strange to now write the mean in terms of them. However, it is easy enough to see that (5.1.1) is equivalent to an integral over the space of pure density matrices, as the integrand is dependent only on $\hat{\rho}^{12}$. Integrating over the space of state vectors instead of the space of density matrices results in some multiple counting, but the effect this has on the weighting is uniform over all the possible states (for any state, the space of equivalent state vectors is just a unit circle), and is thus cancelled by Z_{m_1, m_2}. Given this, and the fact that (5.1.1) satisfies all three properties we require of our mean, this integral is clearly the one we want.

5.1.1 Spherical integration

(5.1.1) is a spherical integral, so this immediately presents one method for evaluation. This method is similar to one used by Folland for evaluating monomial functions over the unit sphere [15]. It should be noted that this method will not be of use for studying

[1] He referred to the volume element as the "Haar measure", by analogy with invariant volumes defined on groups.

the von Neumann entropy in Chapter 6, but we will make considerable use of it in Part III when we come to the combinatorial interpretation of these integrals.

As in Folland's method, we consider only monomials i.e. $f(\hat{\rho}_1^{12}) = (\hat{\rho}_1^{12})^d$ for some non-negative integer d. The mean is then

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = \frac{1}{Z_{m_1 m_2}} \int_{S^{2m_1 m_2 - 1}} d\Omega \mathrm{Tr}[(\hat{\rho}_1^{12})^d]. \tag{5.1.2}$$

We will find this much easier to work with, however, if we choose a basis for our vector space. As the mean is invariant under unitary rotations, we are free to choose any orthonormal basis we wish, so let us choose some separable basis $\left\{ |\phi_a^1\rangle \otimes |\phi_a^2\rangle \, : \, 1 \leq a \leq m_1, \, 1 \leq b \leq m_2 \right\}$. In this basis, any $|\psi\rangle$ is represented by a set of complex coefficients, which we denote x_{ab}, such that

$$|\psi\rangle = x_{ab} |\phi_a^1\rangle \otimes |\phi_b^1\rangle.$$

The matrix components of the RDO are then $[\hat{\rho}_1^{12}]_{a_1 a_2} = x_{a_1 b} \bar{x}_{a_2 b}$. Written out in full, (5.1.2) is

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = \frac{1}{Z_{m_1 m_2}} \int_{S^{2m_1 m_2 - 1}} d\Omega x_{a_1 b_1} \bar{x}_{a_2 b_1} x_{a_2 b_2} \bar{x}_{a_3 b_2} \cdots x_{a_d b_d} \bar{x}_{a_1 b_d}. \tag{5.1.3}$$

Now let us multiply this by

$$\frac{2}{\Gamma(m_1 m_2 + d)} \int_0^\infty \lambda^{2m_1 m_2 - 1 + 2d} e^{-\lambda^2} d\lambda = 1 \tag{5.1.4}$$

(this identity follows from the integral definition of the gamma function). The resulting integral is very cumbersome, so we won't write it out in full here, but consider what it becomes after the substitution $z_{ab} = \lambda x_{ab}$. As x_{ab} are the components of a unit vector in $\mathbb{C}^{m_1 m_2}$ and $\lambda \geq 0$, z_{ab} can be the coefficients of any arbitrary vector in $\mathbb{C}^{m_1 m_2}$, with $\lambda = |z| = \sqrt{z_{ab} \bar{z}_{ab}}$. The invariant volume element in $\mathbb{C}^{m_1 m_2}$, when expressed in Euclidean and spherical polar forms, is $d^{m_1 m_2} z d^{m_1 m_2} \bar{z} \equiv \lambda^{2m_1 m_2 - 1} d\lambda d\Omega$, so (5.1.3) $\times$ (5.1.4) is equivalent to

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = \frac{\Gamma(m_1 m_2)}{\Gamma(m_1 m_2 + d)} \pi^{-m_1 m_2} \int_{\mathbb{C}^{m_1 m_2}} d^{m_1 m_2} z d^{m_1 m_2} \bar{z} e^{-|z|^2} z_{a_1 b_1} \bar{z}_{a_2 b_1} \cdots z_{a_d b_d} \bar{z}_{a_1 b_d}. \tag{5.1.5}$$

This trick deals with the problem of having to integrate over a sphere by converting the integral into a Gaussian integral over a Euclidean complex space, which is much simpler to evaluate. The results of this evaluation are not of immediate importance here however, so we will leave the remaining steps for Chapter 10. For now we will move on to a second method for evaluating (5.1.2), by interpreting it instead as a matrix integral.

5.2 The eigenvalue integral

In Section 5.1 we wrote the mean $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle$ as an integral over state vectors, but we also noted that this is equivalent to an integral over the space of pure density operators $\hat{\rho}^{12}$. We can go a step further, however, by noting that the integrand $\mathrm{Tr}[(\hat{\rho}_1^{12})^d]$ in (5.1.1) depends only on the RDO $\hat{\rho}_1^{12}$, and, given Corollary 4.2.1, only on the eigenvalues of $\hat{\rho}_1^{12}$. Therefore, we can also express $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle$ as an eigenvalue integral over the space of RDOs. But first we need to identify what this space actually is.

Lemma 5.2.1. *Let D_m be the space of unit-trace positive semidefinite Hermitian operators acting on $\mathscr{H}_m$. If and only if $m_1 \leq m_2$, the space of possible RDOs $\hat{\rho}_1^{12}$ is D_{m_1}.*

Proof. Consider the operator $\hat{M}$ given by

$$\hat{M} = \sum_{a=1}^{m_1} p_a |\phi_a^1\rangle\langle\phi_a^1|,$$

where $\{|\phi_a^1\rangle,\, 1 \leq a \leq m_1\}$ is some orthonormal basis spanning $\mathscr{H}_{m_1}$, and the numbers p_a are all non-negative and sum to unity. $\hat{M}$ is therefore a member of D_{m_1}, and any member of D_{m_1} can be written in this form by finding its eigenvector decomposition. Furthermore, if $m_1 \leq m_2$, there is guaranteed to be a pure state of $A_{m_1 m_2}$ which has a RDO $\hat{\rho}_1^{12}$ equal to $\hat{M}$. We can always construct such a state simply by choosing an arbitrary orthonormal basis $\{|\phi_b^2\rangle,\, 1 \leq b \leq m_2\}$ spanning $\mathscr{H}_{m_2}$ and defining the state vector

$$|\psi\rangle = \sum_{a=1}^{m_1} \sqrt{p_a} |\phi_a^1\rangle \otimes |\phi_a^2\rangle.$$

Furthermore, for any such $|\psi\rangle$, its associated RDO is necessarily in D_{m_1} as it is a unit-trace positive semidefinite Hermitian operator.

This construction only works if there are at least as many basis vectors on $\mathscr{H}_{m_2}$ as there are on $\mathscr{H}_{m_1}$. Fact 4.2.1 with m_1 and m_2 swapped shows that, if $m_2 < m_1$, $\hat{\rho}_1^{12}$ can have at most m_2 non-zero eigenvalues. General $\hat{M}$ in D_{m_1} can still have up to m_1 non-zero eigenvalues, however.

Therefore, the set of RDOs $\hat{\rho}_1^{12}$ is D_{m_1} if and only if $m_1 \leq m_2$. □

This is in fact the reason why we have chosen to only use $m_1 \leq m_2$; we can only construct the eigenvalue integral when the dimensions are that way round.

Our mean thus has the form of an integral over the positive semidefinite unit-trace Hermitian operators acting on $\mathscr{H}_{m_1}$. There is already considerable work studying integrals over ensembles of Hermitian matrices in the field of matrix integration; for example, the *Gaussian Unitary Ensemble* (GUE) is an ensemble of Hermitian matrices H with dimension m (without the requirement of being unit trace and positive semidefinite), invariant under unitary transformations. If a function of H dependent only on its eigenvalues (e.g. $\mathrm{Tr}(H^d)$) is integrated over the entire GUE, the result is

an integral over the possible combinations of eigenvalues $\{p_1, \ldots, p_m\}$ of H with joint density function (JDF) [23, p 64]

$$P(p_1, \ldots, p_m) = \frac{1}{Z_{GUE(m)}} \prod_{k=1}^{m} e^{-p_k^2} \prod_{1 \leq i < j \leq m} (p_j - p_i)^2 \tag{5.2.1}$$

(where $Z_{GUE(m)}$ is a normalisation constant).

The GUE of course does not correspond exactly to the "ensemble" of RDOs, but the JDF does share some similarities. This is not overly surprising, as both ensembles have unitary invariance as a defining property.

The JDF of our ensemble is different to reflect the additional structure of our ensemble, specifically:

- *Positive semidefiniteness* - in the GUE, the eigenvalues can be either sign, but we are restricted to non-negative sign only.
- *Unit trace* - the eigenvalues must sum to unity; this will be reflected by a delta function in the JDF.
- *Multiple counting* - this arises from the fact that the RDOs are partial traces of higher-dimensional density operators, with multiple density operators potentially giving rise to the same RDO.

Fortunately, the correct JDF is already known, having been computed by Lloyd and Pagels [21] and further developed by Page [24], so to save a significant quantity of computation we will simply state it here:

$$P(p_1, \ldots, p_{m_1}) = \frac{1}{Z^*_{m_1,m_2}} \delta\left(1 - \sum_{i=1}^{m_1} p_i\right) \prod_{k=1}^{m_1} p_k^{m_2 - m_1} \prod_{1 \leq i < j \leq m_1} (p_j - p_i)^2 \tag{5.2.2}$$

for eigenvalues $p_1, \ldots, p_{m_1} \geq 0$, with the normalising factor $Z^*_{m_1 m_2}$ fixed by the condition

$$\int P(p_1, \ldots, p_{m_1}) dp_1 \cdots dp_{m_1} = 1.$$

This integral is implicitly over the space $\mathbb{R}_+^{m_1}$, where $\mathbb{R}_+$ is the non-negative real line. We will omit this designation for the sake of readability, stating the domain of integration explicitly on integrals whenever it differs.

(5.2.2) shares with (5.2.1) the product term

$$\prod_{1 \leq i < j \leq m_1} (p_j - p_i)^2 = \Delta^2(p_1, \ldots, p_{m_1}),$$

where

$$\Delta(p_1, \ldots, p_{m_1}) = \prod_{1 \leq i < j \leq m_1} (p_j - p_i) = \begin{vmatrix} 1 & p_1 & \cdots & p_1^{m_1 - 1} \\ 1 & p_2 & \cdots & p_2^{m_1 - 1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & p_{m_1} & \cdots & p_{m_1}^{m_1 - 1} \end{vmatrix}$$

is known as the *Vandermonde determinant*. The Vandermonde determinant frequently appears in JDFs for matrix ensembles, and the reason for its inclusion can be understood geometrically: the space of matrices where $p_i \neq p_j$ for any given i and j is m_1-dimensional, while the restricted space where $p_i = p_j$ is only $(m_1 - 1)$-dimensional. Therefore, the contribution to the integral of cases where any $p_i = p_j$ is vanishingly small, and so the JDF must go to zero in all such cases. The Vandermonde determinant naturally enforces this as it vanishes when any two parameters are equal.

5.2.1 Evaluation

As with the state-vector integral in Section 5.1, we need to perform some additional manipulation to make this JDF useful. Again, let us consider the mean $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle$ as an example without loss of generality. The eigenvalue integral is then

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = \frac{1}{Z^*_{m_1,m_2}} \int \delta\left(1 - \sum_{i=1}^{m_1} p_i\right) \prod_{k=1}^{m_1} p_k^{m_2 - m_1} dp_k \prod_{1 \leq i < j \leq m_1} (p_j - p_i)^2 \sum_{a=1}^{m_1} p_a^d. \tag{5.2.3}$$

In this section we will show that this expression actually has a remarkably simple (in comparison at least) closed form representation as a finite hypergeometric series.

Theorem 5.2.1. *For any non-integer $d > 0$ and integers $m_1, m_2 \geq 2$,*

$$\langle Tr[(\hat{\rho}_1^{12})^d] \rangle = \frac{\Gamma(m_1 m_2)}{d\Gamma(m_1 m_2 + d)} \sum_{r \geq 0} \frac{(-1)^r}{r!\Gamma(d - r)} \frac{\Gamma(m_1 + d - r)}{\Gamma(m_1 - r)} \frac{\Gamma(m_2 + d - r)}{\Gamma(m_2 - r)}. \tag{5.2.4}$$

Proof. Note that the theorem states this is true regardless of the order of m_1 and m_2. As in previous sections, we will begin by considering only $m_1 \leq m_2$, and note how to extend the result to all cases at the end.

The first thing we wish to do to simplify (5.2.3) is remove the delta function; the particular method for doing this is due to Page[2] [24], and is analogous to what we did in Section 5.1.1 to remove the restriction of integrating only on the unit sphere in the state-vector integral. Notice that the integrand, including the volume element and excluding the delta function, is of order $m_1 m_2 + d$ in the eigenvalues ($m_1(m_2 - m_1)$ from the $p_k^{m_2 - m_1}$ terms, m_1 from the dp_k, $m_1(m_1 - 1)$ from the Vandermonde term and d from the trace term). So, if we multiply (5.2.3) by the factor

$$\frac{1}{\Gamma(m_1 m_2 + d)} \int_0^\infty \lambda^{m_1 m_2 + d - 1} e^{-\lambda} d\lambda = 1 \tag{5.2.5}$$

(which follows from the definition of the gamma function) and use the substitution

[2] Page was actually working with the entropy integral $\langle \mathrm{Tr}[-\hat{\rho}_1^{12} \ln \hat{\rho}_1^{12}] \rangle$, but the specific co-ordinate substitution he uses is applicable here as well.

$q_i = \lambda p_i$, we get that

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle &= \frac{1}{Z^*_{m_1,m_2}\Gamma(m_1m_2+d)} \int \left[\int_0^\infty \delta\left(\lambda - \sum_{i=1}^{m_1} q_i\right) e^{-\lambda} d\lambda \right] \prod_{k=1}^{m_1} q_k^{m_2-m_1} dq_k \\ &\quad \times \prod_{1\le i<j\le m_1} (q_j - q_i)^2 \sum_{a=1}^{m_1} q_a^d \\ &= \frac{1}{Z^*_{m_1,m_2}\Gamma(m_1m_2+d)} \int \prod_{k=1}^{m_1} q_k^{m_2-m_1} e^{-q_k} dq_k \prod_{1\le i<j\le m_1} (q_j - q_i)^2 \sum_{a=1}^{m_1} q_a^d. \end{aligned} \tag{5.2.6}$$

Note that the scaling of the integrand is affected by the presence of the Dirac delta function. For example,

$$\begin{aligned}\int_0^\infty \delta(1-p) f(p) dp &= f(1) \\ &= \int_0^\infty \delta(\lambda - q) f(q/\lambda) dq \\ &= \int_0^\infty \lambda \delta(\lambda - q) f\left(\frac{q}{\lambda}\right) \frac{dq}{\lambda}.\end{aligned}$$

In the multidimensional case (with multiple p_i and q_i), this scaling can be expressed, with some abuse of notation, as

$$\delta\left(1 - \sum_{i=1}^{m_1} p_i\right) = \lambda \delta\left(\lambda - \sum_{i=1}^{m_1} q_i\right).$$

The additional λ factor this effectively produces then cancels with the λ^{-1} factor in (5.2.5).

5.2.6 bears an even closer resemblance to the GUE, except it contains an exponential term instead of a Gaussian term. This is linked to the fact that the q_i are all strictly non-negative, whereas the p_i in (5.2.1) could be any real number.

In order to evaluate this integral, we wish to make it separable. To do this, we follow the method used by Sen[3] [30]. First we move the sum over a outside the integral:

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = \frac{1}{Z^*_{m_1,m_2}\Gamma(m_1m_2+d)} \sum_{a=1}^{m_1} \int \prod_{k=1}^{m_1} q_k^{m_2-m_1} e^{-q_k} dq_k \prod_{1\le i<j\le m_1} (q_j - q_i)^2 q_a^d.$$

Due to the symmetry of the integral under exchange of eigenvalues, the m_1 different terms in this sum are all equal to each other. Therefore

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = \frac{m_1}{Z^*_{m_1,m_2}\Gamma(m_1m_2+d)} \int \prod_{k=1}^{m_1} q_k^{m_2-m_1} e^{-q_k} dq_k \prod_{1\le i<j\le m_1} (q_j - q_i)^2 q_1^d. \tag{5.2.7}$$

Next we expand out the Vandermonde term. As Sen notes, the Vandermonde determinant remains unchanged if multiples of its columns are added to each other (as

[3] As before, Sen was evaluating the entropy integral, but the method is still applicable in this case.

is true for any determinant) [30]. Therefore, if $\Lambda_k(q)$ are any set of monic polynomials of order k, then

$$\Delta(q_1,\ldots,q_{m_1}) = \begin{vmatrix} \Lambda_0(q_1) & \Lambda_1(q_1) & \cdots & \Lambda_{m_1-1}(q_1) \\ \Lambda_0(q_2) & \Lambda_1(q_2) & \cdots & \Lambda_{m_1-1}(q_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_0(q_{m_1}) & \Lambda_1(q_{m_1}) & \cdots & \Lambda_{m_1-1}(q_{m_1}) \end{vmatrix}. \tag{5.2.8}$$

We could use any such polynomials if we wished, but one choice is particularly useful; the $q_k^{m_2-m_1}e^{-q_k}$ terms in (5.2.7) are the weight functions with respect to which the associated Laguerre polynomials $L_k^{m_2-m_1}(q)$ are orthogonal. As such, if we use these polynomials, defined as

$$L_k^\alpha(q) = \frac{e^q}{q^\alpha}(-1)^k\frac{d^k}{dq^k}(e^{-q}q^{k+\alpha})$$

(their normalisation here is such that they are monic) in place of $\Lambda_k(q)$, many parts of the expansion of (5.2.7) will be zero.

These polynomials satisfy the relations

$$L_k^\alpha(q) = k!\sum_{r=0}^{k}\binom{k+\alpha}{r+\alpha}\frac{(-1)^{k-r}}{r!}q^r, \tag{5.2.9}$$

$$\int_0^\infty q^\alpha e^{-q}L_i^\alpha(q)L_j^\alpha(q)dq = i!(i+\alpha)!\delta_{ij}$$

and

$$\int_0^\infty q^{\alpha+b}e^{-q}L_k^\alpha(q)dq = \frac{b!(\alpha+b)!}{(b-k)!} \tag{5.2.10}$$

(Equations 3, 4 and 5 in [30] respectively, rewritten in our notation and with a few minor rearrangements).

We can write determinants such as (5.2.8) using the Levi-Civita symbol ε. By expanding $\Delta^2(q_1,\ldots,q_{m_1})$ like this, we get

$$\prod_{1\le i<j\le m_1}(q_j-q_i)^2 = \varepsilon_{i_1i_2\ldots i_{m_1}}\varepsilon_{j_1j_2\ldots j_{m_1}}\prod_{k=1}^{m_1}L_{i_k}^{m_2-m_1}(q_k)L_{j_k}^{m_2-m_1}(q_k)$$

(where $0 \le i_k, j_k < m_1$ for all k). Now, when we substitute this into (5.2.7), each term in the Levi-Civita summation will be a separable m_1-dimensional integral over the various q_k, where each q_k-integral collects the corresponding two Laguerre polynomials $L_{i_k}^{m_2-m_1}(q_k)L_{j_k}^{m_2-m_1}(q_k)$. Furthermore, many of these terms will be zero, as the orthogonality property causes all cases where $\{i_2,\ldots,i_{m_1}\} \neq \{j_2,\ldots,j_m\}$ to

vanish. Collecting only the remaining terms, we get

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle &= \frac{m_1\cdot(m_1-1)!}{Z^*_{m_1,m_2}\Gamma(m_1m_2+d)}\prod_{k=1}^{m_1}\int_0^\infty q^{m_2-m_1}e^{-q}[L_{k-1}^{m_2-m_1}(q)]^2dq \\ &\quad\times\sum_{i=1}^{m_1}\frac{\int_0^\infty q_1^{m_2-m_1+d}e^{-q_1}[L_{i-1}^{m_2-m_1}(q_1)]^2dq_1}{\int_0^\infty q_1^{m_2-m_1}e^{-q_1}[L_{i-1}^{m_2-m_1}(q_1)]^2dq_1} \\ &= \prod_{k=0}^{m_1-1}\frac{m_1!k!(k+m_2-m_1)!}{Z^*_{m_1,m_2}\Gamma(m_1m_2+d)}\sum_{i=0}^{m_1-1}\frac{\int_0^\infty q^{m_2-m_1+d}e^{-q}[L_i^{m_2-m_1}(q)]^2dq}{i!(i+m_2-m_1)!}\end{aligned} \tag{5.2.11}$$

The only integral which differs from the orthogonality relation in each case is the q_1 integral which contains an additional factor q_1^d, hence the inclusion of that term separately at the end, and the additional factor of $(m_1-1)!$ in the scale term accounts for all the possible orderings of the remaining (m_1-1) integrals.

This is a convenient point to set the the normalisation constant $Z^*_{m_1,m_2}$. We do so by looking at the $d=0$ case, where

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^0]\rangle = \langle m_1\rangle = m_1.$$

In this case, (5.2.11) becomes

$$\begin{aligned}m_1 &= \prod_{k=0}^{m_1-1}\frac{m_1!k!(k+m_2-m_1)!}{Z^*_{m_1,m_2}\Gamma(m_1m_2)}\sum_{i=0}^{m_1-1}1 \\ &= m_1\prod_{k=0}^{m_1-1}\frac{m_1!k!(k+m_2-m_1)!}{Z^*_{m_1,m_2}\Gamma(m_1m_2)}.\end{aligned}$$

Thus,

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle = \frac{\Gamma(m_1m_2)}{\Gamma(m_1m_2+d)}\sum_{i=0}^{m_1-1}\frac{\int_0^\infty q^{m_2-m_1+d}e^{-q}[L_i^{m_2-m_1}(q)]^2dq}{i!(i+m_2-m_1)!}.$$

We evaluate the remaining integral term

$$I_{d,\alpha,i} = \int_0^\infty q^{\alpha+d}e^{-q}[L_i^\alpha(q)]^2dq$$

by substituting in (5.2.9) and (5.2.10) to get

$$\begin{aligned}I_{d,\alpha,i} &= i!\sum_{r=0}^{i}\binom{i+\alpha}{r+\alpha}\frac{(-1)^{i-r}}{r!}\int_0^\infty q^{\alpha+d+r}e^{-r}L_i^\alpha(q)dq \\ &= i!\sum_{r=0}^{i}\binom{i+\alpha}{r+\alpha}\frac{(-1)^{i-r}}{r!}\frac{(d+r)!(\alpha+d+r)!}{(d+r-i)!},\end{aligned}$$

so

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = \frac{\Gamma(m_1 m_2)}{\Gamma(m_1 m_2 + d)} \sum_{i=0}^{m_1-1} \sum_{r=0}^{i} \frac{(-1)^{i-r}}{r!(i-r)!} \frac{(d+r)!(m_2 - m_1 + d + r)!}{(d+r-i)!(m_2 - m_1 + r)!},$$

What remains is just manipulation of summations. First, we switch the order of the summations and then replace i with $i + r$, giving

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle &= \frac{\Gamma(m_1 m_2)}{\Gamma(m_1 m_2 + d)} \sum_{r=0}^{m_1-1} \sum_{i=r}^{m_1-1} \frac{(-1)^{i-r}}{r!(i-r)!} \frac{(d+r)!(m_2 - m_1 + d + r)!}{(d+r-i)!(m_2 - m_1 + r)!} \\ &= \frac{\Gamma(m_1 m_2)}{\Gamma(m_1 m_2 + d)} \sum_{r=0}^{m_1-1} \frac{1}{r!} \frac{(d+r)!(m_2 - m_1 + d + r)!}{d!(m_2 - m_1 + r)!} \sum_{i=0}^{m_1-r-1} \binom{d}{i} (-1)^i.\end{aligned}$$

We can now remove the sum over i using the fact that

$$\sum_{i=0}^{a} \binom{d}{i} (-1)^i = \binom{d-1}{a} (-1)^a$$

(see Lemma A.0.1 in Appendix A), giving

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle &= \frac{\Gamma(m_1 m_2)}{\Gamma(m_1 m_2 + d)} \sum_{r=0}^{m_1-1} \frac{1}{r!} \frac{(d+r)!(m_2 - m_1 + d + r)!}{d!(m_2 - m_1 + r)!} \\ &\quad \times \binom{d-1}{m_1 - r - 1} (-1)^{m_1 - r - 1} \\ &= \frac{\Gamma(m_1 m_2)}{d\Gamma(m_1 m_2 + d)} \sum_{r=0}^{m_1-1} \frac{(-1)^{m_1 - r - 1}}{(m_1 - r - 1)!(d - m_1 + r)!} \\ &\quad \times \frac{(d+r)!}{r!} \frac{(m_2 - m_1 + d + r)!}{(m_2 - m_1 + r)!}.\end{aligned}$$

Finally, we replace r with $m_1 - r - 1$:

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle &= \frac{\Gamma(m_1 m_2)}{d\Gamma(m_1 m_2 + d)} \sum_{r=0}^{m_1-1} \frac{(-1)^r}{r!\Gamma(d-r)} \frac{\Gamma(m_1 + d - r)}{\Gamma(m_1 - r)} \frac{\Gamma(m_2 + d - r)}{\Gamma(m_2 - r)} \\ &= \frac{\Gamma(m_1 m_2)}{d\Gamma(m_1 m_2 + d)} \sum_{r \geq 0} \frac{(-1)^r}{r!\Gamma(d-r)} \frac{\Gamma(m_1 + d - r)}{\Gamma(m_1 - r)} \frac{\Gamma(m_2 + d - r)}{\Gamma(m_2 - r)}.\end{aligned}$$

The final step here is simply an acknowledgement that the summand is zero for any $r \geq m_1$ when d is a non-integer, meaning that we can write it equivalently as a sum over all integers r.

Making this change highlights the fact that this expression is symmetric under exchange of m_1 and m_2. From Fact 4.2.1 we also know that $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle$ has the same symmetry, so although we have only proven this identity explicitly for $m_1 \leq m_2$, it follows that it will also be true for $m_2 < m_1$. □

The above theorem only considers $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle$ for non-integers d, as in integer cases

(5.2.4) can contain indeterminate terms. Fortunately, the integer cases can be evaluated as well, by taking the limit of the non-integer case:

Corollary 5.2.1. *For positive integers d, m_1 and m_2,*

$$\langle Tr[(\hat{\rho}_1^{12})^d]\rangle = \frac{\Gamma(m_1m_2)}{d!\Gamma(m_1m_2+d)}\sum_{r\geq 0}\binom{d-1}{r}(-1)^r(m_1-r)_d(m_2-r)_d,$$

where $(a)_d = a(a+1)(a+2)\dots(a+d-1) = \Gamma(a+d)/\Gamma(a)$ is the Pochhammer symbol for the rising factorial[4] *[1, p 256].*

Proof. The limit of (5.2.4) as d tends towards a positive integer must exist, for the following two reasons:

- $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle$ is bounded for all $d > 0$, as $0 < \mathrm{Tr}[(\hat{\rho}_1^{12})^d] < m_1$.
- For known integers m_1 and m_2, $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle$ as given by (5.2.4) is a rational function of d. Therefore it must either have a well-defined limit (given by l'Hôpital's rule in indeterminate cases) or a pole at any given d.

As $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle$ is bounded for all $d > 0$, it can't have poles at positive integers d. Therefore, it must have a well-defined limit.

We need to take care with evaluating the limit, however, as the summand becomes indeterminate when d is an integer. However, if we recognise that

$$\lim_{d\to\text{integer}}\frac{\Gamma(a+d)}{\Gamma(a)} = a(a+1)\dots(a+d-1) = (a)_d$$

(the Pochhammer symbol representing the rising factorial [1, p 256]), is a polynomial in a, then we can substitute this into (5.2.4) to get the well-behaved expression

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle &= \frac{\Gamma(m_1m_2)}{d\Gamma(m_1m_2+d)}\sum_{r\geq 0}\frac{(-1)^r}{r!\Gamma(d-r)}(m_1-r)_d(m_2-r)_d \\ &= \frac{\Gamma(m_1m_2)}{d!\Gamma(m_1m_2+d)}\sum_{r\geq 0}\binom{d-1}{r}(-1)^r(m_1-r)_d(m_2-r)_d.\end{aligned}$$

□

This result is important. In particular, it allows us to evaluate $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle$ exactly as a function of m_1 and m_2 for known integer d. By extension, we can then use this to study the behaviour of more general means through Taylor expansion, as we will when we look at the entropy in the next chapter.

In addition, we will return to this result in Part III after we prove the relevance of the integral (5.1.1) in enumerative combinatorics. Once we have shown that these integrals have a combinatorial meaning, having the ability to evaluate them exactly will be of great benefit.

[4] There is some inconsistency in existing literature regarding whether this symbol is used to represent the rising factorial or the falling factorial $a!/(a-d)!$. We use it throughout to mean the rising factorial.

Chapter 6

Lubkin's entropy

We can now turn our attention to the main question of this part, concerning the mean von Neumann entropy of a random pure state of a bipartite quantum system $A_{m_1 m_2}$.

We have stated in Section 4.2 that the state of a subsystem A_{m_1} of a bipartite quantum system $A_{m_1 m_2}$ in a pure state can be represented using a reduced density operator (RDO) $\hat{\rho}_1^{12}$, and that in general this state will be mixed. In terms of this operator, the subsystem has a von Neumann entropy given by the expression [5, p 301]

$$S_1^{12} = -\mathrm{Tr}[\hat{\rho}_1^{12} \ln \hat{\rho}_1^{12}].$$

This expression is dependent only on the eigenvalue spectrum $\{p_1, \ldots, p_{m_1}\}$ of $\hat{\rho}_1^{12}$, i.e.

$$S_1^{12} = -\sum_{a=1}^{m_1} p_a \ln p_a,$$

and we have seen in Section 4.2.1 that the RDOs of A_{m_1} and A_{m_2} always have the same eigenvalue spectrum if $A_{m_1 m_2}$ is in a pure state. As a result, S_1^{12} and $S_2^{12} = -\mathrm{Tr}[\hat{\rho}_2^{12} \ln \hat{\rho}_2^{12}]$ are equal to each other. Because of this symmetry, we will tend to think of the entropy as a property of the overall system instead of just associating it with one subsystem. In this manner, the entropy is often used as a measure of the entanglement of the two subsystems [6]. For convenience we will use the symbol $S_{m_1 m_2} = S_1^{12} = S_2^{12}$ to label the entropy from now on to reflect this symmetry.

As we have said, Lubkin chose to study the mean value $\langle S_{m_1 m_2} \rangle$ of this entropy over all pure states [22]. Lubkin was unable to find a closed-form expression for the mean entropy, but he did propose a method for computing it using the Taylor series expansion

$$\langle S_{m_1 m_2} \rangle = \ln m_1 + \sum_{k=1}^{\infty} \frac{m_1^k}{k(k+1)} (-1)^k \langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^{k+1}] \rangle, \tag{6.0.1}$$

where $\hat{\rho}_1 = \frac{1}{m_1}\hat{\mathrm{I}}$, although he left the question of whether or not this series was convergent unanswered [22]. He was also unable to evaluate general terms in this series explicitly, but he suggested truncating the series at the $k = 1$ term (which he

was able to evaluate) to give the following approximation for the entropy:

$$\langle S_{m_1 m_2} \rangle \approx \ln m_1 - \frac{1}{2}\frac{m_1^2 - 1}{m_1 m_2 + 1}. \tag{6.0.2}$$

This approximation nicely supports Lubkin's arguments, as it indicates that, when $m_2 \gg m_1$, the mean entropy will be very close to $\ln m_1$, which is in fact the maximum obtainable entropy for states of $A_{m_1 m_2}$ when $m_2 \geq m_1$. His conclusions were based on the assumption that (6.0.1) is convergent, however.

In this chapter we will address this assumption, with mixed results: while we will show that (6.0.1) is in fact divergent for all $m_1 > 2$, we will also show that (6.0.2) is nonetheless a good approximation when m_2 is large. Based on this we will also argue that (6.0.1) is in fact an asymptotic expansion of the entropy.

It should be noted however that (6.0.2) is not the best known approximation for $\langle S_{m_1 m_2} \rangle$, as we will show in Section 6.2.1. As well as this, more recent work by other authors has found other methods of evaluating the entropy, both approximately and exactly [24, 16, 27, 30]. We will look at these results briefly in Section 6.2, after first looking at the series convergence problem in Section 6.1.

6.1 Series convergence

We will now determine the convergence properties of Lubkin's series (6.0.1); specifically we will show that it is convergent if and only if $m_1 = 2$. We will do this by finding a closed-form expression for the terms in the series; Lubkin was only able to calculate the first term in his paper [22], but with the results from Section 5.2.1, we now have everything we need to find a general expression for all the terms. We begin by performing some additional manipulation of said results, in order to put them in a more useful format.

Lemma 6.1.1. *For integers $d \geq 1$ and $m_1, m_2 \geq 2$,*

$$\langle Tr[(\hat{\rho}_1^{12})^d] \rangle = \frac{\Gamma(m_1 m_2) d!}{\Gamma(m_1 m_2 + d)} \sum_{r=0}^{m_1 - 1} \binom{d-1}{r} \binom{m_1}{r+1} \binom{m_2 + d - r - 1}{m_2 - 1}. \tag{6.1.1}$$

Proof. We know from Corollary 5.2.1 that

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = \frac{\Gamma(m_1 m_2)}{d! \Gamma(m_1 m_2 + d)} \sum_{r \geq 0} \binom{d-1}{r} (-1)^r (m_1 - r)_d (m_2 - r)_d. \tag{6.1.2}$$

We can rearrange this expression using the identity

$$(m_1 - r)_d = \frac{\Gamma(m_1 + d - r)}{\Gamma(m_1 - r)} = \frac{\partial^d}{\partial u^d} \frac{(-1)^d}{u^{m_1 - r}} \bigg|_{u=1}.$$

Substituting this into (6.1.2), and doing similar to $(m_2 - r)_d$, gives

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle &= \frac{\Gamma(m_1 m_2)}{d!\Gamma(m_1 m_2 + d)} \sum_{r\geq 0} \binom{d-1}{r} (-1)^r \left.\frac{\partial^d}{\partial u^d}\frac{\partial^d}{\partial v^d}\frac{1}{u^{m_1-r}v^{m_2-r}}\right|_{u,v=1} \\ &= \frac{\Gamma(m_1 m_2)}{d!\Gamma(m_1 m_2 + d)} \left.\frac{\partial^d}{\partial u^d}\frac{\partial^d}{\partial v^d}\frac{(1-uv)^{d-1}}{u^{m_1}v^{m_2}}\right|_{u,v=1}.\end{aligned}$$

We now re-expand the derivatives, starting with the v-derivative:

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle &= \frac{\Gamma(m_1 m_2)}{d!\Gamma(m_1 m_2 + d)} \left.\frac{\partial^d}{\partial u^d}\frac{1}{u^{m_1}}\sum_{r=0}^{d-1}\binom{d}{r}\frac{\partial^r}{\partial v^r}(1-uv)^{d-1}\frac{\partial^{d-r}}{\partial v^{d-r}}\frac{1}{v^{m_2}}\right|_{u,v=1} \\ &= \frac{\Gamma(m_1 m_2)}{\Gamma(m_1 m_2 + d)} \sum_{r=0}^{d-1}\binom{d-1}{r}\frac{(-1)^d}{(d-r)!}(m_2)_{d-r}\left.\frac{\partial^d}{\partial u^d}\frac{(1-u)^{d-r-1}}{u^{m_1-r}}\right|_{u=1}\end{aligned}$$

(note that we put $d-1$ as the upper limit of the summation in the product-rule expansion, because $(1-uv)^{d-1}$ differentiated d times is zero, so the final term can automatically be neglected). Now we do the same with the u-derivative, but in this case we only need to consider one term in the product-rule expansion, as the k^{th} derivative of $(1-u)^{d-r-1}$ goes to zero at $u=1$ in all cases except when $k = d-r-1$. Therefore,

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle &= \frac{\Gamma(m_1 m_2)d!}{\Gamma(m_1 m_2 + d)} \sum_{r=0}^{d-1}\binom{d-1}{r}\frac{(m_2)_{d-r}(m_1-r)_{r+1}}{(d-r)!(r+1)!} \\ &= \frac{\Gamma(m_1 m_2)d!}{\Gamma(m_1 m_2 + d)} \sum_{r=0}^{d-1}\binom{d-1}{r}\binom{m_1}{r+1}\binom{m_2+d-r-1}{m_2-1}.\end{aligned}$$

Finally, we wish to change the limit of the summation so that it's independent of d, to make it easier to insert into the binomial expansion of $\langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^k]\rangle$ later. This is easy enough; looking at the summand, we can see that it is zero if $r > d-1$ or $r > m_1 - 1$. Therefore, we get equivalently that

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle = \frac{\Gamma(m_1 m_2)d!}{\Gamma(m_1 m_2 + d)} \sum_{r=0}^{m_1-1}\binom{d-1}{r}\binom{m_1}{r+1}\binom{m_2+d-r-1}{m_2-1}.$$

□

6.1.1 Special case: $m_1 = m_2 = 2$

Before we try the general case, let us try testing for convergence of (6.0.1) in the simplest case $m_1 = m_2 = 2$. This will serve as a useful demonstration of how the general-case proof given in the next section works.

Substituting $m_1 = m_2 = 2$ into (6.1.1) we get

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle = \frac{6d!}{(d+3)!}(d^2 + d + 2).$$

We can now use this to evaluate $\langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^k] \rangle$ through its binomial expansion:

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^k] \rangle &= \sum_{d=0}^{k} \binom{k}{d} \frac{(-1)^{k-d}}{m_1^{k-d}} \langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle \\ &= \sum_{d=0}^{k} \binom{k}{d} \frac{(-1)^{k-d}}{2^{k-d}} \frac{6d!}{(d+3)!}(d^2+d+2).\end{aligned}$$

We then use the identity

$$\binom{k}{d} \frac{d!}{(d+N)!} = \frac{k!}{(k+N)!} \binom{k+N}{d+N}$$

to move the factorial terms outside the summation:

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^k] \rangle &= \frac{6k!}{(k+3)!} \sum_{d=0}^{k} \binom{k+3}{d+3} \frac{(-1)^{k-d}}{2^{k-d}}(d^2+d+2) \\ &= \frac{6k!}{(k+3)!} \sum_{d=3}^{k+3} \binom{k+3}{d} \frac{(-1)^{k+3-d}}{2^{k+3-d}}(d^2-5d+8).\end{aligned}$$

If we replace $(d^2 - 5d + 8)$ with

$$\left(\frac{\partial^2}{\partial u^2} - 4\frac{\partial}{\partial u} + 8 \right) u^d \Big|_{u=1},$$

we can then simplify this by recognising that the summation is an incomplete binomial expansion:

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^k] \rangle &= \frac{6k!}{(k+3)!} \left(\frac{\partial^2}{\partial u^2} - 4\frac{\partial}{\partial u} + 8 \right) \sum_{d=3}^{k+3} \binom{k+3}{d} \frac{(-1)^{k+3-d}}{2^{k+3-d}} u^d \Big|_{u=1} \\ &= \frac{6k!}{(k+3)!} \left(\frac{\partial^2}{\partial u^2} - 4\frac{\partial}{\partial u} + 8 \right) \left(\left(u - \frac{1}{2} \right)^{k+3} \right. \\ &\qquad \left. - \sum_{d=0}^{2} \binom{k+3}{d} \frac{(-1)^{k+3-d}}{2^{k+3-d}} u^d \right) \Big|_{u=1}.\end{aligned}$$

Note in particular that we have now removed all k-dependence in the limits of the summation. this makes the problem of determining the asymptotic behaviour for large k significantly easier.

We now expand out the remaining derivatives and set $u = 1$. As a result we get that

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^k] \rangle = \frac{3[1 + (-1)^k]}{2^k(k+3)}.$$

Substituting this into (6.0.1), we have

$$\langle S_{2,2}\rangle = \ln 2 - \sum_{k=1}^{\infty} \frac{3[1-(-1)^k]}{2k(k+1)(k+4)}.$$

This can easily be seen to converge absolutely, as

$$\left|\frac{3[1-(-1)^k]}{2k(k+1)(k+4)}\right| < \frac{3}{k^3}$$

for all $k \geq 1$, and

$$\sum_{k=1}^{\infty} \frac{3}{k^3} = 3\zeta(3)$$

where ζ is the Riemann zeta function. Furthermore, we can evaluate this series exactly (see Appendix B) to get

$$\langle S_{2,2}\rangle = \frac{1}{3}.$$

This agrees with Page's explicit formula [24] which we describe in Section 6.2.

6.1.2 The general case

Now that we have seen the method in action, we can apply it to the general case. Ultimately we will just be using the comparison test again to determine when the series converges, but in order to be able to do this we must first find an explicit expression for the terms in Lubkin's series so that we can determine their limiting behaviour. The majority of the proof will consist of the necessary manipulations required for this.

Theorem 6.1.1. *Lubkin's series (6.0.1) converges if and only if* $m_1 = 2$.

Proof. In order to compute terms in Lubkin's series, we need to know $\langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^k]\rangle$. By taking its binomial expansion and substituting in (6.1.1) we get

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^k]\rangle &= \sum_{d=0}^{k} \binom{k}{d} \frac{(-1)^{k-d}}{m_1^{k-d}} \langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle \\ &= \frac{(-1)^k}{m_1^k} a_0 + \sum_{d=1}^{k} \binom{k}{d} \frac{(-1)^{k-d}}{m_1^{k-d}} \frac{\Gamma(m_1 m_2) d!}{\Gamma(m_1 m_2 + d)} \sum_{r=0}^{m_1-1} \binom{d-1}{r} \\ &\quad \times \binom{m_1}{r+1} \binom{m_2 + d - r - 1}{m_2 - 1}\end{aligned}$$

(note also that we need to handle the d=0 term separately as (6.1.1) breaks down in this case; for this purpose we define the value $a_0 = \langle \mathrm{Tr}[(\hat{\rho}_1^{12})^0]\rangle = \min(m_1, m_2)$). Then, using the fact that

$$\binom{k}{d} \frac{d!}{(d+N)!} = \frac{k!}{(k+N)!} \binom{k+N}{d+N},$$

this becomes

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12}-\hat{\rho}_1)^k]\rangle &= \frac{(-1)^k}{m_1^k}a_0 + \frac{\Gamma(m_1m_2)k!}{\Gamma(m_1m_2+k)}\sum_{d=1}^{k}\binom{k+m_1m_2-1}{d+m_1m_2-1}\frac{(-1)^{k-d}}{m_1^{k-d}} \\ &\quad\times\sum_{r=0}^{m_1-1}\binom{d-1}{r}\binom{m_1}{r+1}\binom{m_2+d-r-1}{m_2-1} \\ &= \frac{(-1)^k}{m_1^k}a_0 + \frac{\Gamma(m_1m_2)k!}{\Gamma(m_1m_2+k)}\sum_{d=m_1m_2}^{k+m_1m_2-1}\binom{k+m_1m_2-1}{d}\frac{(-1)^{k-d+m_1m_2-1}}{m_1^{k-d+m_1m_2-1}} \\ &\quad\times\sum_{r=0}^{m_1-1}\binom{d-m_1m_2}{r}\binom{m_1}{r+1}\binom{m_2+d-m_1m_2-r}{m_2-1}.\end{aligned}$$

Now, as in Section 6.1.1, we need to replace the polynomial dependence on d in the final terms with a derivative expression. For this we use the identity

$$\frac{1}{(m_2-1)!}\frac{\partial^{m_2-1}}{\partial u^{m_2-1}}\left(\frac{u^{m_2}}{r!}\frac{\partial^r}{\partial u^r}u^{d-m_1m_2}\right)\bigg|_{u=1} = \binom{d-m_1m_2}{r}\binom{m_2+d-m_1m_2-r}{m_2-1}.$$

If we substitute this in and again note that the sum over d is an incomplete binomial expansion, we get

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12}-\hat{\rho}_1)^k]\rangle &= \frac{(-1)^k}{m_1^k}a_0 + \frac{\Gamma(m_1m_2)k!}{\Gamma(m_1m_2+k)}\frac{1}{(m_2-1)!}\frac{\partial^{m_2-1}}{\partial u^{m_2-1}}\left(u^{m_2}\sum_{r=0}^{m_1-1}\binom{m_1}{r+1}\right. \\ &\quad\times\frac{1}{r!}\frac{\partial^r}{\partial u^r}\left[u^{-m_1m_2}\left(u-\frac{1}{m_1}\right)^{k+m_1m_2-1}\right. \\ &\quad\left.\left.-\sum_{d=0}^{m_1m_2-1}\binom{k+m_1m_2-1}{d}\frac{(-1)^{k-d+m_1m_2-1}}{m_1^{k-d+m_1m_2-1}}u^{d-m_1m_2}\right]\right)\bigg|_{u=1}.\end{aligned}$$

Evaluating the derivatives at this point is a fairly tedious process, and the result is

$$\begin{aligned}\langle \mathrm{Tr}[(\hat{\rho}_1^{12}-\hat{\rho}_1)^k]\rangle &= \frac{(-1)^k}{m_1^k}a_0 + \sum_{r=0}^{m_1-1}\binom{m_1}{r+1}\frac{(-1)^r}{r!\Gamma(m_2)}\sum_{q=0}^{m_2-1}\binom{m_2-1}{q}\frac{m_2!(-1)^q}{(q+1)!} \\ &\quad\times\left[\sum_{j=0}^{q+r}\binom{q+r}{j}(-1)^j\frac{k!\Gamma(q+r+m_1m_2-j)}{\Gamma(k+m_1m_2-j)}\right. \\ &\quad\times\left(1-\frac{1}{m_1}\right)^{k+m_1m_2-j-1} - \sum_{d=0}^{m_1m_2-1}\binom{m_1m_2-1}{d} \\ &\quad\left.\times\frac{k!\Gamma(q+r+m_1m_2-d)}{\Gamma(k+m_1m_2-d)}\frac{(-1)^{k-d+m_1m_2-1}}{m_1^{k-d+m_1m_2-1}}\right].\end{aligned}$$

Note in particular the new sums over q and j, which arise from expanding derivatives of products. From this we get the exact form of the terms in Lubkin's expansion (which

we will label $T_k^{m_1,m_2}$ for simplicity), given by

$$T_k^{m_1 m_2} = \frac{(-1)^k m_1^k}{k(k+1)} \langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^{k+1}] \rangle.$$

The full expansion is very long and almost identical to the previous equation, so we don't need to write it all out again.

With this expression we have all the necessary tools to determine convergence, which we will do through the limit comparison test. The number of terms being summed over in the above expression for $T_k^{m_1 m_2}$ is fixed (i.e. independent dependent of k – specifically, there are $1 + \frac{1}{2} m_1 m_2 (m_1 + m_2 + 2 m_1 m_2)$ terms in total), so in order to determine the limiting behaviour of $T_k^{m_1 m_2}$ as k becomes large we just need to determine the limiting behaviour of all the individual terms and find the dominant ones. Each term contains a rational function of k and/or an exponential term, so we need to look for limiting behaviour for each in the form $k^\alpha A^k$. After that it will be easy to see which dominates (i.e. which has the largest A, and which has largest α if multiple terms share the largest A).

When $m_1 = 2$, all terms in the series are at most $\mathcal{O}(k^{-2})$. The summation

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$$

converges, and therefore $\langle S_{2,m_2} \rangle$ converges absolutely. When $m_1 > 2$ however, the limiting behaviour is dominated by the exponentially increasing $(m_1 - 1)^k$ term. The overall limiting term is the $(r = m_1 - 1,\, q = m_2 - 1,\, j = q + r)$ term, which gives

$$T_k^{m_1 m_2} \sim \frac{(m_1 - 1)^k}{k^{(m_1-1)(m_2-1)+2}},$$

which diverges.

Therefore, we have that Lubkin's series is convergent if $m_1 = 2$, and divergent otherwise. □

The only cases where Lubkin's series converges are the simplest cases where $m_1 = 2$ (i.e. where the system being studied is equivalent to a single q-bit), and even in this case the convergence is fairly slow. For all other cases the series diverges exponentially in the limit.

This raises a question, however: specifically why Lubkin's proposed approximation actually appears to be a good approximation, judging by how it compares to the approximation derived by Page [24] (see Section 6.2). One possible explanation is that Lubkin's series may actually be an asymptotic expansion for the entropy. We will examine this possibility in Section 6.3.

6.2 Page's formula

In Section 5.2 we demonstrated a method of calculating means over the space of pure bipartite states, using a joint density function for the eigenvalues of the reduced density operator derived by Lloyd and Pagels [21]. We used it here to find means of the trace of powers of the reduced density operator, but the authors who previously developed this method had in fact used it to evaluate the entropy itself directly [24, 30]. We will discuss their results and compare them to Lubkin's proposed approximation here, before using them to provide a now-rigorous argument for the validity of Lubkin's approximation.

Page was the first to attempt to use this method to evaluate the entropy. While he did not prove a general formula, he did manage to evaluate the special cases $m_1 = 2$ and $m_1 = 3$, giving

$$\langle S_{2,m_2} \rangle = \sum_{k=m_2+1}^{2m_2-1} \frac{1}{k} \quad \text{and} \quad \langle S_{3,m_2} \rangle = \sum_{k=m_2+1}^{3m_2} \frac{1}{k} - \frac{1}{m_2}.$$

Based on these two cases, within which some patterns can be noticed (the harmonic sum in particular), he conjectured the general formula

$$\langle S_{m_1 m_2} \rangle = \sum_{k=m_2+1}^{m_1 m_2} \frac{1}{k} - \frac{m_1 - 1}{2m_2}, \tag{6.2.1}$$

which he later found to also hold when $m_1 = 4$ and $m_1 = 5$ [24].

It was then proven a few years later by a number of authors that this formula is correct in general [16, 27, 30]. Each of these proofs used a slightly different method, but all were based on the eigenvalue integral Page began his investigation with. As we noted in Section 5.2.1, the method we used to prove Theorem 5.2.1 follows the method used by Sen [30]; it should be no surprise, then, that we are able to use our result from Theorem 5.2.1 to verify Page's formula via a parallel method (See Appendix C).

The existence of this general formula does not invalidate Lubkin's search for an approximation, however. While (5.2.6) has the distinct advantage of being closed-form (such that $\langle S_{m_1 m_2} \rangle$ can be computed exactly in finite time for given m_1 and m_2), it has the disadvantage that it is not a smooth function of m_1 or m_2, so looking at how the entropy varies as the dimensions change is not easy. (6.2.1) can be recast in a form that allows such a smooth continuation, i.e.

$$\langle S_{m_1 m_2} \rangle = \psi(m_1 m_2 + 1) - \psi(m_2 + 1) - \frac{m_1 - 1}{2m_2},$$

where $\psi(z) = \frac{d}{dz} \ln \Gamma(z)$ is the digamma function, but this is still reliant on special functions so is not ideal. Lubkin's proposed approximation has the distinct advantage of being a particularly simple function of m_1 and m_2. It also includes the term $\ln m_1$, thus relating the mean entropy visibly to the system's maximum entropy.

For this reason, Page also aimed to find an approximate form for the entropy (he

was unaware of Lubkin's prior work initially), and he showed that

$$\langle S_{m_1 m_2} \rangle \simeq \ln m_1 - \frac{m_1}{2m_2}$$

is a good approximation when $1 \ll m_1 \leq m_2$. As Page noted, this agrees with Lubkin's approximation where the two regions of validity overlap (i.e. when $1 \ll m_1 \ll m_2$) [24].

6.2.1 Validating Lubkin

We can go further, however, and gain a rigorous understanding of the validity Lubkin's approximation using Page's explicit formula. We are even able to get a clear idea of how large the error is. We do this by proving another approximation first:

Theorem 6.2.1.

$$\langle S_{m_1 m_2} \rangle = \ln m_1 - \frac{m_1^2 - 1}{2m_1 m_2} + \mathcal{O}\left(\frac{1}{m_2^2}\right) \tag{6.2.2}$$

for arbitrary m_1 and m_2.

Proof. We start by writing (6.2.1) as

$$\langle S_{m_1 m_2} \rangle = H_{m_1 m_2} - H_{m_2} - \frac{m_1 - 1}{2m_2}, \tag{6.2.3}$$

where H_n is the n^{th} harmonic number. H_n has the known asymptotic expansion

$$H_n = \ln n + \gamma + \frac{1}{2n} + \mathcal{O}\left(\frac{1}{n^2}\right),$$

where γ is the Euler-Mascheroni constant [8]. If we substitute this into (6.2.3), we get

$$\begin{aligned}\langle S_{m_1 m_2} \rangle &= \ln(m_1 m_2) - \ln m_2 + \frac{1}{2m_1 m_2} - \frac{1}{2m_2} - \frac{m_1 - 1}{2m_2} + \mathcal{O}\left(\frac{1}{m_1^2 m_2^2}\right) + \mathcal{O}\left(\frac{1}{m_2^2}\right) \\ &= \ln m_1 - \frac{m_1^2 - 1}{2m_1 m_2} + \mathcal{O}\left(\frac{1}{m_2^2}\right).\end{aligned}$$

□

In addition to the exact formula, Page mentions in the same paper an asymptotic formula for the entropy [24]. It should be noted that if we were to include further terms from the asymptotic expansion of H_n above, we would reproduce this same expansion. If we do this, including one more term, we get that the error in the approximation (6.2.2) is

$$\delta = \langle S_{m_1 m_2} \rangle - \ln m_1 + \frac{m_1^2 - 1}{2m_1 m_2} = \frac{m_1^2 - 1}{12 m_1^2 m_2^2} + \mathcal{O}\left(\frac{1}{m_2^3}\right). \tag{6.2.4}$$

This suggests that $|\delta| \leq 1/(12m_2^2)$, and numerical calculation of all cases with $m_1, m_2 \leq 100$ supports this guess, with δ apparently tending toward $1/(12m_2^2)$ as m_1 and m_2 increase (see Figure 6.2.1).

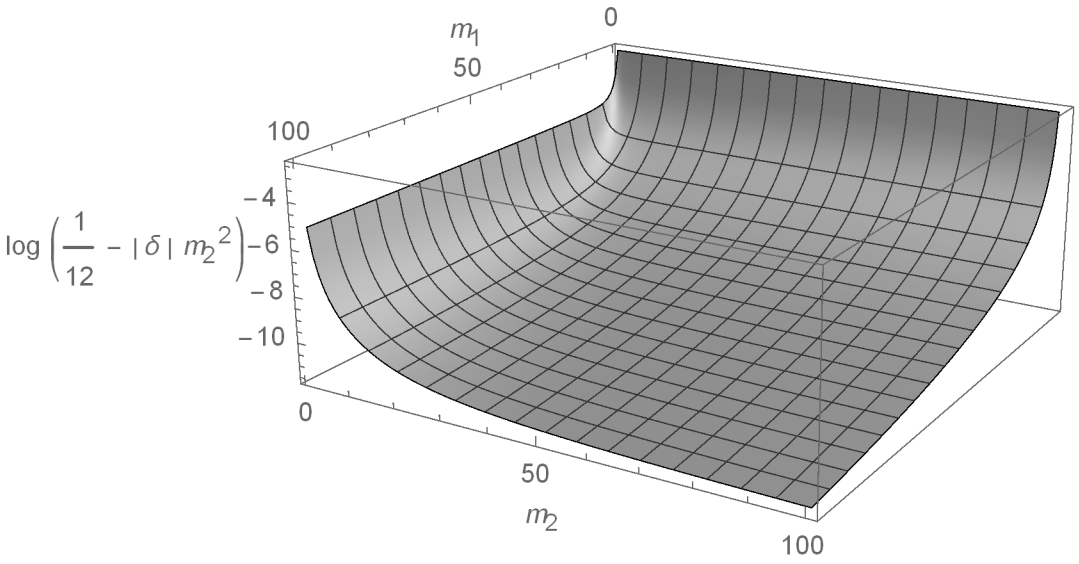


Figure 6.2.1: A graph demonstrating the closeness of the error δ associated with the entropy approximation (6.2.2) to the postulated upper bound $1/(12m_2^2)$ (see (6.2.4)). The fact that the plot is apparently monotonic decreasing in both m_1 and m_2 implies that the error tends towards $1/(12m_2^2)$ as the dimensions become large.

This approximation has a clear similarity in form to Lubkin's approximation (6.0.2), although they are not exactly the same. We can use one to prove the other, however, through a simple bit of manipulation.

Theorem 6.2.2.

$$\langle S_{m_1 m_2} \rangle = \ln m_1 - \frac{1}{2}\frac{m_1^2 - 1}{m_1 m_2 + 1} + \mathcal{O}\left(\frac{1}{m_2^2}\right)$$

for all m_1 and m_2.

Proof. From Theorem 6.2.1 we have that

$$\langle S_{m_1,m_2} \rangle = \ln m_1 - \frac{m_1^2 - 1}{2m_1 m_2} + \mathcal{O}\left(\frac{1}{m_2^2}\right).$$

We also know from the binomial expansion that

$$\frac{1}{m_1 m_2 + 1} = \frac{1}{m_1 m_2} + \mathcal{O}\left(\frac{1}{m_1^2 m_2^2}\right)$$

when $m_1, m_2 \geq 2$. Therefore, we have also that

$$\begin{aligned}\langle S_{m_1,m_2} \rangle &= \ln m_1 - \frac{1}{2}\frac{m_1^2 - 1}{m_1 m_2 + 1} + \mathcal{O}\left(\frac{m_1^2 - 1}{m_1^2 m_2^2}\right) + \mathcal{O}\left(\frac{1}{m_2^2}\right) \\ &= \ln m_1 - \frac{1}{2}\frac{m_1^2 - 1}{m_1 m_2 + 1} + \mathcal{O}\left(\frac{1}{m_2^2}\right).\end{aligned}$$

□

We therefore have confirmation of the validity of Lubkin's approximation. However, the difference between this and (6.2.2) is

$$\begin{aligned}|\Delta| &= \frac{m_1^2-1}{2m_1m_2} - \frac{1}{2}\frac{m_1^2-1}{m_1m_2+1} \\ &\approx \frac{m_1^2-1}{2m_1^2m_2^2} \\ &\approx \frac{1}{2m_2^2},\end{aligned}$$

which is approximately six times larger than the error of (6.2.2), so while Lubkin's approximation is valid according to the above proof, it is in general going to be worse than (6.2.2), potentially by a factor of six. Thus, while this method does confirm Lubkin's approximation, it does so only by first providing a better alternative.

6.3 Divergent series

The fact that Lubkin's approximation is actually valid, as shown in Theorem 6.2.2, despite being based of an in-general divergent series expansion, raises questions about the exact nature of the series (6.0.1). One possibility is that it is an asymptotic expansion of the entropy:

Definition 6.3.1. A sequence of functions $\phi_k(z)$ is an asymptotic expansion for $f(z)$ around z_0 (z_0 can be infinite) if [11]

$$\phi_{k+1}(z) = o(\phi_k(z))$$

for all k and, for any K,

$$f(z) = \sum_{k=1}^{K} \phi_k(z) + o(\phi_K(z)) \text{ as } z \to z_0, \tag{6.3.1}$$

where $u(z) = o(v(z))$ implies

$$\lim_{z\to z_0} \frac{u(z)}{v(z)} = 0.$$

In this context we are looking at the series

$$\langle S_{m_1m_2}\rangle = \ln m_1 + \sum_{k=1}^{\infty} T_k^{m_1m_2},$$

where

$$T_k^{m_1m_2} = \frac{m_1^k}{k(k+1)}(-1)^k\langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \hat{\rho}_1)^{k+1}]\rangle,$$

and there is some evidence to indicate that this is an asymptotic expansion in the limit $m_2 \to \infty$. While we have not been able to determine the general behaviour of the

$T_k^{m_1 m_2}$ in this limit, explicit computation of individual terms gives that

$$\lim_{m_2\to\infty} m_2^{\lfloor k/2\rfloor+1} T_k^{m_1 m_2} = \begin{cases} \frac{1}{2}(m_1^2-1) & k=1 \\ -\frac{1}{6m_1}(m_1^2-1)(m_1^2-4) & k=2 \\ \frac{1}{12m_1}(m_1^2-1)(2m_1^2-3) & k=3 \\ -\frac{1}{4m_1^2}(m_1^2-1)(m_1^2-2)(m_1^2-4) & k=4 \\ \frac{1}{6m_1^2}(m_1^2-1)(m_1^4-3m_1^2+3) & k=5 \\ -\frac{1}{6m_1^3}(m_1^2-1)(m_1^2-4)(3m_1^4-10m_1^2+15) & k=6 \\ \frac{1}{8m_1^3}(m_1^2-1)(2m_1^6-8m_1^4+15m_1^2-15) & k=7 \\ \quad\cdots & \end{cases}.$$

If we assume that this pattern continues for all k, we'd get that $T_k^{m_1 m_2} = \mathcal{O}(m_2^{-\lfloor k/2\rfloor-1})$ and $T_k^{m_1 m_2} = o(m_2^{-\lfloor k/2\rfloor})$. The sequence $T_k^{m_1,m_2}$ would thus not immediately be asymptotic, but if consecutive terms with the same order in m_2 (e.g. $T_2^{m_1 m_2}$ and $T_3^{m_1 m_2}$) were paired together, then it would be, as e.g. $T_6^{m_1 m_2} + T_7^{m_1 m_2} = o(T_4^{m_1 m_2} + T_5^{m_1 m_2})$.

This would still not prove that (6.0.1) is specifically an asymptotic expansion of the entropy, however. This would require showing that all possible truncations of the series tend asymptotically to the entropy according to (6.3.1). Again, there is some evidence to support this. In addition to the fact that the series was derived directly from the entropy, we also have the fact that the series is *Borel summable*, and its Borel sum is exactly the entropy. Borel summation is a method of assigning values to infinite series which is guaranteed to agree with the correct value for convergent series, but is also able to give finite values in some cases where the series diverges:

Theorem 6.3.1. *The Borel sum of the series (6.0.1) exists and equals the entropy* $\langle S_{m_1 m_2}\rangle$.

Proof. We start by defining the formal power series

$$\langle S_{m_1 m_2}(w)\rangle = \ln m_1 + \sum_{k=1}^{\infty} \frac{m_1^k w^{k-1}}{k(k+1)}(-1)^k \langle \mathrm{Tr}[(\hat\rho_1^{12} - \hat\rho_1)^{k+1}]\rangle,$$

which is equivalent to Lubkin's series when $w = 1$. The Borel transform of this (given by replacing w^k with $t^k/k!$) is

$$\begin{aligned} \langle \mathcal{B}S_{m_1 m_2}(t)\rangle &= \ln m_1 + \sum_{k=1}^{\infty} \frac{m_1^k t^{k-1}}{(k+1)!}(-1)^k \langle \mathrm{Tr}[(\hat\rho_1^{12} - \hat\rho_1)^{k+1}]\rangle \\ &= \ln m_1 - \frac{\langle \mathrm{Tr}[\exp\{-m_1 t(\hat\rho_1^{12} - \hat\rho_1)\} - \hat{\mathrm{I}} + m_1 t(\hat\rho_1^{12} - \hat\rho_1)]\rangle}{m_1 t^2} \\ &= \ln m_1 - \frac{e^t \langle \mathrm{Tr}[\exp(-m_1 t \hat\rho_1^{12})]\rangle - m_1}{m_1 t^2}. \end{aligned}$$

In terms of the Borel transform, the Borel sum of (6.0.1) (which we evaluate through

integration by parts) is

$$\begin{aligned}\int_0^\infty e^{-t}\langle \mathcal{B}S_{m_1m_2}(t)\rangle dt &= \ln m_1 - \frac{1}{m_1}\int_0^\infty (\langle \mathrm{Tr}[\exp(-m_1 t\hat{\rho}_1^{12})]\rangle - m_1 e^{-t})\frac{dt}{t^2} \\ &= \ln m_1 + \frac{1}{m_1}\int_0^\infty \frac{d^2}{dt^2}(\langle \mathrm{Tr}[\exp(-m_1 t\hat{\rho}_1^{12})]\rangle - m_1 e^{-t})\ln t dt \\ &= \ln m_1 + \frac{1}{m_1}\int_0^\infty (m_1^2\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^2\exp(-m_1 t\hat{\rho}_1^{12})]\rangle - m_1 e^{-t})\ln t dt.\end{aligned}$$

This integral will converge, as can be seen by writing the mean out in terms of the eigenvalues of $\hat{\rho}_1^{12}$, which are necessarily non-negative. Using the fact that

$$\int_0^\infty a e^{-at}\ln t dt = -\gamma - \ln a$$

where γ is the Euler-Mascheroni constant (see Lemma A.0.2 in Appendix A), we get

$$\int_0^\infty e^{-t}\langle \mathcal{B}S_{m_1m_2}(t)\rangle dt = \ln m_1 - \langle \mathrm{Tr}[\hat{\rho}_1^{12}(\gamma + \ln(m_1\hat{\rho}_1^{12})]\rangle + \gamma.$$

Most terms cancel, due to the fact that

$$\langle \mathrm{Tr}(\hat{\rho}_1^{12})\rangle = 1,$$

and all that remains is

$$\int_0^\infty e^{-t}\langle \mathcal{B}S_{m_1m_2}(t)\rangle dt = -\langle \mathrm{Tr}(\hat{\rho}_1^{12}\ln\hat{\rho}_1^{12})\rangle = \langle S_{m_1m_2}\rangle.$$

□

This provides a second meaningful identification between (6.0.1) and the entropy, supporting the assertion that it is an asymptotic expansion of the entropy, although this does not constitute a proof in itself.

6.3.1 The reason for divergence, and convergent alternatives

The Borel summation process used above provides some additional insight into the reasons why Lubkin's series converges if and only if $m_1 = 2$. Consider the mean entropy as an integral with integrand

$$-\sum_{k=1}^{m_1} p_k \ln p_k,$$

where p_k are again the eigenvalues of $\hat{\rho}_1^{12}$. Lubkin's method involved expanding each term in this integrand as a Taylor series around $p_k = 1/m_1$, and the resulting infinite series is guaranteed to converge if all p_k lie within radius $1/m_1$ of $1/m_1$ (due to the branch points that appear when any $p_k = 0$). Given that all the p_k are real, we can

equivalently state the domain of convergence as[1] $0 \leq p_k \leq 2/m_1 \,\forall\, k$.

In the case $m_1 = 2$, the domain of convergence is $0 \leq p_k \leq 1$, which covers the entire domain of the integral. This allows Lubkin's series to converge in these cases. When $m_1 > 2$, however, the domain of the integral starts to include regions where the summation diverges, so the convergence of the resulting series is no longer guaranteed. Lubkin's hypothesis was based on the idea that the divergences may in a sense cancel each other out in these cases. We have now seen that this is not the case, however.

Borel summation is able to circumvent this problem, though. Borel summation works by taking an expression which is convergent on some finite interval and extending its domain of convergence to the entire positive real line. Applying Borel summation to our integrand here therefore gives an expression which is convergent within the entire domain of the integral. The result, however, tells us nothing new, as the result we get back is merely the defining expression of the entropy that we began with.

This analysis does suggest the possibility of alternative series expansions, however, which may be constructed in such a way as to ensure convergence. We simply need to expand around some point $\alpha \geq 1/2$, giving

$$\langle S_{m_1 m_2} \rangle = m_1 \alpha - \ln \alpha - 1 + \sum_{k=1}^{\infty} \frac{(-1)^k}{k(k+1)\alpha^k} \langle \mathrm{Tr}[(\hat{\rho}_1^{12} - \alpha \hat{\mathrm{I}})^{k+1}] \rangle. \tag{6.3.2}$$

This series of course has a very similar form to (6.0.1), and the two become equivalent if we set $\alpha = 1/m_1$.

While we have defined this series to have the useful property of being convergent, it should be noted that it is not useful as an approximation. Lubkin and Page's approximations have the distinct advantage that their zeroth-order terms are $\ln m_1$, and are thus asymptotic to the entropy at large dimensions. Our constant α is defined independently of the two dimensions, however, giving a zeroth-order asymptotic behaviour which does not at all resemble that of the entropy. So while we conjectured in Section 6.3 that (6.0.1) may be an asymptotic expansion, it is clear that (6.3.2) is not. We show it here purely as a demonstration of the principles behind the convergence properties of (6.0.1); it is of little use in the actual evaluation of the entropy, given the other available options that we have already considered.

[1] We give a closed interval here because the entropy has a well-defined limit at both ends of the interval.

Chapter 7

Conclusions

In summary, we have proven that Lubkin's series expansion for the mean von Neumann entropy of a finite-dimensional bipartite quantum system in a pure state [22] is divergent when the dimension of the subsystem being studied is greater than 2. We determined this by finding closed-form expressions for the individual series terms, from which their asymptotic behaviour could be derived. This showed that the convergence was dependent on m_1, the dimension of the subsystem being observed, with the series converging slowly (as k^{-3}) when $m_1 = 2$, and diverging exponentially otherwise.

We also showed that the approximate formula Lubkin proposed was, in spite of having been derived from this series, a valid approximation of the entropy for large dimensions. This implies that the series expansion may be an asymptotic expansion of the entropy, though we don't know if this is true to all orders.

It must also be noted that, while we proved the validity of Lubkin's approximation, we at the same time showed that better approximations exist, such as (6.2.2), which is both a simpler expression and numerically closer to the correct value. In fact we were only able to establish the validity of Lubkin's approximation by comparing it to this one.

The more important results from this part, however, lie in the methods we used to prove the above results, in particular the matrix-integral methods we used to evaluate the expressions $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle$ in closed form. These expressions, and in a broader sense the methods themselves, have a lot of significance in enumerative combinatorics, as we will see in the next part. Specifically they can be used for evaluating generating functions for enumerating rooted hypermaps, and related expressions can then be used for counting rooted maps and constellations. This will be the focus of the next part, so we will leave detailed discussion of these results until then. In the next chapter, however, we will begin by taking a clear look at $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle$ in particular, to see how their properties originally led to the discovery of their combinatorial meaning.

Part III

Enumeration of rooted constellations

Chapter 8

$\langle \mathbf{Tr}[(\hat{\rho}_1^{12})^d] \rangle$ as combinatorial functions

From this point onwards we will leave quantum theory behind, moving instead into the field of enumerative combinatorics. We have discussed how it is already known that these two fields overlap with each other in places. We know that, in some cases, mean properties of quantum systems can be expressed as summations over combinatorial objects [33, 14, 13, 25]. We also know that, conversely, quantum expressions can be used as generating functions for enumerating combinatorial objects [7, 39]. Now we will look at how these principles apply to our expressions from Part II.

Let us take a closer look at these expressions, specifically

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = \frac{\Gamma(m_1 m_2)}{d!\Gamma(m_1 m_2 + d)} \sum_{r \geq 0} \binom{d-1}{r} (-1)^r (m_1 - r)_d (m_2 - r)_d,$$

defined for positive integers d (see Corollary 5.2.1). For any given d, this is a function of m_1 and m_2 only, and by inspection we see that it is a rational function of m_1 and m_2, with a denominator of the form[1]

$$\frac{\Gamma(m_1 m_2 + d)}{\Gamma(m_1 m_2)} = m_1 m_2 (m_1 m_2 + 1) \cdots (m_1 m_2 + d - 1).$$

The numerator is then everything else, and we define the functions

$$P_d(m_1, m_2) = \frac{\Gamma(m_1 m_2 + d)}{\Gamma(m_1 m_2)} \langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle \tag{8.0.1}$$

$$= \frac{1}{d!} \sum_{r \geq 0} \binom{d-1}{r} (-1)^r (m_1 - r)_d (m_2 - r)_d \tag{8.0.2}$$

to be these numerators. While it is not immediately obvious from this definition, especially given the $1/d!$ factor, these functions are polynomials with integer coefficients

[1] Strictly speaking this is not actually the simplest form the of the denominator in general, as the numerator will turn out to be a multiple of $m_1 m_2$ as well. However, when we proceed to look at these expressions as generating functions, it will be important to leave the $m_1 m_2$ factor in the numerator, so we choose not to factor it out of the denominator for that reason.

(the fact that the coefficients are integers can be inferred from Theorem 10.2.1), and if we evaluate the first few cases explicitly (note that we can do this easily as only the first d terms in the summation are non-zero) we get

$$\begin{aligned}
P_1(m_1,m_2) &= m_1m_2\\
P_2(m_1,m_2) &= m_1m_2^2 + m_1^2m_2\\
P_3(m_1,m_2) &= (m_1m_2^3 + 3m_1^2m_2^2 + m_1^3m_2) + m_1m_2\\
P_4(m_1,m_2) &= (m_1m_2^4 + 6m_1^2m_2^3 + 6m_1^3m_2^2 + m_1^4m_2^2) + (5m_1m_2^2 + 5m_1^2m_2)\\
P_5(m_1,m_2) &= (m_1m_2^5 + 10m_1^2m_2^4 + 20m_1^3m_2^3 + 10m_1^4m_2^2 + m_1^5m_2)\\
&\qquad + (15m_1m_2^3 + 40m_1^2m_2^2 + 15m_1^3m_2) + 8m_1m_2\\
&\vdots
\end{aligned}$$

Although no general form is forthcoming beyond the existing summation form, we can already note a number of general trends. For instance, $P_d(m_1,m_2)$ appears to be of order $(d+1)$, odd[2] when d is even and vice versa. Also, there are some noticeable patterns in the individual terms such as the coefficient of $m_1^2m_2^{d-1}$ appearing to be $\frac{1}{2}d(d-1)$. One of the most telling, however, is that if we set $m_1 = m_2 = 1$ (i.e. if we sum over all the coefficients in each function), we get $P_d(1,1) = d!$. This is particularly indicative of some combinatorial meaning.

Another interesting fact about these functions, again which hints at a combinatorial interpretation, is that they satisfy a recursion relation. This arises from the fact that (8.0.2) is a hypergeometric series with a finite number of non-zero terms, and it is known that a fixed-order recursion relation will always exist for such a series [26, p 64].

Theorem 8.0.2. *The sequence of functions $P_d(m_1,m_2)$ satisfy*

$$\begin{aligned}
(d+3)P_{d+2}(m_1,m_2) = (2d+3)(m_1+m_2)P_{d+1}(m_1,m_2)\\
+ d[(d+1)^2 - (m_1-m_2)^2]P_d(m_1,m_2)
\end{aligned} \tag{8.0.3}$$

for all $d \geq 1$.

Proof. Let us rewrite (8.0.2) in the form

$$P_d(m_1,m_2) = \sum_{r\geq 0} F(d,r)$$

where

$$F(d,r) = \frac{1}{d!}\binom{d-1}{r}(-1)^r(m_1-r)_d(m_2-r)_d.$$

We have omitted the parameters m_1 and m_2 here purely for the sake of readability and

[2]Odd in the sense that, if the signs of all the parameters are changed, the overall sign of the function changes i.e. $P_d(-m_1,-m_2) = -P_d(m_1,m_2)$. Similarly, $P_d(-m_1,-m_2) = P_d(m_1,m_2)$ for an even function.

consistency with [26]. This function satisfies the relation

$$G(d,r+1) - G(d,r) = (d+3)F(d+2,r) - (2d+3)(m_1+m_2)F(d+1,r) \\ - d[(d+1)^2 - (m_1-m_2)^2]F(d,r) \tag{8.0.4}$$

where

$$\frac{G(d,r)}{F(d,r)} = \frac{d!(d-r-1)!dr(m_1+d-r)(m_2+d-r)}{(d+2)!(d-r+1)!}[2d^3 - 3d^2r - dm_1m_2 + dr^2 + 7d^2 \\ - dm_1 - dm_2 - 7dr - 3m_1m_2 + m_1r + m_2r + r^2 + 8d - m - n - 4r + 3].$$

This fact was derived using Zeilberger's algorithm [26, 38], and can be verified through substitution and cancellation.

We then sum (8.0.4) over all $r \geq 0$. On the left hand side the F terms sum give $P_d(m_1,m_2)$ etc. while the right hand side telescopes out to $\lim_{r\to\infty} G(d,r) - G(d,0)$. But $G(d,0) = 0$, and $G(d,r) = 0$ for all $r > (d+1)$, so the right hand side is zero in the limit. Therefore, the result of this summation is that

$$(d+3)P_{d+2}(m_1,m_2) = (2d+3)(m_1+m_2)P_{d+1}(m_1,m_2) \\ + d[(d+1)^2 - (m_1-m_2)^2]P_d(m_1,m_2).$$

□

This recursion relation, along with the initial cases $P_1(m_1,m_2) = m_1m_2$ and $P_2(m_1,m_2) = m_1m_2(m_1+m_2)$, give us all of the functions in the sequence. In addition, it allows us to prove some of the properties which we guessed previously:

Corollary 8.0.1. *For all $d \geq 1$, $P_d(m_1,m_2)$ is a polynomial of order $(d+1)$, which is odd if d is even and even if d is odd.*

Proof. Assume this holds up to $d = \delta$, for some $\delta \geq 2$. (8.0.3) then gives that

$$P_{\delta+1}(m_1,m_2) = \frac{2\delta+1}{\delta+2}(m_1+m_2)P_\delta(m_1,m_2) + (\delta-1)\frac{\delta^2-(m_1-m_2)^2}{\delta+2}P_{\delta-1}(m_1,m_2).$$

The right hand side is then a polynomial of order $\delta+2$, as expected, and it is even if and only if $(\delta+1)$ is odd. We know that $P_d(m_1,m_2)$ has the desired properties for $d \leq 2$, so it follows by induction that it is true for all $d \geq 1$. □

Corollary 8.0.2. $P_d(1,1) = d!$ *for all $d \geq 1$.*

Proof. Again, assume this is true up to $d = \delta$ for some $\delta \geq 2$. When $m_1 = m_2 = 1$,

(8.0.3) becomes

$$\begin{aligned} P_{\delta+1}(1,1) &= 2\frac{2\delta+1}{\delta+2}P_\delta(1,1) + \frac{\delta^2}{\delta+2}P_{\delta-1}(1,1) \\ &= 2\frac{2\delta+1}{\delta+2}\delta! + \frac{\delta^2(\delta-1)}{\delta+2}(\delta-1)! \\ &= (\delta+1)!. \end{aligned}$$

It is easy to see that it's true for $d \leq 2$, so it is true for all $d \geq 1$ by induction. □

All of this hints at some deeper meaning to these functions, but none of it says what this meaning actually is. We can identify this meaning, however, by evaluating the coefficients at various orders (see Appendix D) and comparing them to prior enumeration work. As it turns out, all the coefficients up to $d = 12$ can be found in Appendix B of [35], which gives the results of enumerating rooted hypermaps. By comparing these two sets of results we find that $P_d(m_1, m_2)$ *is a generating function enumerating rooted hypermaps with* d *darts and one face, partitioning them by number of edges and vertices.*

While we can state this fairly confidently in light of the agreement over the first twelve orders, this is far from a proof. We do have all the necessary tools to prove it, however, as we will in Chapter 10. We will then examine these tools further in the following chapters, seeing how they can be used to enumerate all rooted hypermaps (Chapter 11) as well as rooted maps and constellations (Chapter 12).

First, however, we need to properly introduce and define these various terms. This is the purpose of Chapter 9.

Chapter 9

Basic concepts

In this chapter we will give an introduction to a number of combinatorial concepts which will see heavy use in the following chapters, in particular constellations, hypermaps and generating functions. This chapter is purely an explanation of existing concepts and does not introduce any new results, although it does serve to highlight which properties/interpretations will be most relevant to the following work. For instance, in the case of hypermaps, which have multiple equivalent interpretations, we will focus on interpreting them as 3-constellations, as opposed to the geometric interpretation of diagrams on orientable surfaces, as the former is much more useful for our purposes.

9.1 Hypermaps and constellations

9.1.1 Permutations

One of the fundamental concepts of combinatorics is the the *permutation*, i.e. a bijective function mapping from a finite ordered set to itself. In this context we will define a permutation as follows:

Definition 9.1.1. A d-permutation is a bijective function from some set D with d elements to itself. We will call d the permutation's *degree*, for consistency with the terminology we use for hypermaps.

Definition 9.1.2. The symmetric group Sym_d is the group consisting of all permutations acting on the set $[1 \dots d]$.

We have defined permutations here as acting on arbitrary sets. However, the group of all permutations acting on any given set of size d is isomorphic to Sym_d, so in many cases we will identify d-permutations as simply being those acting on $[1 \dots d]$ for simplicity. We will encounter permutations acting on other defined sets later, though, where hypermaps involve permutations acting on the set of *darts* in the hypermap, so it is helpful to bear this more general definition in mind.

We will represent permutations here using cycle notation. For example, the 5-permutation $g = (143)(25)$ maps 1 to 4, 4 to 3, 3 to 1, and 2 and 5 to each other. The two bracketed sets of values are known as *cycles* (so g has two cycles).

Another way of representing this permutation would be

$$g : \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \to \begin{bmatrix} 4 \\ 5 \\ 1 \\ 3 \\ 2 \end{bmatrix}.$$

While many different equivalent representations for permutations exist, the cycle representation will be of most use to us as it is the cycles of each permutation which we are interested in; when we look at hypermaps, we will see that cycles correspond to geometric features of the hypermaps, specifically edges, vertices and faces. We denote the number of cycles in a permutation g as $\mathrm{cyc}(g)$.

Permutations can be composed together to produce permutations of higher degree through the *direct sum* $\oplus$:

Definition 9.1.3. Given permutations $\{g_1, g_2, \ldots, g_N\}$ acting respectively on the disjoint sets $\{D_1, D_2, \ldots, D_N\}$, their direct sum $g = g_1 \oplus g_2 \oplus \ldots \oplus g_N$ is defined on $D_1 \cup D_2 \cup \cdots \cup D_N$ such that

$$g(x) = \begin{cases} g_1(x), & x \in D_1 \\ g_2(x), & x \in D_2 \\ \vdots & \\ g_N(x), & x \in D_N \end{cases}.$$

The direct sum provides a useful method of decomposing permutations into subpermutations. The direct sum preserves cycles in each of the subpermutations, so the maximum number of parts any given permutation can be decomposed into is equal to the number of cycles it contains e.g. $(136)(25)(4) = (136) \oplus (25) \oplus (4)$. Thus the only permutations which can't be decomposed are one-cycles. We will revisit direct sums and decompositions when looking at constellations, as they provide a simple way of understanding the *transitivity* property of constellations (see the next section), which is very useful for understanding the process of computing generating functions of general hypermaps in Chapter 11.

9.1.2 Constellations

Definition 9.1.4. A k-constellation acting on a set D of size d is a sequence $[g_1, \ldots, g_k]$ of d-permutations with the following two properties:

- *Transitivity*: The group generated by $[g_1, \ldots, g_k]$ acts transitively on the set D,

- *Closure*: The product $g_1 g_2 \cdots g_k$ equals the identity.

The number k is the constellation's *length*, and d its *degree*. [20, p 8].

At its simplest, a constellation is an ordered set of permutations. As is seen in the above definition, however, some additional structure is required of the permutations. These properties, which we refer to here as transitivity and closure, have the interesting result of making 1- and 2-constellations trivial, as any 2-constellation is just $[g, g^{-1}]$ for some 1-cycle g, while there is only one 1-constellation, $[\mathrm{I}_1]$ where $\mathrm{I}_1 = (1)$ is the degree-one identity permutation. [20, p 9]. Thus we will only be studying constellations of length three or more here, with the case $k = 3$ being equivalent to hypermaps, as we shall see later.

The transitivity property in particular imposes a sort of connectedness on a hypermap, by requiring that any element of D can be mapped to any other element of D just through sequential application of the permutations $g_1, \ldots, g_k$. If a sequence of permutations did not satisfy this property, then it would be possible to decompose it into a direct sum of constellations of lower degree i.e.

$$\begin{bmatrix} (123)(4)(5) \\ (13)(2)(45) \\ (1)(23)(45) \end{bmatrix} = \begin{bmatrix} (123) \\ (13)(2) \\ (1)(23) \end{bmatrix} \oplus \begin{bmatrix} (4)(5) \\ (45) \\ (45) \end{bmatrix}.$$

Thus, in the same way that 1-cycles are permutations which cannot be decomposed into a direct sum of smaller permutations, a constellation is a closed sequence of permutations which cannot be decomposed into a direct sum of sequences of lower degree.

While any sequence of permutations satisfying these properties is a constellation, some constellations with different representations are isomorphic to each other. This is the case when two permutations have the same structure except for a relabelling of the elements in the set the permutations act on. For instance,

$$\begin{bmatrix} (123)(4)(5) \\ (13)(2)(45) \\ (1)(23)(45) \end{bmatrix} \text{ and } \begin{bmatrix} (143)(2)(5) \\ (13)(4)(25) \\ (1)(43)(25) \end{bmatrix}$$

are isomorphic, the only difference being the swapping of the labels 2 and 4. We can express this more formally:

Definition 9.1.5. Two k-constellations of degree d, $[g_1, \ldots, g_k]$ and $[h_1, \ldots, h_k]$, acting respectively on sets D and D', are *isomorphic* if there exists a bijection $p : D \to D'$ such that [20, p 9]

$$[pg_1p^{-1}, \ldots, pg_kp^{-1}] = [h_1, \ldots, h_k].$$

In the enumerative work of the following chapters we will be concerned specifically with enumeration of *isomorphism classes* of constellations, rather than enumeration of

distinct representations. This has been the general practice of previous authors as well [34, 3, 35], as typically it is only the structure which is important, so counting cases which behave in exactly the same way as each other is redundant.

9.1.3 Rooted constellations

We will not be looking at isomorphism classes under the very general type of isomorphisms given in Definition 9.1.5, however, but at a more restricted type of isomorphism class called a *rooted constellation*. We define these as follows:

Definition 9.1.6. A rooted k-constellation C of degree d is an isomorphism class of k-constellations acting a given set D such that, for any two constellations c_1 and c_2 in C there exists a permutation p which maps c_1 to c_2 and preserves a specified element x of D, called the *root* .

So, as a given set D has a set of constellation isomorphism classes associated with it, a pair (D, x) consisting of a set D and a single distinguished element $x \in D$ is associated with a set of rooted constellation equivalence classes. The isomorphism for general constellations imposes the idea that all the elements of D are equivalent to each other, so exchanging them doesn't change the structure of the constellation. In the rooted case x is now considered distinct from the other elements, so any reordering of the elements only preserves the rooted constellation's structure if the root is preserved (i.e. if $p(x) = x$).

When we look at rooted k-constellations acting on $[1, \ldots, d]$, we will by convention take the root to be $x = 1$.

9.1.4 Hypermaps

While we will eventually proceed to derive enumeration results for rooted constellations of arbitrary length, the focus of most of this work will be on the simplest non-trivial case, 3-constellations, which are equivalent to *hypermaps*:

Definition 9.1.7. A hypermap is a 2-cell embedding of a hypergraph on an orientable surface O; a hypergraph (V, E) consists of a set V of vertices and a family E of families of vertices from V, called edges. A 2-cell embedding means that the complement of the hypergraph O on the surface is a disjoint union of regions homeomorphic to the open disc (see Figures 9.1.1 and 9.1.2 for a visualisation of the process of embedding a hypergraph in a surface).

To understand how a hypergraph may be embedded in a surface to produce a hypermap, it is helpful to consider the hypergraph as a set of points representing the vertices, with the edges a number of groupings of these vertices (see Figure 9.1.1a). Note that edges are specifically *families* so that an edge can contain particular vertices

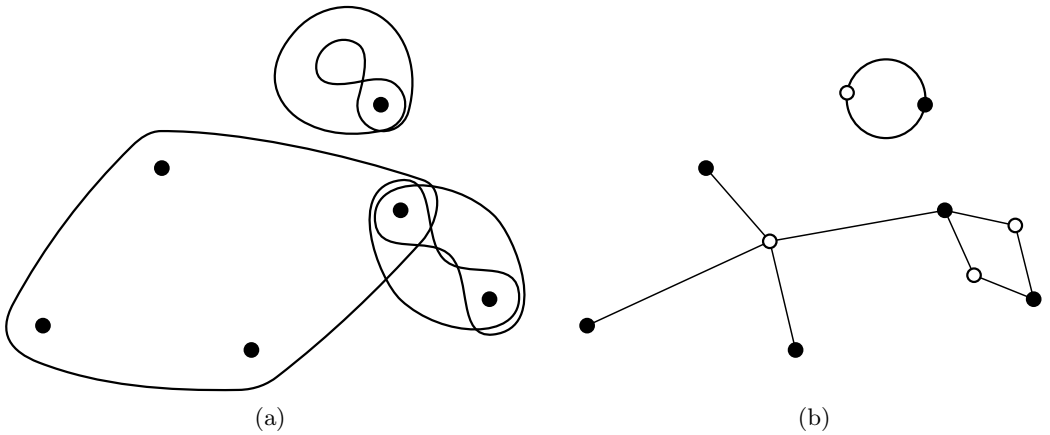


Figure 9.1.1: (a) A hypergraph and (b) the equivalent bipartite graph representation, showing how edges can contain vertices more than once, and how multiple edges can be equivalent. This hypergraph cannot form a hypermap as it is disjoint, so cannot form a 2-cell embedding.

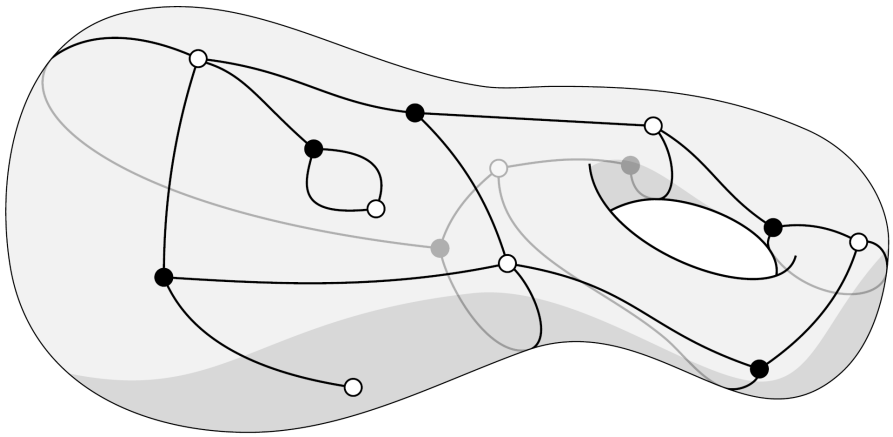


Figure 9.1.2: A hypermap with seven vertices, seven edges, six faces and twenty darts, embedded in a genus-one surface. This satisfies Euler's characteristic equation, which for a hypermap is $v + e + f - d = 2 - 2g$.

multiple times. Similarly E is a family such that the hypergraph can contain multiple edges with the same connectivity. These are a generalisation of the concept of graphs (vertices connected pairwise by lines, called edges) where the edges are no longer restricted to connecting to two vertices, but can instead connect to any positive number of vertices.

To understand the embedding process, however, it is helpful to use a particular representation of hypergraphs as bipartite bicoloured graphs (see Figure 9.1.1b). Here we replace each edge by a new point, labelled in white, and then connect it to each of the vertices it contains by a line, which we refer to as a *dart* [36]. If an edge contains a vertex more than once, then they are connected by more than one dart.

Given this, the process of embedding a hypergraph in an orientable surface is equivalent to the process of embedding a bipartite graph. Embedding a graph produces a map, so we say that embedding a hypergraph produces a hypermap. An example of a hypermap is shown in Figure 9.1.2, where the vertices and edges are represented by points on the surface, and the darts as lines connecting them. Again, all of

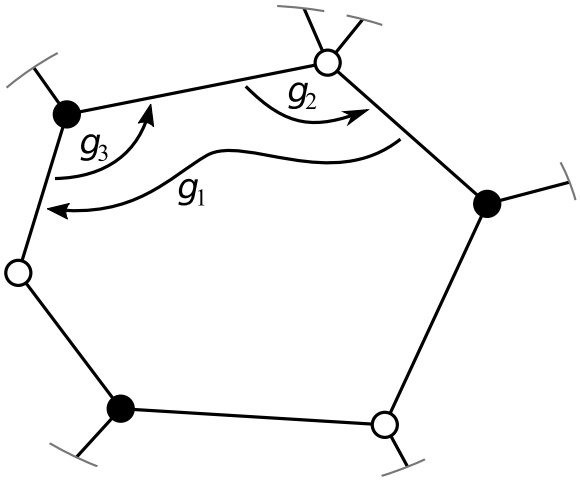


Figure 9.1.3: The actions of the three permutations g_1, g_2 and g_3 on darts in a hypermap, demonstrating how the composition $g_1g_2g_3$ maps darts to themselves.

the faces (connected regions in the complement of the hypermap) are required to be homeomorphic to the open disc.

As with constellations, we can consider isomorphism classes of hypermaps. If we have two hypermaps embedded on two surfaces, then they are said to be isomorphic if there is a homeomorphism between the two surfaces which exactly maps the vertices, edges, faces and darts of one hypermap onto the vertices, edges, faces and darts of the other. Given this fact, if we consider only isomorphism classes of hypermaps, then the exact shape of the surface onto which they're embedded is not relevant. The only properties of the surface which have any impact on the hypermap are the genus and orientability, both of which are preserved under homeomorphisms.

Given this, it is useful to be able to define hypermaps in a geometry-independent manner, and constellations provide a means of doing so. For the remainder of our discussion of hypermaps we will supersede Definition 9.1.7 with the following:

Definition 9.1.8. A hypermap is a 3-constellation [20, p 43].

We can see that these two definitions are equivalent by the following argument. Consider a geometric hypermap, and extract from it the set D containing its darts. Each dart can be associated uniquely to a single vertex, edge and face (the choice of vertex and edge are easy as the dart is connected to only one of each; by convention we will associate a dart to the first face you encounter moving anticlockwise around its adjacent vertex). We define three permutations, g_1, g_2 and g_3 acting on D such that they permute the darts anticlockwise around their associated faces, edges and vertices respectively. These permutations are transitive on D as hypermaps are always connected, so any dart can be moved to any other dart just by repeatedly rotating them

round their adjacent vertices and edges. They also satisfy $g_1 g_2 g_3 = 1$, as can be seen geometrically from their actions (see Figure 9.1.3). By Definition 9.1.4, the sequence $[g_1, g_2, g_3]$ is therefore a 3-constellation.

Conversely, a geometric hypermap can be constructed from a 3-constellation $[g_1, g_2, g_3]$ as follows: first, associate the elements in the set D that the constellation acts on with line segments (darts). Then, taking each dart in turn and, repeatedly applying g_3 followed by g_2, construct a set of open surfaces (faces) such that the darts, in the order they are encountered, form their boundaries in a clockwise direction (when multiple darts result in faces with the same boundary, as will be the case whenever a face has more than one vertex adjacent to it, keep only one such face). Once all possible faces have been constructed, join them together wherever their boundaries share a dart. The result is a closed, orientable surface with a hypermap embedded in it.

Therefore, geometric hypermaps and 3-constellations are equivalent to each other, so we are free to directly associate the two. The concept of isomorphism also maps over naturally, as both hypermap isomorphism and constellation isomorphism amount to mapping the darts of one hypermap onto the darts of another. We also carry the concept of rooting over:

Definition 9.1.9. A *rooted hypermap* is a rooted 3-constellation.

Geometrically this means specifying one of the darts to be distinct from the others; this dart is the hypermap's *root*.

An important point relating to the connection between hypermaps and constellations is the interpretation of cycle structure within the constellations. g_3 rotates darts anticlockwise around their adjacent vertices, so if g_3 is applied repeatedly, all darts will eventually do a complete cycle around their adjacent vertex and return to their starting point. Therefore the cycles in g_3 correspond directly to the vertices in the hypermap, and the set of darts in each cycle is exactly the set of darts connected to the corresponding vertex. Similarly, cycles in g_2 correspond to edges and cycles in g_1 to faces. This has the nice result that, if we wish to know the number of vertices, edges and faces in a hypermap, we simply need to count the cycles in its 3-constellation.

This also allows us to prove another interesting result relating to counting of hypermaps:

Lemma 9.1.1. *Let $N_{d,v,e,f}$ be the number of rooted hypermaps with d darts, v vertices, e edges and f faces. Then*

$$N_{d,v,e,f} = N_{d,e,v,f} = N_{d,v,f,e},$$

along with all other such permutations of v, e and f.

Proof. Consider the set of all rooted hypermaps $[g_1, g_2, g_3]$ with v vertices, e edges, f faces and d darts. The mapping $T_1 : [g_1, g_2, g_3] \to [g_1^{-1}, g_3^{-1}, g_2^{-1}]$

is an isomorphism-preserving bijection between such rooted hypermaps and those that instead have e vertices and v edges. Therefore the two sets are the same size, i.e. $N_{d,v,e,f} = N_{d,e,v,f}$. By the same argument, the bijection $T_2 : [g_1, g_2, g_3] \to [g_2^{-1}, g_1^{-1}, g_3^{-1}]$ gives that $N_{d,v,e,f} = N_{d,v,f,e}$. All other equalities follow from sequential application of these two. □

This result will manifest itself when we start computing generating functions, where we will find that the generating functions are always symmetric in their arguments.

9.2 Generating functions

One final concept which we will be making considerable use of is that of generating functions. These are a very simple concept, however, so we will only give a brief introduction in order to clarify the terminology we will be using in the following chapters.

Consider some sequence of numbers a_k for all $0 \leq k < K$ (K may be infinity). This sequence has an associated *generating function* A defined as

$$A(x) = \sum_{k=0}^{K-1} a_k x^k,$$

i.e. a function (specifically a polynomial if K is finite) with the sequence as the coefficients in its Maclaurin series expansion. In some infinite cases this series will diverge for finite x, in which case we will have to think of $A(x)$ instead as a formal power series, although we will still refer to it as a generating function. In cases like this it is sometimes useful to instead use an *alternating generating function*

$$B(x) = \sum_{k=0}^{\infty} (-1)^k b_k x^k.$$

Generating functions are of great use in enumerative combinatorics. For example, let us say that the number of k-permutations with n disjoint cycles is $N_{k,n}$. For any given k there will be k such values to describe the partition of the $k!$ such permutations. We can simplify this information considerably, however, by combining these all into a single generating function

$$p_k(x) = \sum_{n=1}^{k} N_{k,n} x^n.$$

We say that this generating function *enumerates the set of k-permutations by number of cycles*. As it happens, this function has a simple closed form:

$$p_k(x) = (x)_k = \frac{\Gamma(x+k)}{\Gamma(x)}.$$

where $(x)_k$ is the rising factorial [10, p 213]. Generating functions with closed forms

such as this are very useful as they can reveal much more information about the objects being enumerated than the individual counts can. In this case, for instance, setting $x = 1$ gives us that

$$p_k(1) = \sum_{n=1}^{k} N_{k,n} = k!$$

i.e. there are $k!$ k-permutations. This is, of course, well known, but we will make a fair amount of use of this particular principle (setting a parameter to unity to sum over various possibilities) when dealing with hypermaps and constellations.

Chapter 10

One-face rooted hypermaps

In this chapter we will prove the fact that our functions $P_d(m_1, m_2)$ (defined in Chapter 8) are in fact generating functions for enumerating one-face rooted hypermaps by number of edges and vertices. This is our first new result in enumerative combinatorics, and the methods we develop here provide the groundwork for everything which follows in later chapters for enumeration of general rooted hypermaps, maps and constellations.

10.1 Why matrix integration

When evaluating $P_d(m_1, m_2)$ in Chapter 5, we made considerable use of matrix integration. It is interesting, and indeed instructive, to consider why this particular topic, and the particular types of matrix integrals we will be using, should have such a close connection to combinatorics.

As we have noted, this work is not the first time a connection between matrix integration and combinatorics has been encountered, or even the first time such a connection has arisen from work in quantum theory. For example, work in two-dimensional quantum gravity has produced methods of enumerating maps via computation of matrix integrals [39]. There are many thematic similarities between this procedure and ours – both use Gaussian integrals and integration over spaces of Hermitian matrices, for instance, and both give integrals which can be evaluated to give closed-form generating functions.

As will be seen from the results in this chapter, it is no surprise that two methods involving Gaussian integration over complex matrices would produce combinatorial results. This arises from two facts: a Gaussian-weighted integral over a polynomial can be re-expressed as a multi-derivative expression, and evaluation of these multi-derivative expressions involves summation over permutation groups. We will look at the case of enumerating permutations as a way to illustrate these principles.

10.1.1 Enumerating permutations

As we stated in Section 9.2, the generating function for enumerating k-permutations by number of cycles is already known, so this is not an attempt to derive new

results. Rather it is a demonstration of the principles which link matrix integrals to combinatorics, and in particular the specific tools which we will be making use of in the following chapters.

Consider the integral

$$p_k(m) = \frac{1}{\pi^m} \int_{\mathbb{C}^m} d^m \boldsymbol{v} d^m \bar{\boldsymbol{v}} e^{-\boldsymbol{v}\cdot\boldsymbol{v}} (\boldsymbol{v}\cdot\boldsymbol{v})^k, \tag{10.1.1}$$

where $\boldsymbol{v}$ is an m-dimensional complex vector. This integral can be evaluated by switching to spherical polar coordinates. If we let $r = \sqrt{\boldsymbol{v}\cdot\boldsymbol{v}}$ and integrate out the angular part, we get

$$\begin{aligned} p_k(m) &= \frac{2}{\Gamma(m)} \int_0^\infty e^{-r^2} r^{2k+2m+1} dr \\ &= \frac{\Gamma(m+k)}{\Gamma(m)}, \end{aligned}$$

which is already known to be the generating function for enumerating permutations by cycle count. This integral provides us with a new method for proving this fact, however:

Theorem 10.1.1. *The integral (10.1.1) is the generating function for enumerating k-permutations by number of cycles.*

Proof. Let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ be two real m-vectors. We choose some orthonormal basis for $\mathbb{R}^m$ such that these and $\boldsymbol{v}$ can be written in terms of their components in this basis. Then, in terms of these components, we write (10.1.1) (using the convention of summing over repeated indices) as

$$\begin{aligned} p_k(m) &= \frac{1}{\pi^m} \int_{\mathbb{C}^m} d^m v d^m \bar{v} e^{-|v|^2 + \alpha v_i + \beta_i \bar{v}_i} (v_i \bar{v}_i)^k \Big|_{\alpha,\beta=0} \\ &= \frac{1}{\pi^m} \frac{\partial}{\partial\alpha_{i_1}} \frac{\partial}{\partial\beta_{i_1}} \cdots \frac{\partial}{\partial\alpha_{i_k}} \frac{\partial}{\partial\beta_{i_k}} \int_{\mathbb{C}^m} d^m v d^m \bar{v} e^{-|v|^2 + \alpha_i v_i + \beta_i \bar{v}_i} \Big|_{\alpha,\beta=0} \\ &= \frac{\partial}{\partial\alpha_{i_1}} \frac{\partial}{\partial\beta_{i_1}} \cdots \frac{\partial}{\partial\alpha_{i_k}} \frac{\partial}{\partial\beta_{i_k}} e^{\alpha_i \beta_i} \Big|_{\alpha,\beta=0} \\ &= \frac{\partial}{\partial\alpha_{i_1}} \cdots \frac{\partial}{\partial\alpha_{i_k}} (\alpha_{i_1} \cdots \alpha_{i_k}) \Big|_{\alpha=0}. \end{aligned} \tag{10.1.2}$$

Here we have used our ability to replace v_i with $\partial/\partial\alpha_i$ and $\bar{v}_i$ with $\partial/\partial\beta_i$ to remove the polynomial part $(v_i\bar{v}_i)^k$ from the Gaussian integral in order to simplify its evaluation. This leaves us with a multi-derivative expression involving α and β, and after performing all of the β-derivatives we are left with a k^{th} derivative of a k^{th} order monomial of α.

The process of performing the α-derivatives is trickier, but can be done combinatorially. When we expand the derivative out fully we get a total of $k!$ terms, one for each pairing of derivatives to α terms (each α is paired with only one derivative

as differentiating it twice would take it to zero). This is equivalent to a sum over all k-permutations, each permutation corresponding to one of these pairings. Given that

$$\frac{\partial}{\partial \alpha_i}\alpha_j = \delta[i,j],$$

where $\delta[i,j]$ is the m-dimensional Kronecker delta, we can write this sum of pairings as

$$p_k(m) = \sum_{g \in Sym_k} \delta[i_1, i_{g(1)}] \cdots \delta[i_k, i_{g(k)}].$$

For each g, the product of deltas contracts completely, and each cycle in g contracts to a single factor of m (i.e. a cycle (152) would give a term $\delta[i_1,i_5]\delta[i_5,i_2]\delta[i_2,i_1] = \delta[i_1,i_1] = m$). Therefore

$$p_k(m) = \sum_{g \in Sym_k} m^{\mathrm{cyc}(g)},$$

which is exactly the generating function for enumerating k-permutations by number of cycles. □

We can now see much more clearly why Gaussian integrals such as this can produce generating functions. Of particular importance is the fact that the monomial function in the integrand is a complete contraction (all indices appear exactly twice and are summed over), and v terms are always paired with $\bar{v}$ terms in this contraction. This means that, when we perform the integration-differentiation process which leads to (10.1.2), what results is a sum of completely contracted products of Kronecker deltas, which themselves produce monomial functions of the dimensions of the matrix.

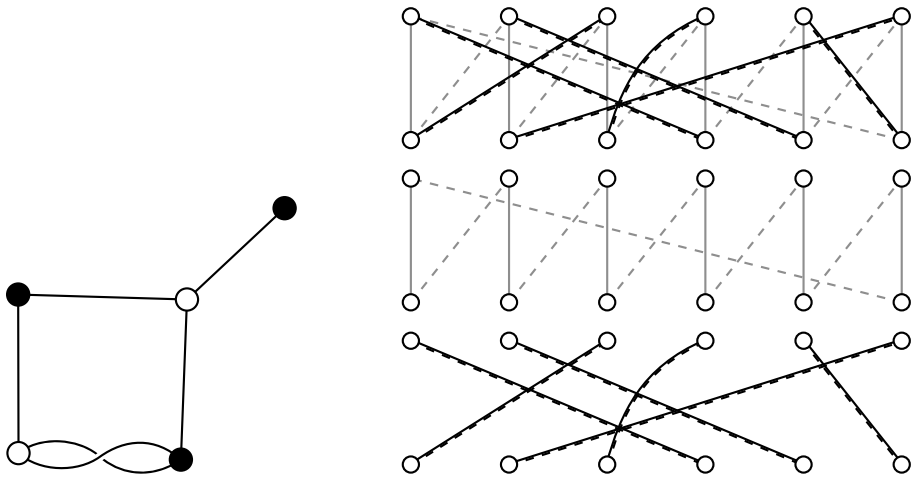
The integrals used in this example bear a similarity to those discussed in Section (5.1.1), so naturally the proof in the next section follows a very similar procedure to the one above. We are thus ready to proceed to the main purpose of this chapter: enumeration of one-face hypermaps.

10.2 The generating function

In the previous section we showed how to convert a matrix integral into a sum over permutations. We will now use this to show how to enumerate one-face rooted hypermaps. It is easy to see that this is equivalent to summing over permutations, for the following reason:

Lemma 10.2.1. *There is a bijection between the set of rooted hypermaps with d darts and one face, and the set Sym_d of d-permutations.*

Proof. A hypermap with d darts is a 3-constellation of degree d, where the number of cycles in the first permutation corresponds to the number of faces (Definition 9.1.8). Any hypermap with one face can therefore be written, up to isomorphism, as $[\sigma, g, (\sigma g)^{-1}]$, where $\sigma = (12\ldots d)$ and g is some other permutation. Furthermore,



(a) (b)

Figure 10.2.1: (a) The hypergraph corresponding to the genus one one-face hypermap $\{(123456), (143)(256), (125)(3)(46)\}$, and (b) the corresponding ladder diagram. For clarity the two subcomponents – the backbone in grey, and the overlaid permutation in black – have been separated out.

when we consider only rooted hypermaps, there are no non-trivial isomorphisms available which preserve this particular normal form (the only isomorphisms which would preserve the given σ correspond to the action of permutations $p = \sigma^k$ for some integer k, and the only case like this which preserves the root is when $p = 1$). We therefore see that no two hypermaps written in this form with different g can be in the same rooted hypermap isomorphism class. Thus, this representation provides a bijection between d-permutations and one-face rooted hypermaps with d darts. □

Keeping in mind our goal of showing that $P_d(m_1, m_2)$ is a generating function for counting one-face rooted hypermaps, this lemma implies the result of Corollary 8.0.2 – that $P_d(1, 1) = d!$ – as $P_d(1, 1)$ should equal the total number of d-permutations, which we know to be $d!$.

This correspondence given above can be represented diagrammatically, using what we will refer to here as *ladder diagrams*. Consider, for example, the permutation $g = (143)(256)$. Through Lemma 10.2.1 this corresponds to a genus one rooted hypermap, whose hypergraph is shown in Figure 10.2.1a. We build its ladder diagram as follows:

1. Draw two rows of d points (two rows of six in this example), and join them pairwise between the rows by solid vertical lines.

2. Draw six dashed lines which represent $\sigma = (12 \dots d)$ mapping from the bottom row to the top row. These along with the d vertical lines are called the *backbone* of the diagram.

3. Draw six double lines (half solid, half dashed) which represent g mapping from the top to the bottom.

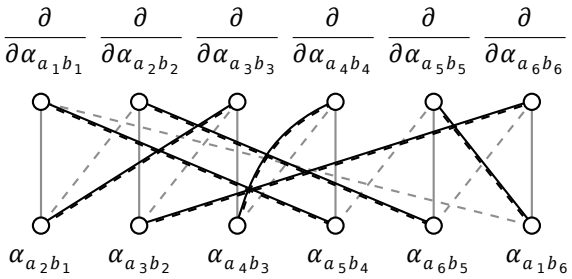


Figure 10.2.2: One of the Kronecker delta contractions involved in the expansion of $P_6(m_1, m_2)$. This diagram corresponds to the term $\delta[a_1, a_{\sigma g(1)}]\delta[b_1, b_{g(1)}]\cdots\delta[a_6, a_{\sigma g(6)}]\delta[b_6, b_{g(6)}]$ with $g = (143)(256)$, and the diagram of contraction pairs visibly matches Figure 10.2.1b.

The ladder diagram for $g = (143)(256)$ is shown in Figure 10.2.1b.

This ladder method gives a quick method of extracting the hypermap's properties from g. If we follow the solid lines through the ladder diagram, moving up on the backbone then down on the double lines, and repeat this over and over, this corresponds simply to repeated applications of g to the top row. As such, each closed loop of solid lines is a cycle in g, so the number of solid loops equals the number of edges in the hypermap. Doing the same with the dashed lines corresponds to repeated applications of σg, and so by the same argument the number of dashed loops equals the number of vertices. This visualisation will make the following proof somewhat simpler to follow.

Theorem 10.2.1. *$P_d(m_1, m_2)$ is a generating function enumerating one-face rooted hypermaps with d darts by number of vertices and edges.*

Proof. We have from (8.0.1) and (5.1.5) that

$$\begin{aligned} P_d(m_1, m_2) &= \frac{\Gamma(m_1m_2 + d)}{\Gamma(m_1m_2)}\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d]\rangle \\ &= \frac{1}{\pi^{m_1m_2}}\int_{\mathbb{C}^{m_1m_2}} d^{m_1m_2}z d^{m_1m_2}\bar{z} e^{-|z|^2} z_{a_1b_1}\bar{z}_{a_2b_1}\cdots z_{a_db_d}\bar{z}_{a_1b_d}, \end{aligned}$$

where z_{ab} are components of an $m_1 \times m_2$-dimensional complex matrix. The method used in Theorem (10.1.1) for evaluating Gaussian integrals applies equally well here; we just need to define our dummy variables α and β as $m_1 \times m_2$-dimensional real matrices. We then get that

$$\begin{aligned} P_d(m_1, m_2) &= \frac{\partial}{\partial\alpha_{a_1b_1}}\frac{\partial}{\partial\beta_{a_2b_1}}\cdots\frac{\partial}{\partial\alpha_{a_db_d}}\frac{\partial}{\partial\beta_{a_1b_d}} e^{\alpha_{ab}\beta_{ab}}\Big|_{\alpha,\beta=0} \\ &= \frac{\partial}{\partial\alpha_{a_1b_1}}\cdots\frac{\partial}{\partial\alpha_{a_db_d}}(\alpha_{a_2b_1}\alpha_{a_3b_2}\cdots\alpha_{a_1b_d})\Big|_{\alpha=0}. \end{aligned}$$

As before, this expression is equivalent to a sum over permutations. We can write the terms in this sum entirely in terms of m_1- and m_2-dimensional Kronecker deltas, but for clarity we will represent them diagrammatically, as seen for example in Figure 10.2.2. This diagram represents all the individual contractions arising in the term corresponding to a particular permutation g matching the derivatives to the α terms.

Single solid lines represent contractions over matching b-indices, while single dashed lines represent contractions over matching a-indices. The double lines correspond to the derivative pairings, each of which produces a $\delta[a_i, a_{\sigma g(i)}]$ and a $\delta[b_i, b_{g(i)}]$. Therefore, in each term the m_1-dimensional Kronecker deltas with a-indices contract along the dashed lines, producing a factor of m_1 for each closed dashed loop, and the m_2-dimensional deltas with b-indices contract over the solid loops, producing a factor of m_2 for each loop.

These contraction diagrams clearly correspond to the ladder diagrams we discussed previously. We have already shown that there is one such ladder diagram for each one-face rooted hypermap with d darts, and we have here that there is one such diagram for each contraction term in $P_d(m_1, m_2)$. Therefore, there is a one-to-one correspondence between the contraction terms and one-face rooted hypermaps. Furthermore, by comparing the interpretations of the loops in the two types of diagrams we see that the exponent of m_1 in each term is the number of vertices in the corresponding hypermap and the exponent of m_2 the number of edges. Thus, $P_d(m_1, m_2)$ is exactly the generating function enumerating one-face rooted hypermaps with d darts by number of vertices and edges. □

10.3 Evaluating $P_d(m_1, m_2)$

We have now shown explicitly that $P_d(m_1, m_2)$ is the generating function enumerating one-face rooted hypermaps with d darts by number of vertices and edges. Thus, by computing these functions exactly and finding their coefficients, we are able count how many of the corresponding types of rooted hypermap there are. The question is what benefits this new method brings. As we have seen, there already exists a method by Walsh for enumerating rooted hypermaps by generating all of the hypermaps individually [35], so we do have something to compare these results to.

One way we could compare these two methods is by looking at the amount of time it takes to compute any given value or set of counts, and how this increases with d, but the difference between the two methods is somewhat more fundamental than just speed; while our method is built on generating functions, which essentially give us a short-cut to the result, Walsh's method is based on individually generating each hypermap and computing their properties. Which method is preferable therefore really depends on what you need – the hypermaps themselves or just the counts.

10.3.1 Walsh's method

Let us consider Walsh's method, and the principles it is based on, first. This method generates each hypermap individually and then computes its properties. Direct generation of combinatorial objects is by no means simple, of course. Taking the simpler example of permutations, it is very easy to generate an individual permutation (simply take a finite, ordered set and randomly shuffle the elements). One can generate any permutation this way of course, but this is not the same as enumerating them. We

would have to store all of the previously-generated permutations in order to check if each new one had been counted already, and the fact that this method is random in nature means we'd never know for certain if we had found all cases. In order to overcome these two issues, what is needed is a systematic method for generating each case which is not reliant on checking for uniqueness, and which halts when all cases have been found. Algorithms of this form exist for permutations, with one of the better known – and most efficient – being Heap's algorithm [18, 29].

Similar principles apply with hypermaps. We could easily generate all 3-sequences of permutations using Heap's algorithm (taking the last one to be the inverse of the product of the others to ensure they multiply to unity), but we would then have to check each case for transitivity, wasting a lot of time generating cases which aren't actually hypermaps. Then we would also have to store all previously-generated cases in order to check that new ones weren't isomorphic to them, requiring significant time and memory.

Walsh overcame both of these problems by using an alternate representation of hypermaps, whereby each hypermap is represented by a code string. This originated from a similar code for rooted plane trees, which was then generalised by Lehman to a code for rooted maps [34]. By enforcing certain conditions on these codes they could be made to correspond only to bipartite, bicoloured maps, thus allowing their use for generating hypermaps as well. The enumeration process then requires just generating all codes satisfying these conditions, as the nature of the codes ensures that they are nonisomorphic [35], much as our ladder-diagram representation for one-face rooted hypermaps ensured that all the cases generated were nonisomorphic.

The exact nature of the codes is not important for our work here, but one property they have is that, as is the case with the constellation representation of hypermaps, it is possible to extract the number of vertices, edges and faces from each code with ease. Thus the hypermaps can be partitioned by these characteristics after generation, allowing individual counts of each type to be made. Walsh provides such counts for rooted hypermaps up to 12 darts, noting that the time taken is roughly proportional to the number of hypermaps counted, and the memory requirements are just the memory required to store a single hypermap code, along with that needed for the counts themselves [35].

This is as efficient as a direct enumeration algorithm can be, making it ideal for applications which require the actual structure of the individual hypermaps, but in terms of absolute speed it is nonetheless quite slow, simply because of how many rooted hypermaps there are. The number of rooted hypermaps with d darts grows as at least $d!$, after all.

10.3.2 Direct enumeration by our method

If we are only interested in one-face rooted hypermaps, the first improvement on Walsh's method we can make is a new direct enumeration process which is slightly more optimised for the one-face case, based on Lemma 10.2.1. As they stand, Walsh's

codes can encode any number of faces, so extracting only the one-face cases with an unmodified version of his algorithm would involve generating *all* rooted hypermaps and filtering the results. But as we have shown, each one-face rooted hypermap is equivalent to a permutation on d elements, and these can be enumerated extremely efficiently using Heap's algorithm. From these permutations we construct constellations, which gives us the structure of the hypermaps including vertex/edge counts.

As with Walsh's method, the memory requirements of this process grows linearly with the size of the hypermap and the time taken grows with the number of hypermaps. This method is thus only an improvement if we are only looking at one-face cases, where it saves looking at unnecessary cases. This method is not applicable beyond those cases unfortunately.

However, to see the real benefits of our new method, we must move beyond direct enumeration.

10.3.3 Closed-form generating functions

In Theorem 10.2.1 we showed that the generating function $P_d(m_1, m_2)$ can be expressed as a Gaussian matrix integral. Evaluation of this integral gives us the direct enumeration method described above, but its real value lies in its connection to the quantum work discussed in Part II, as this work allows us to write $P_d(m_1, m_2)$ in closed form. As we stated in (8.0.2),

$$P_d(m_1, m_2) = \frac{1}{d!} \sum_{r \geq 0} \binom{d-1}{r} (-1)^r (m_1 - r)_d (m_2 - r)_d.$$

While we have chosen to write this as an indefinite summation, we note as before that the binomial coefficient ensures that the summand is zero for all $r \geq d$. The two rising-factorial terms $(m_1 - r)_d$ and $(m_2 - r)_d$ produce finite-order polynomials in m_1 and m_2, and as such this expression can be used to compute $P_d(m_1, m_2)$ exactly as a finite-order polynomial, from which the coefficients (the hypermap counts) can be extracted.

To compare this result quantitatively to the direct enumeration methods above, including Walsh's, we computed these functions individually in Mathematica, timing how long it took to calculate each one. It is difficult to find a theoretical expression for the computational complexity for this process, as the numbers used during the calculation rise rapidly (at least as $d!$) so any algorithm intending to compute these functions at high order must necessarily make use of variable-size integer storage, adding an additional layer of complexity. However, based on the results of a run going up to order 100 the time-dependence appears to be approximately polynomially bounded, with the shown cases fitting roughly to $\mathcal{O}(d^{3.3})$ (see Figure 10.3.1). This is in comparison to the algorithms taking constant time per hypermap, which would run in $\mathcal{O}(d!)$.

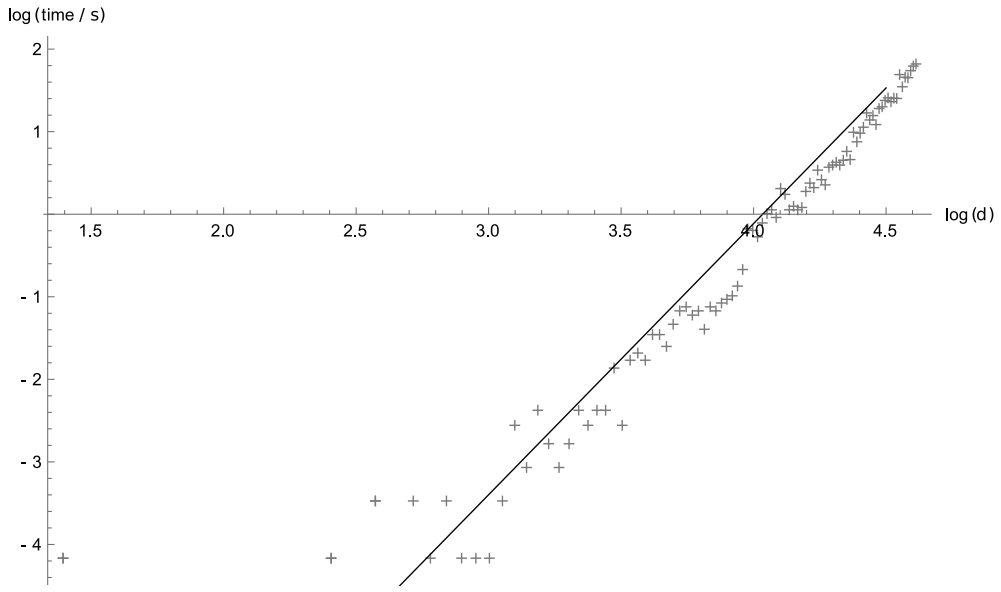


Figure 10.3.1: A log-log plot showing the time taken to compute $P_d(m_1, m_2)$ up to $d = 100$ in Mathematica on a Dell XPS 12 with an Intel Core i7-3537U CPU. The fitted line is $T = (1.755 \times 10^{-6})d^{3.285}$.

10.3.4 Recursion relation

There is one more method we can use to calculate these generating functions, which can in fact provide an even more significant speed boost than the generating-function method above: recursion relations. This method is really just an extension of the generating-function method, however, as the recursion relation's proof follows directly from the form of the hypergeometric series (8.0.2), and then gives a new method of computing these functions in terms of functions of lower order.

We have already proven the relevant recursion relation in Theorem 8.0.2 – as (8.0.3) states,

$$\begin{aligned}(d+3)P_{d+2}(m_1, m_2) = (2d+3)(m_1+m_2)P_{d+1}(m_1, m_2) \\ + d[(d+1)^2 - (m_1 - m_2)^2]P_d(m_1, m_2)\end{aligned}$$

for any $d \geq 1$ – and we have already seen how this recursion can be used to prove inductively a number of facts, such as the fact that $P_d(m_1, m_2)$ is symmetric, and that $P_d(1, 1) = d!$. Additionally, we can rewrite (8.0.3) as a recursion relation for the hypermap counts themselves:

Corollary 10.3.1. *Let $N_{d,v,e}$ be the number of rooted hypermaps with d darts, v vertices and e edges. Then*

$$\begin{aligned}(d+3)N_{d+2,v,e} = (2d+3)(N_{d+1,v-1,e} + N_{d+1,v,e-1}) \\ + d[(d+1)^2 N_{d,v,e} - N_{d,v-2,e} + 2N_{d,v-1,e-1} - N_{d,v,e-2}]. \quad (10.3.1)\end{aligned}$$

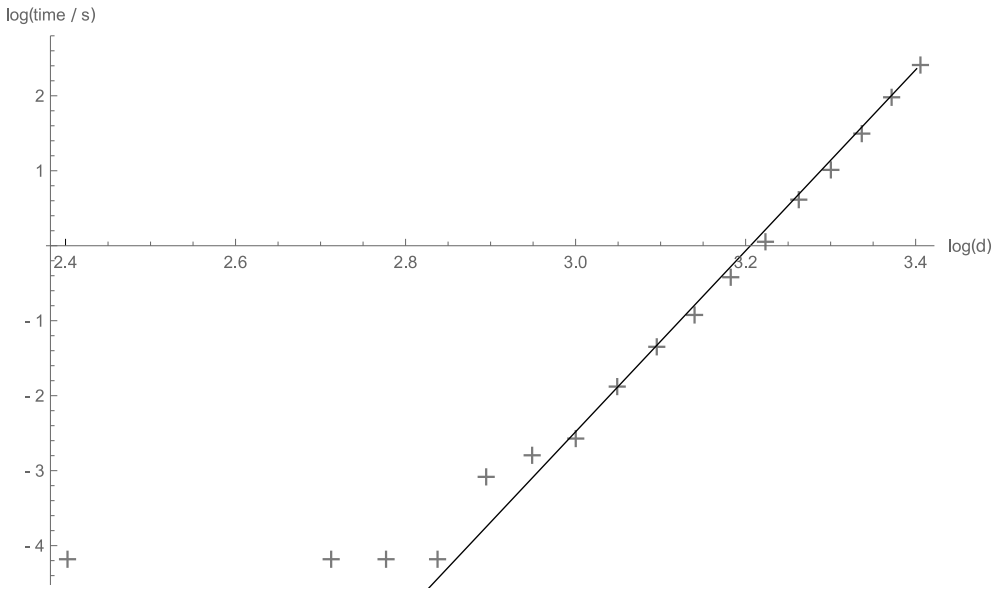


Figure 10.3.2: The time taken to compute $P_d(m_1, m_2)$ via its recursion relation up to $d = 30$, assuming no prior knowledge of the lower-degree generating functions in each case, in Mathematica on a Dell XPS 12 with an Intel Core i7-3537U CPU. The fitted line is to $T = (1.574 \times 10^{-17})d^{12.07}$. Lower values of d were omitted from the fit as their times are visibly rounded up due to the CPU clock's finite resolution.

Proof. Write

$$P_d(m_1, m_2) = \sum_{v=1}^{\infty} \sum_{e=1}^{\infty} N_{d,v,e} m_1^v m_2^e$$

and substitute this into (8.0.3). Collecting together the $m_1^v m_2^e$ terms then gives (10.3.1). □

Of course, we can also come up with an explicit formula for $N_{d,v,e}$ by expanding (8.0.2), using the fact that the rising factorial has the expansion

$$(a)_k = \sum_{n=0}^{k} (-1)^{k-n} s(k, n) a^n,$$

with $s(k, n)$ being the Stirling numbers of the first kind. However, the Stirling numbers are themselves computed recursively, so this method has little added benefit over using (10.3.1).

It's difficult to compare this method to the generating-function method in Section 10.3.3 in terms of time complexity, in that the two methods are designed to be used in different contexts. The generating-function method is able to compute the functions in isolation, without any knowledge of functions of lower order, so it is not surprising that, in this context, it is more efficient. Figure 10.3.2 shows the time taken to compute each $P_d(m_1, m_2)$ recursively, assuming no prior knowledge of lower-degree generating functions at each step, and it can be seen that the times rise much more rapidly than in Figure 10.3.1.

The recursive method, however, is by far the most efficient method to use if the

prior cases *are* already known, where the time to compute each case is only affected by the amount of time it takes to perform the various integer/polynomial arithmetic operations required to evaluate (8.0.3), the number of such steps being performed not depending on d. The recursive method is also more suited to the task of proving patterns and trends in the generating functions, particularly through inductive proofs.

10.3.5 Discussion

So, while our enumeration methods based on calculating $P_d(m_1, m_2)$ cannot provide as much information about the individual hypermaps as Walsh's method can, it is clear that our methods have definite advantages. If we just want to know the overall numbers of different sizes of rooted hypermaps, it is much faster to get these from the generating functions than it is to count the hypermaps themselves. In addition, generating functions and recursion relations are ideally suited to proving patterns and trends in the counts.

It should be noted that some prior work, which we were initially unaware of, has dealt with and evaluated functions equivalent to $P_d(m_1, m_2)$ – a survey of several methods for doing so is given in [28]. These methods arose from a combinatorial problem equivalent to the one addressed in this chapter, namely the problem of enumerating factorisations of the permutation $(12 \dots d)$ into a product of two permutations. Each such factorisation $g_1 g_2 = (12 \dots d)$ has a one-to-one correspondence to the one-face rooted hypermap $[(12 \dots d), g_2^{-1}, g_1^{-1}]$, so these two enumeration problems are equivalent, and thus result in the same generating function.

The various generating function expressions discussed in [28] have a number of similarities with our expression (8.0.2) for $P_d(m_1, m_2)$, particularly in the use of hypergeometric series, though the exact forms differ in each case. However, what is important for the purposes of the rest of this thesis is not the form or value of the functions themselves but the method we used to derive (8.0.2). This method is applicable beyond the one-face rooted hypermap case studied in this chapter, and we will begin to explore this in the next chapter by extending our method to cover all rooted hypermaps, showing how to overcome the one-face restriction. After passing this hurdle we will then find that the methods can be used on still broader problems, such as enumerating rooted maps and constellations (see Chapter 12).

Chapter 11

General rooted hypermaps

In the previous chapter we looked exclusively at the enumeration of rooted hypermaps with one face, and showed that we were able to find closed-form expressions for the generating functions enumerating them by number of edges and vertices. As is often the case, this special case has a much simpler solution than the general case will have, but that doesn't mean that the general case doesn't have a solution. In this chapter we will look at this general case: enumeration of all rooted hypermaps[1]. The solution we will find here is nowhere near as neat as in the one-face case, but it is still in a form which allows us to compute generating functions (and therefore rooted hypermap counts) algorithmically.

One potential way to proceed to other numbers of faces would be to adjust the method shown in the last chapter to count rooted hypermaps with other numbers of faces. The two-face case is reasonably easy to do (see Section 11.1), but the method gets progressively more complex as the number of faces increases.

However, the fact that the functions $P_d(m_1, m_2)$ are symmetric (as a result of Lemma 9.1.1) hints at a simpler method. Let $P_d^{(f)}(m_1, m_2)$ be the generating function for f-face rooted hypermaps with d darts, with $P_d^{(1)}(m_1, m_2) \equiv P_d(m_1, m_2)$. If we sum over all these we get a new generating function

$$H_d(m_0, m_1, m_2) = \sum_{f=1}^{d} m_0^f P_d^{(f)}(m_1, m_2),$$

which enumerates all rooted hypermaps with d darts at once. Lemma 9.1.1 again means that this function will be symmetric in all three of its parameters. This suggests that it may itself have a reasonably simple form – at least simpler than the individual $P_d^{(f)}(m_1, m_2)$ terms are.

The final result of this chapter will be a method of computing these global generating functions. Unlike in the one-face case, however, the method for computing these necessarily involves the use of recursion relations, each generating function being dependent on all those for lower dart counts.

[1] We will look at an even more general case, enumeration of all rooted constellations, in the next chapter.

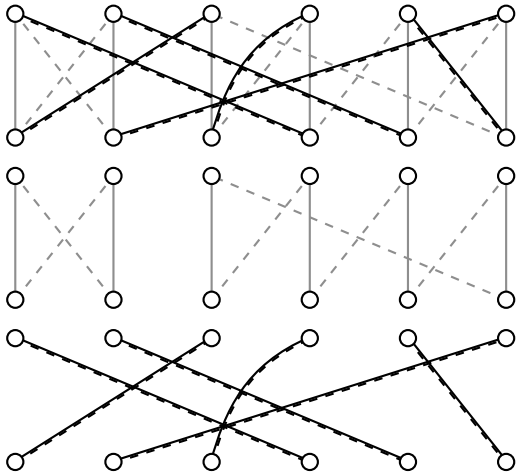


Figure 11.1.1: The ladder diagram for the hypermap $\{(12)(3456), (143)(256), (16425)(3)\}$. This is the same basic diagram as in Figure 10.2.1b, with the only difference being in the backbone (grey), which now has two cycles of length 2 and 4.

11.1 The two-face case

We will nonetheless look at the two-face case on its own first, as it demonstrates the general process through which the methods in the previous chapter are generalised to multiple faces. As in the previous chapter we will represent hypermaps using ladder diagrams (see Figure 11.1.1). The exact same construction principle is used here: we start off by representing our rooted hypermap as $[\sigma, g, (\sigma g)^{-1}]$, where σ is a two-cycle with all indices in its cycle notation in the correct order (e.g. (12)(3456), (123)(456)).

In the previous chapter we showed how one can then sum over all possible permutations g and produce a Gaussian integral expression for a generating function which enumerates the associated ladder diagrams. The exact same principle applies here – as we labelled the expression associated with a one-cycle of length d as $P_d(m_1 m_2)$, we will label the expression for a 2-cycle σ with cycles of length i and j as $P_{i,j}(m_1, m_2)$.

For a given permutation g, the associated term in $P_{i,j}(m_1, m_2)$ is

$$\delta[a_1, a_{\sigma g(1)}]\delta[b_1, b_{g(1)}] \cdots \delta[a_d, a_{\sigma g(d)}]\delta[b_d, b_{g(d)}],$$

exactly as it was in the one-face case (with the only change being the definition of σ), and this produces a monomial in m_1 and m_2 with the exponents being the number of solid and dashed loops in the associated ladder diagram, as we want. The difference here is how the above converts into a derivative expression when we sum over all g. As an example, when $\sigma = (12)(3456)$ i.e. when $i = 2$ and $j = 4$, we get

$$P_{2,4}(m_1, m_2) = \frac{\partial}{\partial \alpha_{a_1 b_1}} \cdots \frac{\partial}{\partial \alpha_{a_6 b_6}} (\alpha_{a_2 b_1} \alpha_{a_1 b_2} \alpha_{a_4 b_3} \alpha_{a_5 b_4} \alpha_{a_6 b_5} \alpha_{a_3 b_6}) \Big|_{\alpha=0}$$

(note that the a-indices on the α terms have been permuted by σ), and the Gaussian

integral corresponding to this is

$$P_{2,4}(m_1,m_2) = \frac{1}{\pi^{m_1m_2}} \int_{\mathbb{C}^{m_1m_2}} d^{m_1m_2}z d^{m_1m_2}\bar{z} e^{-|z|^2} z_{a_1b_1}\bar{z}_{a_2b_1} z_{a_2b_2}\bar{z}_{a_1b_2} \\ \times z_{a_3b_3}\bar{z}_{a_4b_3} z_{a_4b_4}\bar{z}_{a_5b_4} z_{a_5b_5}\bar{z}_{a_6b_5} z_{a_6b_6}\bar{z}_{a_3b_6}.$$

As in the one-face case, we wish to convert this integral into an eigenvalue integral in order to make it useful. First, we convert the Gaussian integral to a spherical integral, through a reverse of the process used in Section 5.1.1. This gives us

$$P_{2,4}(m_1,m_2) = \frac{\Gamma(m_1m_2+6)}{2\pi^{m_1m_2}} \int_{S^{2m_1m_2-1}} d\Omega x_{a_1b_1}\bar{x}_{a_2b_1} x_{a_2b_2}\bar{x}_{a_1b_2} \\ \times x_{a_3b_3}\bar{x}_{a_4b_3} x_{a_4b_4}\bar{x}_{a_5b_4} x_{a_5b_5}\bar{x}_{a_6b_5} x_{a_6b_6}\bar{x}_{a_3b_6},$$

and using $(\hat{\rho}_1^{12})_{a_1a_2} = x_{a_1b}\bar{x}_{a_2b}$ we have

$$P_{2,4}(m_1,m_2) = \frac{\Gamma(m_1m_2+6)}{2\pi^{m_1m_2}} \int_{S^{2m_1m_2-1}} d\Omega \mathrm{Tr}[(\hat{\rho}_1^{12})^2]\mathrm{Tr}[(\hat{\rho}_1^{12})^4].$$

Following the same procedure through in the general case, we find that

$$P_{i,j}(m_1,m_2) = \frac{\Gamma(m_1m_2+i+j)}{2\pi^{m_1m_2}} \int_{S^{2m_1m_2-1}} d\Omega \mathrm{Tr}[(\hat{\rho}_1^{12})^i]\mathrm{Tr}[(\hat{\rho}_1^{12})^j]. \tag{11.1.1}$$

Each of the $\mathrm{Tr}[(\hat{\rho}_1^{12})^i]$ terms in the integrand corresponds directly to a cycle of length i in σ (just as the $\mathrm{Tr}[(\hat{\rho}_1^{12})^d]$ term in the one-face case corresponds to a one-cycle of length d), while the prefactor depends only on the integrand's order in $\hat{\rho}_1^{12}$, which is $i+j$.

We could continue all the way through, evaluating this integral in closed form using the method of Lloyd, Pagels, Page and Sen [21, 24, 30]. However, the resulting expression is less useful to us than the one-face case was, and all the results it could give us will be obtainable from the general case as well. So we will not bother with this here, and instead move on to how we use $P_{i,j}(m_1,m_2)$ to build generating functions.

11.1.1 Constructing the generating functions

The functions $P_{i,j}(m_1,m_2)$ which we defined in the previous section are not themselves hypermap generating functions, for a number of reasons.

- They include non-transitive cases. The summation over permutations which defines $P_{i,j}(m_1,m_2)$ places no restrictions on the permutation g, so cases where $[\sigma, g, (\sigma g)^{-1}]$ is non-transitive (and is therefore not a hypermap) are included in the sum.

- Some of the counted hypermaps are isomorphic to each other. As in the one-face case, the fact that we are counting rooted hypermaps ensures that we don't need to consider isomorphisms which cycle the indices of the first cycle in σ – although such isomorphisms still give hypermaps with the same σ, they do not preserve the

root. However, in the two-face case, isomorphisms which only cycle the indices in the second cycle of σ do preserve the root, so must be considered. These result in each valid rooted hypermap being counted j times (see Appendix E for the proof of this).

Neither of these points were issues in the one-face case, but now we are moving to more general cases we need to account for them. This is fortunately not too difficult. The non-transitive cases can be accounted for by using the fact that the non-transitive cases can be decomposed a direct sums of transitive cases, and the multiple counting is easy to account for as the size of each hypermap's isomorphism class only depends on the cycle structure of σ, and not on g.

Theorem 11.1.1. *The generating function $P_d^{(2)}(m_1, m_2)$ for enumerating two-face rooted hypermaps with d darts by number of vertices and edges is*

$$P_d^{(2)}(m_1, m_2) = \sum_{j=1}^{d-1} \frac{1}{j} \left[P_{d-j,j}(m_1, m_2) - P_{d-j}(m_1, m_2) P_j(m_1, m_2) \right].$$

Proof. We start with the function $P_{d-j,j}(m_1, m_2)$, where we use $i = d - j$ to fix the overall dart count at d. j must be between 1 and $d - 1$ inclusive, as both of the cycles in σ must be of non-zero length.

We first need to subtract the non-transitive cases. Each such case is a direct sum of two transitive one-face cases, one of length $d - j$ and the other of length j, so the generating function for these is simply the product of two one-face generating functions. Subtracting this away gives

$$P_{d-j,j}(m_1, m_2) - P_{d-j}(m_1, m_2) P_j(m_1, m_2),$$

which is a sum only over valid hypermaps. This still contains multiple counting, but as each hypermap is in an isomorphism class of size j, we can account for this by simply dividing by j. We then sum over all possible values of j, with the resulting generating function being

$$P_d^{(2)}(m_1, m_2) = \sum_{j=1}^{d-1} \frac{1}{j} \left[P_{d-j,j}(m_1, m_2) - P_{d-j}(m_1, m_2) P_j(m_1, m_2) \right].$$

□

This example demonstrates the basic principles of generalising to all rooted hypermaps, demonstrating the types of degeneracies which need to be accounted for. We could, if we wished, apply the same methods to higher numbers of faces, but the resulting expressions, even without explicitly evaluating the integrals, will become impractical very quickly, so we will at this point move on to looking at the general case.

11.2 Enumerating all rooted hypermaps

In the previous section we built the generating function for two faces by first defining the function $P_{i,j}(m_1, m_2)$. This function is itself a generating function, but instead of hypermaps it counts ladder diagrams, which are related to hypermaps but do not require the transitivity property. For our general case we now need to generalise these functions further.

$P_{i,j}(m_1, m_2)$ is defined entirely by the backbone of the ladder diagrams, the permutation σ, with the indices i and j being the lengths of the cycles in σ. When σ has more than two cycles i.e. N cycles with lengths $d_1, \ldots, d_N$, then the obvious generalisation of $P_{i,j}(m_1, m_2)$ as given in (11.1.1) is

$$P_{d_1,\ldots,d_N}(m_1, m_2) = \frac{\Gamma(m_1 m_2 + d)}{2\pi^{m_1 m_2}} \int_{S^{2m_1 m_2 - 1}} d\Omega \prod_{r=1}^{N} \mathrm{Tr}[(\hat{\rho}_1^{12})^{d_r}], \tag{11.2.1}$$

where $d = \Sigma_{r=1}^{N} d_r$. It is fortunately reasonably easy to prove that this is correct.

Lemma 11.2.1. *Let σ be the permutation on $[1 \ldots d]$ with cycle lengths $d_1, \ldots, d_N$, such that the indices in its cycle representation are in the correct order, e.g. if $\{d_1, d_2, d_3\} = \{3, 2, 4\}$ then $\sigma = (123)(45)(6789)$. Then $P_{d_1,\ldots,d_N}(m_1, m_2)$ as given in (11.2.1) is a generating function enumerating all ladder diagrams with σ as their backbone, partitioning them by number of solid and dashed loops.*

Proof. This proof works in much the same way as that for Theorem 10.2.1. First we need to convert (11.2.1) into a Gaussian integral using the method from Section 5.1.1. As was the case there, the integrand in our spherical integral is a monomial of order d in the components of $\hat{\rho}_1^{12}$, or equivalently a monomial of order $2d$ in the unit vector components x_{ab} and their complex conjugates, so all of the same basic steps work here.

First we rewrite (11.2.1) using the fact that $[\hat{\rho}_1^{12}]_{a_1 a_2} = x_{a_1 b} \bar{x}_{a_2 b}$. It is most convenient to express the integral as

$$P_{d_1,\ldots,d_N}(m_1, m_2) = \frac{\Gamma(m_1 m_2 + d)}{2\pi^{m_1 m_2}} \int_{S^{2m_1 m_2 - 1}} d\Omega \prod_{i=1}^{d} x_{a_i b_i} \bar{x}_{a_{\sigma(i)} b_i}.$$

With the way σ is defined, this product reproduces the product of traces seen in (11.2.1). Now we multiply this by the Gaussian factor

$$\frac{2}{\Gamma(m_1 m_2 + d)} \int_0^\infty \lambda^{2m_1 m_2 - 1 + 2d} e^{-\lambda^2} d\lambda = 1$$

and substitute in $z_{ab} = \lambda x_{ab}$ to get

$$\begin{aligned} P_{d_1,\ldots,d_N}(m_1,m_2) &= \frac{1}{\pi^{m_1 m_2}} \int_{\mathbb{C}^{m_1 m_2}} d^{m_1 m_2} z d^{m_1 m_2} \bar{z} e^{-|z|^2} \prod_{i=1}^{d} z_{a_i b_i} \bar{z}_{a_{\sigma(i)} b_i} \\ &= \frac{\partial}{\partial \alpha_{a_1 b_1}} \frac{\partial}{\partial \beta_{a_{\sigma(1)} b_1}} \cdots \frac{\partial}{\partial \alpha_{a_d b_d}} \frac{\partial}{\partial \beta_{a_{\sigma(d)} b_d}} e^{\alpha \cdot \beta} \Big|_{\alpha,\beta=0} \\ &= \frac{\partial}{\partial \alpha_{a_1 b_1}} \cdots \frac{\partial}{\partial \alpha_{a_d b_d}} (\alpha_{a_{\sigma(1)} b_1} \cdots \alpha_{a_{\sigma(d)} b_d}) \Big|_{\alpha=0}. \end{aligned}$$

This last step should look familiar from Theorem 10.2.1. As we did there, we now evaluate this expression by expanding out the multi-derivative. Each pairing of derivatives to α terms – one for each d-permutation g in Sym_d – produces a product of Kronecker deltas:

$$P_{d_1,\ldots,d_N}(m_1,m_2) = \sum_{g \in Sym_d} \delta[a_1, a_{\sigma g(1)}] \delta[b_1, b_{g(1)}] \cdots \delta[a_d, a_{\sigma g(d)}] \delta[b_d, b_{g(d)}]. \quad (11.2.2)$$

Each such term then contracts down to a monomial in m_1 and m_2, where the exponent of m_1 is the number of cycles in σg and the exponent of m_2 is the number of cycles in g.

But we know that these terms can also be represented in terms of ladder diagrams. The generalisation of the ladder diagrams used in Theorem 10.2.1 is simple, as it was in Figure 11.1.1; we just replace the dashed "backbone" lines with those given by our new, multi-cycled σ. The number of dashed loops then equals the number of cycles in σg, and the number of solid loops equals the number of cycles in g. There is a one-to-one correspondence between d-permutations g and ladder diagrams with backbone σ, so $P_{d_1,\ldots,d_N}(m_1,m_2)$ is a generating function enumerating them by number of solid and dashed loops. □

This lemma gives us what we need to be able to evaluate these generating functions, as it relates the $P_{d_1,\ldots,d_N}(m_1,m_2)$ to matrix integrals of the type we used in Part II, and we have already looked at how to evaluate such integrals in Chapter 5. Of course it is the generating functions for rooted hypermaps that we are after, not those for ladder diagrams, so we will save this evaluation work for later. Next we will see how to construct the hypermap generating functions themselves using $P_{d_1,\ldots,d_N}(m_1,m_2)$.

11.2.1 Eliminating over-counting

Any rooted hypermap can be represented as a ladder diagram. This can be seen quite easily through construction: represent the hypermap as the constellation $[\sigma, g, (\sigma g)^{-1}]$, then make a ladder diagram with σ as the backbone and g as the permutation. The number of vertices is the number of dashed loops in the diagram, the number of edges the number of solid loops, and the number of faces the number of cycles in the backbone. Summing over all rooted hypermaps is not as simple as summing over all

ladder diagrams, however, as we have already seen in Section 11.1. There are effectively three factors which cause over-counting:

- *Cyclic degeneracy of non-rooted cycles* – We saw in the two-face case that there is a degeneracy caused by isomorphisms which cycle the indices in the second cycle of σ (Appendix E). With higher face counts, the same is true for each non-rooted cycle in σ (all except the first, as we are still using the convention of associating the root with the label 1), so if the cycle lengths are $d_1, \ldots, d_N$, the degeneracy is $d_2 d_3 \cdots d_N$.
- *Ordering degeneracy of non-rooted cycles* – Isomorphisms which simply reorder the non-rooted cycles produce a second degeneracy. For instance, any rooted hypermap counted in $P_{d_1,d_2,d_3}(m_1, m_2)$ will also be counted in $P_{d_1,d_3,d_2}(m_1, m_2)$. In a rooted hypermap (where the ladder diagram is connected) the cycles of σ must be linked together fully by lines coming from g, and as no two lines can connect to the same cycle at the same point all of these inter-cycle links are distinguishable from each other. Thus, any non-trivial reordering of the non-rooted cycles will result in a distinct ladder diagram, meaning that the degeneracy in each case is $(N-1)!$, the number of orderings of the non-rooted cycles.
- *Non-transitive cases* – In the two-face case we saw that the function $P_{i,j}(m_1, m_2)$ counts non-transitive cases (corresponding to disconnected ladder diagrams). These by definition are not hypermaps, but we were able to subtract them away by treating disconnected ladder diagrams as products of two smaller connected diagrams. The same principle applies with higher face counts.

We must account for the non-transitive cases first. We again do so by factoring and subtracting out the disconnected ladder diagrams, but to do so for general face counts we use a recursive method:

Lemma 11.2.2. *Let $\bar{P}_{d_1,\ldots,d_N}(m_1, m_2)$ be a generating function corresponding to the same sum over ladder diagrams as $P_{d_1,\ldots,d_N}(m_1, m_2)$, but only including connected ladder diagrams. Then for any N and $d_1, \ldots, d_N$,*

$$P_{d_1,\ldots,d_N}(m_1, m_2) = \sum_{u \subset \{d_2,\ldots,d_N\}} \bar{P}_{d_1,u_1,u_2,\ldots}(m_1, m_2) P_{\bar{u}_1,\bar{u}_2,\ldots}(m_1, m_2), \tag{11.2.3}$$

where the sum is over all subsets u of $\{d_2, \ldots, d_N\}$ including the empty set and $\{d_2, \ldots, d_N\}$ itself, and $\bar{u}$ is the complement of u. When $\bar{u}$ is empty, we define the special case $P(m_1, m_2) = 1$.

Proof. This is a simple factorisation of the sum over ladder diagrams. We divide each ladder into two parts – a connected ladder which contains the rooted cycle and a (potentially disconnected, potentially empty) ladder containing everything else. The $\bar{P}_{d_1,u_1,u_2,\ldots}(m_1, m_2)$ term enumerates the possible configurations of the connected, rooted part, while the $P_{\bar{u}_1,\bar{u}_2,\ldots}(m_1, m_2)$ term enumerates the rest. □

This relation is rather cumbersome, particularly given the nature of the summation. However, for our purposes it can be simplified massively, by defining three *helper functions*:

Definition 11.2.1.

$$\Pi_d^{(N)}(m_1,m_2;x) = \sum_{d_1=1}^{\infty}\frac{x^{d_1}}{d_1}\cdots\sum_{d_N=1}^{\infty}\frac{x^{d_N}}{d_N}P_{d,d_1,\ldots,d_N}(m_1,m_2)$$

$$\bar{\Pi}_d^{(N)}(m_1,m_2;x) = \sum_{d_1=1}^{\infty}\frac{x^{d_1}}{d_1}\cdots\sum_{d_N=1}^{\infty}\frac{x^{d_N}}{d_N}\bar{P}_{d,d_1,\ldots,d_N}(m_1,m_2) \tag{11.2.4}$$

$$\Sigma^{(N)}(m_1,m_2;x) = \sum_{d_1=1}^{\infty}\frac{x^{d_1}}{d_1}\cdots\sum_{d_N=1}^{\infty}\frac{x^{d_N}}{d_N}P_{d_1,\ldots,d_N}(m_1,m_2), \tag{11.2.5}$$

with the special cases $\Pi_d^{(0)}(m_1,m_2;x) = \bar{\Pi}_d^{(0)}(m_1,m_2;x) = P_d(m_1,m_2)$ and $\Sigma^{(0)}(m_1,m_2;x) = 1$.

Lemma 11.2.3.

$$\Pi_d^{(N)}(m_1,m_2;x) = \sum_{k=0}^{N}\binom{N}{k}\bar{\Pi}_d^{(k)}(m_1,m_2;x)\Sigma^{(N-k)}(m_1,m_2;x). \tag{11.2.6}$$

Proof. To arrive at this identity, first we perform a slight relabelling of the indices in (11.2.3):

$$P_{d,d_1,\ldots,d_N}(m_1,m_2) = \sum_{u\subset\{d_1,\ldots,d_N\}}\bar{P}_{d,u_1,u_2,\ldots}(m_1,m_2)P_{\bar{u}_1,\bar{u}_2,\ldots}(m_1,m_2).$$

Now we sum over each of the d_i, with an additional factor of x^{d_i}/d_i in each sum. The left hand side simply becomes $\Pi_d^{(N)}(m_1,m_2;x)$, and if u has k elements in it, the summand in the right hand side becomes $\bar{\Pi}_d^{(k)}(m_1,m_2;x)\Sigma^{(N-k)}(m_1,m_2;x)$. As this only depends on k and not the specific makeup of u, we can collect together all possible u with the same length, of which there are $\binom{N}{k}$. Therefore,

$$\Pi_d^{(N)}(m_1,m_2;x) = \sum_{k=0}^{N}\binom{N}{k}\bar{\Pi}_d^{(k)}(m_1,m_2;x)\Sigma^{(N-k)}(m_1,m_2;x).$$

□

This recursion relation is much simpler than (11.2.3). While it contains less information (you cannot reconstruct $\bar{P}_{d_1,\ldots,d_N}(m_1,m_2)$ from $\bar{\Pi}_d^{(k)}(m_1,m_2;x)$), it does contain all the information required to build generating functions for rooted hypermaps, as we will see in the next section. The fact that these helper functions are connected through a recursion relation is a clear indicator of the fact that our final generating functions will also be defined using recursions.

11.2.2 Building the generating function

While our stated aim was to calculate the functions $H_d(m_0, m_1, m_2)$ enumerating rooted hypermaps for fixed dart count, it is actually a lot easier to define and work with the global generating function $H(m_0, m_1, m_2; x)$ which enumerates all dart counts simultaneously. This is hinted at by the form of the helper functions we defined in the last section, which feature terms like $x^{d_1+\cdots+d_N}$ (the exponent here corresponds to a dart count on a hypermap). Having defined $\bar{P}_{d_1,\ldots,d_N}(m_1, m_2)$, we now have all the tools necessary to state an expression for this function. We will then return to the $H_d(m_0, m_1, m_2)$ in Section 11.2.4.

Let us define $H(m_0, m_1, m_2; x)$ as a function which enumerates all rooted hypermaps, partitioning them by number of faces, edges, vertices and darts. Additionally, we shall say that it is an alternating generating function in x (the parameter which partitions by number of darts), as this will help make the resulting integral expressions converge. This function is given by

$$H(m_0, m_1, m_2; x) = \sum_{N=1}^{\infty} \frac{m_0^N}{(N-1)!} \sum_{d_1=1}^{\infty} (-x)^{d_1} \times \sum_{d_2=1}^{\infty} \frac{(-x)^{d_2}}{d_2} \cdots \sum_{d_N=1}^{\infty} \frac{(-x)^{d_N}}{d_N} \bar{P}_{d_1,\ldots,d_N}(m_1, m_2). \tag{11.2.7}$$

This includes the term $(-x)^{d_1+\ldots+d_N}$ to enumerate by dart count and m_0^N to enumerate by face count, and is divided by the factor $(N-1)!d_2 \cdots d_N$ to account for the two types of degeneracy associated with the unrooted cycles. By comparing the above with (11.2.4) we see that

$$\begin{aligned} H(m_0, m_1, m_2; x) &= \sum_{N=1}^{\infty} \frac{m_0^N}{(N-1)!} \sum_{d_1=1}^{\infty} (-x)^{d_1} \bar{\Pi}_{d_1}^{(N-1)}(m_1, m_2; -x) \\ &= \sum_{N=0}^{\infty} \frac{m_0^{N+1}}{N!} \sum_{d=1}^{\infty} (-x)^{d} \bar{\Pi}_{d}^{(N)}(m_1, m_2; -x). \end{aligned}$$

Now that we have $H(m_0, m_1, m_2; x)$ in terms of our helper functions, we can use (11.2.6) to eliminate the dependence on the transitive-only functions $\bar{\Pi}_d^{(N)}(m_1, m_2; x)$ and $\bar{P}_{d_1,\ldots,d_N}(m_1, m_2)$ in favour of the easier-to-calculate non-transitive equivalents.

Theorem 11.2.1. *Let*

$$F(m_0, m_1, m_2; x) = \sum_{N=0}^{\infty} \frac{m_0^N}{N!} \Sigma^{(N)}(m_1, m_2; -x). \tag{11.2.8}$$

Then

$$H(m_0, m_1, m_2; x) F(m_0, m_1, m_2; x) = x \frac{\partial}{\partial x} F(m_0, m_1, m_2; x). \tag{11.2.9}$$

Proof. First, consider the right hand side.

$$\begin{aligned}
x\frac{\partial}{\partial x}F(m_0,m_1,m_2;x) &= x\frac{\partial}{\partial x}\sum_{N=0}^{\infty}\frac{m_0^N}{N!}\Sigma^{(N)}(m_1,m_2;-x)\\
&= x\frac{\partial}{\partial x}\sum_{N=0}^{\infty}\frac{m_0^N}{N!}\sum_{d_1=1}^{\infty}\frac{(-x)^{d_1}}{d_1}\cdots\sum_{d_N=1}^{\infty}\frac{(-x)^{d_N}}{d_N}P_{d_1,\ldots,d_N}(m_1,m_2)\\
&= N\sum_{N=0}^{\infty}\frac{m_0^N}{N!}\sum_{d_1=1}^{\infty}(-x)^{d_1}\\
&\quad\times\sum_{d_2=1}^{\infty}\frac{(-x)^{d_2}}{d_2}\cdots\sum_{d_N=1}^{\infty}\frac{(-x)^{d_N}}{d_N}P_{d_1,\ldots,d_N}(m_1,m_2).
\end{aligned}$$

The factor of N comes from the fact that any of the N different summations could be moved to the front and differentiated first. Using our helper functions we can now simplify this expression again:

$$\begin{aligned}
x\frac{\partial}{\partial x}F(m_0,m_1,m_2;x) &= \sum_{N=0}^{\infty}\frac{m_0^N}{(N-1)!}\sum_{d_1=1}^{\infty}(-x)^{d_1}\Pi_d^{(N-1)}(m_1,m_2;-x)\\
&= \sum_{N=0}^{\infty}\frac{m_0^{N+1}}{N!}\sum_{d=1}^{\infty}(-x)^{d}\Pi_d^{(N)}(m_1,m_2;-x).
\end{aligned}$$

Now consider the left hand side of (11.2.9). When we expand it in terms of helper functions, we get

$$\begin{aligned}
H(m_0,m_1,m_2;x)F(m_0,m_1,m_2;x) &= \sum_{k=0}^{\infty}\frac{m_0^{k+1}}{k!}\sum_{d=1}^{\infty}(-x)^{d}\bar{\Pi}_d^{(k)}(m_1,m_2;-x)\\
&\quad\times\sum_{N=0}^{\infty}\frac{m_0^N}{N!}\Sigma^{(N)}(m_1,m_2;-x)\\
&= \sum_{k=0}^{\infty}\sum_{N=k}^{\infty}\frac{m_0^{N+1}}{k!(N-k)!}\sum_{d=1}^{\infty}(-x)^{d}\\
&\quad\times\bar{\Pi}_d^{(k)}(m_1,m_2;-x)\Sigma^{(N-k)}(m_1,m_2;-x)\\
&= \sum_{N=0}^{\infty}\frac{m_0^{N+1}}{N!}\sum_{d=1}^{\infty}(-x)^{d}\sum_{k=0}^{N}\binom{N}{k}\\
&\quad\times\bar{\Pi}_d^{(k)}(m_1,m_2;-x)\Sigma^{(N-k)}(m_1,m_2;-x).
\end{aligned}$$

At this point the sum over k is exactly (11.2.6), so we can substitute this in, getting

$$\begin{aligned}
H(m_0,m_1,m_2;x)F(m_0,m_1,m_2;x) &= \sum_{N=0}^{\infty}\frac{m_0^{N+1}}{N!}\sum_{d=1}^{\infty}(-x)^{d}\Pi_d^{(N)}(m_1,m_2;-x)\\
&= x\frac{\partial}{\partial x}F(m_0,m_1,m_2;x).
\end{aligned}$$

□

$F(m_0, m_1, m_2; x)$ depends only on the non-transitive functions $\Sigma^{(N)}(m_1, m_2; x)$, which are much better suited for evaluation that the transitive equivalent functions would be, as we will see in the next section. Thus we see that the best way for us to evaluate $H(m_0, m_1, m_2; x)$ is to first evaluate $F(m_0, m_1, m_2; x)$ and infer it from that using (11.2.9).

It's worth noting that we can write (11.2.9) as

$$H(m_0, m_1, m_2; x) = x\frac{\partial}{\partial x} \ln[F(m_0, m_1, m_2; x)],$$

but we will find as we evaluate $F(m_0, m_1, m_2; x)$ that it is not easy to take the logarithm of the resulting expression. The most appropriate method for evaluating these functions is in fact to treat them both as formal power series in x, and use (11.2.9) to relate the terms in these series to each other recursively, as we will discuss in Section 11.2.4.

11.2.3 Evaluating $F(m_0, m_1, m_2; x)$

To evaluate $F(m_0, m_1, m_2; x)$, we now go back to the functions $P_{d_1,\dots,d_N}(m_1, m_2)$ defined in (11.2.1), in terms of which all the functions used in the previous few sections have been defined. We initially defined $P_{d_1,\dots,d_N}(m_1, m_2)$ using matrix integrals before proving their combinatorial interpretations, and we can now use them to find an integral expression for $F(m_0, m_1, m_2; x)$.

To do this we need to return again to methods we used in Part II for manipulating such matrix integrals. We have already used the Gaussian integration methods introduced in Section 5.1 in proving Lemma 11.2.1, but this one of two approaches we've used, the other being the eigenvalue-integral method from Section 5.2. This method allowed us to evaluate the one-face rooted hypermap generating functions in closed form, and it will be similarly useful here.

Theorem 11.2.2. *For fixed m_1 and m_2,*

$$F(m_0, m_1, m_2; x) \propto \int \prod_{k=1}^{m_1} \frac{q_k^{m_2-m_1} e^{-q_k} dq_k}{(1+q_k x)^{m_0}} \prod_{1\le i<j\le m_1} (q_j - q_i)^2.$$

Proof. We begin by with (11.2.1):

$$P_{d_1,\dots,d_N}(m_1, m_2) = \frac{\Gamma(m_1 m_2 + d)}{2\pi^{m_1 m_2}} \int_{S^{2m_1 m_2 - 1}} d\Omega \prod_{r=1}^{N} \mathrm{Tr}[(\hat{\rho}_1^{12})^{d_r}],$$

noting that the integrand here is order d in $\hat{\rho}_1^{12}$, and can be expressed entirely in terms of the eigenvalues of $\hat{\rho}_1^{12}$. As we saw in Section 5.2, we can write this as an integral over these eigenvalues [21]:

$$P_{d_1,\dots,d_N}(m_1, m_2) = \frac{\Gamma(m_1 m_2 + d)}{\Gamma(m_1 m_2)} \int dp_1 \cdots dp_{m_1} P(p_1, \dots, p_{m_1}) \prod_{r=1}^{N} \left(\sum_{a_r=1}^{m_1} p_{a_r}^{d_r} \right),$$

where the normalised joint density function $P(p_1, \ldots, p_{m_1})$ is defined in 5.2.2. Note the prefactor, which comes from the fact that the normalised volume element for the spherical integral is $\frac{\Gamma(m_1 m_2)}{2\pi^{m_1 m_2}} d\Omega$. Furthermore we can use Page's rescaling method to simplify this further [24] (see Section 5.2.1):

$$\begin{aligned} P_{d_1,\ldots,d_N}(m_1, m_2) &= \frac{1}{Z^*_{m_1,m_2}\Gamma(m_1 m_2)} \int \prod_{k=1}^{m_1} q_k^{m_2-m_1} e^{-q_k} dq_k \\ &\quad \times \prod_{1\le i<j\le m_1} (q_j - q_i)^2 \prod_{r=1}^{N} \left(\sum_{a_r=1}^{m_1} q_{a_r}^{d_r} \right) \\ &= \frac{1}{\Lambda} \int \prod_{k=1}^{m_1} q_k^{m_2-m_1} e^{-q_k} dq_k \\ &\quad \times \prod_{1\le i<j\le m_1} (q_j - q_i)^2 \prod_{r=1}^{N} \left(\sum_{a_r=1}^{m_1} q_{a_r}^{d_r} \right). \end{aligned} \tag{11.2.10}$$

$Z^*_{m_1,m_2}$ is the same normalising factor that appeared in Section 5.2.1, but from now on for the sake of readability we adopt the symbol Λ to collect up the entire x-independent prefactor, as this term will appear in a lot of places in the following work. In this case the normalisation is such that $P_{0,\ldots,0}(m_1, m_2) = m_1^N$, as follows from (11.2.1).

Now we substitute this into (11.2.5). Each sum over d_r gets paired with the sum over a_r in (11.2.10). The resulting N double sums are indistinguishable from each other so can be grouped together:

$$\Sigma^{(N)}(m_1, m_2; x) = \frac{1}{\Lambda} \int \prod_{k=1}^{m_1} q_k^{m_2-m_1} e^{-q_k} dq_k \prod_{1\le i<j\le m_1} (q_j - q_i)^2 \left(\sum_{d=1}^{\infty} \frac{x^d}{d} \sum_{a=1}^{m_1} q_a^d \right)^N.$$

Substituting this into (11.2.8), we get

$$\begin{aligned} F(m_0, m_1, m_2; x) = \frac{1}{\Lambda} \int \prod_{k=1}^{m_1} q_k^{m_2-m_1} e^{-q_k} dq_k \prod_{1\le i<j\le m_1} (q_j - q_i)^2 \\ \times \sum_{N=0}^{\infty} \frac{m_0^N}{N!} \left(\sum_{d=1}^{\infty} \frac{(-x)^d}{d} \sum_{a=1}^{m_1} q_a^d \right)^N. \end{aligned}$$

We have to be careful with our manipulation of these summations, as they don't converge for all possible values of $q_1, \ldots, q_{m_1}$. However, Lemma A.0.3 in Appendix A gives us that, when treated as formal power series in x,

$$\sum_{N=0}^{\infty} \frac{m_0^N}{N!} \left(\sum_{d=1}^{\infty} \frac{(-x)^d}{d} \sum_{a=1}^{m_1} q_a^d \right)^N \equiv \prod_{a=1}^{m_1} \frac{1}{(1+q_a x)^{m_0}}.$$

Therefore,

$$F(m_0, m_1, m_2; x) = \frac{1}{\Lambda} \int \prod_{k=1}^{m_1} \frac{q_k^{m_2-m_1} e^{-q_k} dq_k}{(1+q_k x)^{m_0}} \prod_{1\le i<j\le m_1} (q_j - q_i)^2. \tag{11.2.11}$$

This eliminates the non-convergence problem entirely, as the resultant integrals converge absolutely as long as x is positive[2]. □

We now want to evaluate this expression in a closed form, as we did in Section 5.2.1. We have to depart slightly from the method used there, however. Previously we used a method developed by Sen, where we rewrote the Vandermonde discriminant term

$$\prod_{1\leq i<j\leq m_1} (q_j - q_i)^2$$

in terms of associated Laguerre polynomials, allowing us to separate out the m_1 different integrals and evaluate each separately [30]. That method relied on the fact that the integrals were weighted by $q_k^{m_2-m_1}e^{-q_k}$, the weight function for the associated Laguerre polynomials; it was due to the orthogonality of the polynomials that the integrals became separable.

In (11.2.11), however, the weight functions are now

$$\frac{q_k^{m_2-m_1}e^{-q_k}}{(1+q_kx)^{m_0}}.$$

We can still use Sen's basic method, but as the weight function has changed, the required set of orthogonal polynomials will also be different. It is incredibly difficult to evaluate these polynomials in any exact form, but as it happens we are able to infer all of their necessary properties (see Appendix G). Using these results, we can now evaluate the integrals in (11.2.11).

Theorem 11.2.3. *When expressed as a formal power series in x,*

$$\begin{aligned} F(m_0,m_1,m_2;x) &= \sum_{k_0=0}^{\infty}\cdots\sum_{k_{m_1-1}=0}^{\infty}\prod_{1\leq i<j\leq m_1}\left(1-\frac{k_j-k_i}{j-i}\right) \\ &\quad \prod_{a=0}^{m_1-1}\frac{(m_0)_{k_a}(m_2-a)_{k_a}}{k_a!}(-x)^{k_a}. \end{aligned} \tag{11.2.12}$$

Proof. For known m_0, m_1, m_2 and x, let

$$w(q) = \frac{q^{m_2-m_1}e^{-q}}{(1+qx)^{m_0}}.$$

Treating this as a weight function on the interval $[0,\infty)$, its moments, expressed as

[2]This requirement is why we defined $H(m_0,m_1,m_2;x)$ to be an *alternating* power series in x – so that the domain in which the integral converges would be the positive-x domain.

power series in x, are

$$\begin{aligned}\mu_n &= \int_0^\infty q^n \frac{q^{m_2-m_1}e^{-q}}{(1+qx)^{m_0}}dq \\ &= \sum_{k=0}^{\infty} \frac{\Gamma(m_0+k)}{k!\Gamma(m_0)}(-x)^k \int_0^\infty q^{m_2-m_1+n+k}e^{-q}dq \\ &= \sum_{k=0}^{\infty} \frac{(m_2-m_1+n+k)!\Gamma(m_0+k)}{k!\Gamma(m_0)}(-x)^k. \end{aligned} \tag{11.2.13}$$

Let us define the orthogonal polynomials $K_n(q)$ as order-n monic polynomials in q, satisfying the orthogonality relation

$$\int_0^\infty K_i(q)K_j(q)w(q)dq = I_i\delta_{i,j}$$

where

$$I_n = \int_0^\infty [K_n(q)]^2 w(q)dq.$$

As Sen remarks [30], we can write the Vandermonde discriminant in terms of these polynomials:

$$\prod_{1\le i<j\le m_1}(q_j-q_i)^2 = \begin{vmatrix} K_0(q_1) & K_1(q_1) & \cdots & K_{m_1-1}(q_1) \\ K_0(q_2) & K_1(q_2) & \cdots & K_{m_1-1}(q_2) \\ \vdots & \vdots & \ddots & \vdots \\ K_0(q_{m_1}) & K_1(q_{m_1}) & \cdots & K_{m_1-1}(q_{m_1}) \end{vmatrix}^2.$$

When we substitute this into (11.2.11) and expand the determinants out, we get

$$F(m_0,m_1,m_2;x) = \frac{\epsilon_{i_1\cdots i_{m_1}}\epsilon_{j_1\cdots j_{m_1}}}{\Lambda}\prod_{k=1}^{m_1}\int_0^\infty K_{i_k}(q_k)K_{j_k}(q_k)w(q_k)dq_k,$$

where we sum over each of the i and j indices from 0 to m_1-1, and ϵ is the Levi-Civita symbol. The integrals are now exactly the orthogonality relation for our polynomials, so only cases where $i_k = j_k$ for all k contribute. Counting up all such cases we get that

$$\begin{aligned} F(m_0,m_1,m_2;x) &= \frac{m_1!}{\Lambda}\prod_{k=1}^{m_1}\int_0^\infty K_{k-1}(q)K_{k-1}(q)w(q)dq \\ &= \frac{m_1!}{\Lambda}\prod_{k=1}^{m_1} I_{k-1}. \end{aligned} \tag{11.2.14}$$

Using Corollary F.0.1 from Appendix G this becomes

$$F(m_0,m_1,m_2;x) = \frac{m_1!}{\Lambda}|\mathbf{M}_{m_1}|$$

where

$$\mathbf{M}_{m_1} = \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{m_1-1} \\ \mu_1 & \mu_2 & \cdots & \mu_{m_1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{m_1-1} & \mu_{m_1} & \cdots & \mu_{2m_1-2} \end{bmatrix}.$$

We now substitute (11.2.13) into this. By writing all the terms in the first column as sums over the same variable, k_{m_1}, then all the terms in the second column sums over k_{m_1-1} etc. we make it so that all of the k-sums can be factored out of the determinant. This gives us

$$\begin{aligned} F(m_0, m_1, m_2; x) &= \frac{m_1!}{\Lambda} \sum_{k_1=0}^{\infty} \frac{\Gamma(m_0+k_1)}{k_1!\Gamma(m_0)}(-x)^{k_1} \cdots \sum_{k_{m_1}=0}^{\infty} \frac{\Gamma(m_0+k_{m_1})}{k_{m_1}!\Gamma(m_0)}(-x)^{k_{m_1}} \\ &\quad \times \Phi(k_1, \ldots, k_{m_1}) \end{aligned} \tag{11.2.15}$$

where

$$\Phi(k_1, \ldots, k_{m_1}) = \begin{vmatrix} (\alpha+k_{m_1})! & (\alpha+k_{m_1-1}+1)! & \cdots & (\alpha+k_1+m_1-1)! \\ (\alpha+k_{m_1}+1)! & (\alpha+k_{m_1-1}+2)! & \cdots & (\alpha+k_{_1}+m_1)! \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha+k_{m_1}+m_1-2)! & (\alpha+k_{m_1-1}+m_1-1)! & \cdots & (\alpha+k_{_1}+2m_1-3)! \\ (\alpha+k_{m_1}+m_1-1)! & (\alpha+k_{m_1-1}+m_1)! & \cdots & (\alpha+k_{_1}+2m_1-2)! \end{vmatrix} \tag{11.2.16}$$

and $\alpha = m_2 - m_1$.

We now simplify this determinant through a series of row operations. First we subtract $(\alpha+1)$ times the first row from the second, $(\alpha+2)$ times the second row from the third etc. to get

$$\Phi(k_1, \ldots, k_{m_1}) = \begin{vmatrix} (\alpha+k_{m_1})! & \cdots & (\alpha+k_1+m_1-1)! \\ k_{m_1}(\alpha+k_{m_1})! & \cdots & (k_1+m_1-1)(\alpha+k_1+m_1-1)! \\ \vdots & \ddots & \vdots \\ k_{m_1}(\alpha+k_{m_1}+m_1-3)! & \cdots & (k_1+m_1-1)(\alpha+k_1+2m_1-4)! \\ k_{m_1}(\alpha+k_{m_1}+m_1-2)! & \cdots & (k_1+m_1-1)(\alpha+k_1+2m_1-3)! \end{vmatrix}.$$

Next we subtract $(\alpha+1)$ times the second row from the third, $(\alpha+2)$ times the third from the fourth etc. We repeat this process a total of m_1-1 times, reducing the number of row operations in each step by one each time. When all the steps are finished we have

$$\Phi(k_1, \ldots, k_{m_1}) = \begin{vmatrix} (\alpha+k_{m_1})! & \cdots & (\alpha+k_1+m_1-1)! \\ k_{m_1}(\alpha+k_{m_1})! & \cdots & (k_1+m_1-1)(\alpha+k_1+m_1-1)! \\ \vdots & \ddots & \vdots \\ k_{m_1}^{m_1-2}(\alpha+k_{m_1})! & \cdots & (k_1+m_1-1)^{m_1-2}(\alpha+k_1+m_1-1)! \\ k_{m_1}^{m_1-1}(\alpha+k_{m_1})! & \cdots & (k_1+m_1-1)^{m_1-1}(\alpha+k_1+m_1-1)! \end{vmatrix}$$

and therefore

$$\begin{aligned}\Phi(k_1,\ldots,k_{m_1}) &= \Delta(k_{m_1}, k_{m_1-1}+1,\ldots,k_1+m_1-1)\\ &\quad\times\prod_{a=1}^{m_1}(m_2-m_1+k_{m_1+1-a}+a-1)!\\ &= \prod_{1\le j<i\le m_1}(k_{m_1+1-i}-k_{m_1+1-j}+i-j)\\ &\quad\times\prod_{a=1}^{m_1}(m_2-m_1+k_{m_1+1-a}+a-1)!.\end{aligned}$$

Finally we reverse the order of the products, by substituting $a\to m_1+1-a$, $i\to m_1+1-i$ and $j\to m_1+1-j$. This gives

$$\Phi(k_{m_1},\ldots,k_1)=\prod_{1\le i<j\le m_1}(j-i+k_i-k_j)\prod_{a=1}^{m_1}(m_2+k_a-a)!. \tag{11.2.17}$$

We now substitute this into (11.2.15) to get

$$\begin{aligned}F(m_0,m_1,m_2;x) = \frac{m_1!}{\Lambda}&\sum_{k_1=0}^{\infty}\frac{\Gamma(m_0+k_1)(m_2+k_1-1)!}{k_1!\Gamma(m_0)}(-x)^{k_1}\\ &\cdots\sum_{k_{m_1}=0}^{\infty}\frac{\Gamma(m_0+k_{m_1})(m_2+k_{m_1}-m_1)!}{k_{m_1}!\Gamma(m_0)}(-x)^{k_{m_1}}\\ &\times\prod_{1\le i<j\le m_1}(j-i+k_i-k_j).\end{aligned} \tag{11.2.18}$$

We are able to fix the normalisation constant at this point. Choosing $x=0$, (11.2.8) becomes

$$F(m_0,m_1,m_2;0)=\sum_{N=0}^{\infty}\frac{m_0^N}{N!}\Sigma^{(N)}(m_1,m_2;0)=1 \tag{11.2.19}$$

As the only $\Sigma^{(N)}(m_1,m_2;x)$ which doesn't go to zero as $x\to 0$ is $\Sigma^{(0)}(m_1,m_2;x)=1$. Putting this into (11.2.18) gives

$$1=\frac{m_1!}{\Lambda}\prod_{a=1}^{m_1}(m_2-a)!\prod_{1\le i<j\le m_1}(j-i).$$

This fixes Λ exactly, and gives

$$\begin{aligned}F(m_0,m_1,m_2;x) = \sum_{k_1=0}^{\infty}\cdots\sum_{k_{m_1}=0}^{\infty}&\prod_{1\le i<j\le m_1}\left(1-\frac{k_j-k_i}{j-i}\right)\\ &\times\prod_{a=1}^{m_1}\frac{(m_0)_{k_a}(m_2-a+1)_{k_a}}{k_a!}(-x)^{k_a}.\end{aligned}$$

For simplicity let us shift all of the k indices down by one:

$$F(m_0, m_1, m_2; x) = \sum_{k_0=0}^{\infty} \cdots \sum_{k_{m_1-1}=0}^{\infty} \prod_{0 \le i < j < m_1} \left(1 - \frac{k_j - k_i}{j - i}\right) \times \prod_{a=0}^{m_1-1} \frac{(m_0)_{k_a}(m_2 - a)_{k_a}}{k_a!}(-x)^{k_a}.$$

□

This expression shares some similarities with the one-face generating functions $P_d(m_1, m_2)$ given in (8.0.2), in particular the rising factorials, which produce polynomial functions in m_0 and m_2. However, there is a key difference. We were able to evaluate the $P_d(m_1, m_2)$ exactly as smooth functions of m_1 and m_2 – this was important as the values we were interested in, representing counts of rooted hypermaps, were given by the power-series coefficients of these functions. We can't do this here, however, for two reasons. The first is that (11.2.12) includes infinite sums, so we cannot evaluate it in finite time. The second is that the limits of the products are dependent on m_1, meaning we can only evaluate them for integer values of m_1 and can't directly come up with smooth functions.

Both of these problems can be dealt with, however, as we will see in the next section. We can avoid the infinite sums simply by looking at the each power of x one at a time, while the lack of continuity in m_1 can be avoided by evaluating at a number of specific values of m_1 and interpolating, using the knowledge that the generating functions being evaluated must have polynomial dependence on m_1.

11.2.4 Evaluating $H(m_0, m_1, m_2; x)$

This summation expression (11.2.12) for $F(m_0, m_1, m_2; x)$ is as close as we are going to get to a simple explicit formula for $H(m_0, m_1, m_2; x)$. But while we may not be able to find any simpler, more direct expressions for $H(m_0, m_1, m_2; x)$, like when we found (8.0.2) for $P_d(m_1, m_2)$, we can use what we have now to evaluate terms in $H(m_0, m_1, m_2; x)$ and extract our hypermap counts from them. In this section we will look in detail at the method for doing this.

Our first step is to break up the global generating function back up into the simpler generating functions $H_d(m_0, m_1, m_2)$, corresponding to enumerating with fixed numbers of darts. We first discussed these functions right at the start of the chapter, and they relate to the global function through the identity

$$H(m_0, m_1, m_2; x) = \sum_{d=1}^{\infty} (-x)^d H_d(m_0, m_1, m_2). \tag{11.2.20}$$

This may seem like a step backwards, after having dealt with functions that count all numbers of darts at the same time, but there is good reason for using both types of functions in conjunction. The manipulations which led to (11.2.12) were only possible

because we dealt with all degrees at once; if we had limited ourselves to one degree at a time the resulting expressions would have been prohibitively complicated. Now we have these expressions, however, the functions $H_d(m_0, m_1, m_2)$ are much better suited for the final evaluation steps as each one is a finite-order polynomial, the coefficients of which we are able to compute in finite time.

To compute each $H_d(m_0, m_1, m_2)$ we need to break up $F(m_0, m_1, m_2; x)$ in a similar manner, by saying that

$$F(m_0, m_1, m_2; x) = \sum_{d=0}^{\infty} (-x)^d F_d(m_0, m_1, m_2).$$

Note that we have a zeroth-order term this time, unlike in (11.2.20), where the zeroth-order term is absent due to the fact that there are no rooted hypermaps with zero darts. We already know that $F(m_0, m_1, m_2)$ has a zeroth-order term, however, from (11.2.19).

We now use the relation (11.2.9) to link these two sequences:

$$\begin{aligned}\sum_{d'=1}^{\infty} (-x)^{d'} H_{d'}(m_0, m_1, m_2) \sum_{d=0}^{\infty} (-x)^d F_d(m_0, m_1, m_2) &= x \frac{\partial}{\partial x} \sum_{d=0}^{\infty} (-x)^d F_d(m_0, m_1, m_2) \\ &= \sum_{d=1}^{\infty} d(-x)^d F_d(m_0, m_1, m_2).\end{aligned}$$

Matching together powers of $(-x)$ on both sides we get

$$\sum_{k=1}^{d} H_k(m_0, m_1, m_2) F_{d-k}(m_0, m_1, m_2) = d F_d(m_0, m_1, m_2). \tag{11.2.21}$$

Thus, if we can calculate each $F_d(m_0, m_1, m_2)$, we can recursively calculate each $H_d(m_0, m_1, m_2)$ as well.

Evaluating the $F_d(m_0, m_1, m_2)$ requires collecting together all terms in (11.2.12) of the same order in $(-x)$. The result is

$$\begin{aligned}F_d(m_0, m_1, m_2) = \sum_{\substack{k_0, \ldots, k_{m_1-1} \geq 0 \\ k_0 + \ldots + k_{m_1-1} = d}} & \prod_{0 \leq i < j < m_1} \left(1 - \frac{k_j - k_i}{j - i}\right) \\ & \times \prod_{a=0}^{m_1-1} \frac{(m_0)_{k_a} (m_2 - a)_{k_a}}{k_a!},\end{aligned} \tag{11.2.22}$$

where the sum is over all partitions of d into m_1 non-negative integers.

As we have already noted, this expression can only be evaluated for fixed positive integers m_1, although when m_1 is known it can then be evaluated as a polynomial in m_0 and m_2. With the knowledge that $F_d(m_0, m_1, m_2)$ *has* a symmetric polynomial form[3],

[3]We know that $H_k(m_0, m_1, m_2)$ has a symmetric polynomial form, and that $F_0(m_0, m_1, m_2) = 1$. That all $F_d(m_0, m_1, m_2)$ are also symmetric polynomials then follows by induction from (11.2.21).

however, we can infer this form by evaluating $F_d(m_0, m_1, m_2)$ at a number of specific values of m_1 and then interpolating from these. We can infer the order in m_1 by finding the order in m_0 – each term in the summation in (11.2.22) goes as m_0^d for large m_0, so it is at most order d in m_1 – and therefore it is sufficient to evaluate $F_d(m_0, m_1, m_2)$ at all $0 \leq m_1 \leq d$.

Two special cases need to be accounted for, when the above summation scheme breaks down. One is when $m_1 = 1$, where we take the Vandermonde term to be unity and get

$$F_d(m_0, 1, m_2) = \frac{(m_0)_d (m_2)_d}{d!}. \tag{11.2.23}$$

The second is when $m_1 = 0$. To evaluate this case we again make use of the known symmetry of $F_d(m_0, m_1, m_2)$. Evaluating instead at $m_0 = 0$, we get

$$\begin{aligned} F_d(0, m_1, m_2) &= \sum_{\substack{k_0, \ldots, k_{m_1-1} \geq 0 \\ k_0 + \ldots + k_{m_1-1} = d}} \prod_{1 \leq i < j < m_1} \left(1 - \frac{k_j - k_i}{j - i}\right) \prod_{a=0}^{m_1-1} \frac{\delta_{k_a,0} (m_2 - a)_{k_a}}{k_a!} \\ &= \sum_{\substack{k_0, \ldots, k_{m_1-1} \geq 0 \\ k_0 + \ldots + k_{m_1-1} = d}} \prod_{a=0}^{m_1-1} \delta_{k_a,0} = \delta_{d,0}. \end{aligned}$$

Thus we infer that $F_d(m_0, 0, m_2) = \delta_{d,0}$ as well.

11.2.5 Implementation and Results

We can implement this algorithm in any of a number of different software packages. Mathematica is a suitable choice, and Algorithm 11.1 shows one possible implementation. It uses the following steps:

1. Ftmp[d,m1] gives $F_d(m_0, m_1, m_2)$ as a polynomial in m_0 and m_2, for known d and m_1, by evaluating the summation expression (11.2.22) directly. Note the use of **Apply** to construct the variable-dimension summation. The special cases $m_1 = 0$ and $m_1 = 1$ are evaluated separately; the $d = 0$ special case is not calculated as, given that it equals unity, it is trivial to factor it out of the recursion relation.

2. F[d] gives $F_d(m_0, m_1, m_2)$ as as a polynomial in m_0, m_1 and m_2, for known d. It is computed by polynomial interpolation from Ftmp[d,m1], using Mathematica's built-in **InterpolatingPolynomial** function.

3. H[d] gives $H_d(m_0, m_1, m_2)$, using (11.2.21).

11.2.6 Sums over subsets and equivalences

The above results are somewhat lacking, in that they don't allow us to derive exact closed-form expressions for our generating functions. In Chapter 10 we had found closed form summatory expressions for the one-faced-rooted-hypermap generating functions,

Algorithm 11.1 An implementation in Mathematica for computing the generating functions $H_d(m_0, m_1, m_2)$.

```
Ftmp[d_, m1_] := Apply[
  Sum, Prepend[
    Table[
      {k[i], 0, d - Sum[k[j], {j, 0, i - 1}]}, {i, 0, m1 - 2}
    ],
    Product[
      1 - (k[j] - k[i])/(j - i), {j, 1, m1 - 1}, {i, 0, j - 1}
    ] Product[
      Pochhammer[m0, k[a]] Pochhammer[m2 - a, k[a]]/k[a]!,
      {a, 0, m1 - 1}
    ] /. k[m1 - 1] :> (d - Sum[k[c], {c, 0, m1 - 2}])
  ]
];
Ftmp[d_, 0] := 0;
Ftmp[d_, 1] := Pochhammer[m0, d] Pochhammer[m2, d]/d!;
F[d_] := F[d] = InterpolatingPolynomial[
  Table[{mtmp, Ftmp[d, mtmp]}, {mtmp, 0, d}], m1
];
H[d_] := H[d] = d F[d] - Sum[H[t] F[d - t], {t, 1, d - 1}];
```

whereas here we can only evaluate the generating functions via an indirect, algorithmic method.

That does not mean that we can't find any closed-form results for general rooted hypermaps, however. The only reason we can't do so in general is because the summatory expression (11.2.22) for $F_d(m_0, m_1, m_2)$ contains products and sums whose ranges are dependent on the parameter m_1. But as we noted previously, this means that if we set m_1 to a constant, what remains *can* be written in closed form. In particular, we already noted in (11.2.23) that

$$F_d(m_0, 1, m_2) = \frac{(m_0)_d (m_2)_d}{d!}.$$

This particular case has a combinatorial meaning of its own – setting m_1 to unity in $H_d(m_0, m_1, m_2)$ gives us a generating function which enumerates all rooted hypermaps with d darts by number of faces and vertices, with all possible numbers of edges summed over. We now have a simple recursion relation for said generating functions:

$$H_d(m_0, 1, m_2) = \frac{(m_0)_d (m_2)_d}{(d-1)!} - \sum_{k=1}^{d-1} \frac{(m_0)_{d-k} (m_2)_{d-k}}{(d-k)!} H_k(m_0, 1, m_2), \qquad (11.2.24)$$

where we have simply substituted $m_1 = 1$ into (11.2.21). We can also compute $H_d(1, m_1, m_2)$ and $H_d(m_0, m_1, 1)$, using the fact that $H_d(m_0, m_1, m_2)$ is completely symmetric.

We can perform further summations by setting additional parameters to unity. In particular, we have the following result:

Theorem 11.2.4. *Let H_d be the total number of rooted hypermaps with d darts. Then*

$$H_d = d \cdot d! - \sum_{k=1}^{d-1} (d-k)! H_k \tag{11.2.25}$$

for all $d > 1$, with $H_1 = 1$.

Proof. There is only one way to construct a rooted hypermap with one dart, so $H_1 = 1$. In general, we have that $H_d = H_d(1, 1, 1)$. Substituting this into (11.2.24) gives the above result. □

The first few such values are 1, 3, 13, 71, 461, 3447 etc.

This is not the only context in which this sequence arises [31][4]. Much like how we inferred the fact that $P_d(m_1, m_2)$ was a generating function by examining its coefficients, this appearance of the same counts in other contexts allows us to infer a number of properties and equivalences for rooted hypermaps. For example, we can derive a generating function for the sequence H_d:

Theorem 11.2.5. *The alternating generating function for the sequence H_d, for $d \geq 1$, is*

$$H(x) = -\left[\frac{1+x}{x} + \frac{1}{\sum_{n=0}^{\infty} n!(-x)^{n+1}}\right] = \frac{e^{-1/x}}{\Gamma(0, 1/x)} - \frac{1+x}{x},$$

where $\Gamma(n, z) = \int_z^{\infty} e^{-w} w^{n-1} dw$ is the incomplete gamma function.

Proof. $H_d = a(d+1)$ for all $d \geq 1$, where $a(n)$ is defined in [31, 9]. We have from the same source that, for $n \geq 1$, the generating function for the sequence $a(n)$ is

$$\begin{aligned} f(x) &= \sum_{n=1}^{\infty} (-x)^n a(n) \\ &= 1 - \frac{1}{\sum_{n=0}^{\infty} n!(-x)^n} \end{aligned}$$

(note that we have chosen to define this as an alternating generating function to match $H(m_0, m_1, m_2; x)$; the only change from [31] is the sign of x). Thus, accounting for the shift in index and the lack of term corresponding to $a(1)$, we have that the sequence H_d has the generating function

$$\begin{aligned} H(x) &= \sum_{d=1}^{\infty} (-x)^d a(d+1) \\ &= \frac{f(x) + x}{-x} \\ &= -\left[\frac{1+x}{x} + \frac{1}{\sum_{n=0}^{\infty} n!(-x)^{n+1}}\right]. \end{aligned}$$

[4]See the FORMULA section for confirmation that this sequence is given by the recursion (11.2.25).

The summation is divergent for non-zero x, but we can find a non-summatory form for it using Borel summation. Let

$$\alpha(x) = \sum_{n=0}^{\infty} n!(-x)^n.$$

We define the Borel transform of this as

$$\mathcal{B}\alpha(w) = \sum_{n=0}^{\infty} \frac{n!(-w)^n}{n!} = \frac{1}{1+w}$$

when $|w| \leq 1$. We then recover $\alpha(x)$ by analytically extending $\mathcal{B}\alpha(w)$ to the entire positive real line, and using the definition of the Borel sum

$$\begin{aligned}
\alpha(x) &= \int_0^{\infty} e^{-w}\mathcal{B}\alpha(wx)dw \\
&= \int_0^{\infty} \frac{e^{-w}}{1+wx}dw \\
&= \int_{1/x}^{\infty} \frac{e^{-w+1/x}}{wx}dw \\
&= \frac{e^{1/x}}{x}\Gamma\left(0, \frac{1}{x}\right).
\end{aligned}$$

Thus,

$$\begin{aligned}
H(x) &= -\left[\frac{1+x}{x} - \frac{1}{x\alpha(x)}\right] \\
&= \frac{e^{-1/x}}{\Gamma(0,1/x)} - \frac{1+x}{x}.
\end{aligned}$$

□

We can also infer a number of bijections from rooted hypermaps to other types of object. For instance, [31] gives us that the set of rooted d-hypermaps is, for any $d \geq 1$, bijective with both the set of all connected $(d+1)$-permutations [19], and the set of all $(d+1)$-permutations with no global descents [2].

In neither of these cases is there an immediately apparent reason for the connection (the structures of these objects are not trivially equivalent to rooted hypermaps in any of the representations we have used so far), but we know from their shared recursion relations that the sets have the same cardinalities, which is all that's required for a bijection to exist.

We have one other bijection, though, where the reason *is* apparent, which again is noted in [31]:

Theorem 11.2.6. *There is a bijection between the set of rooted d-hypermaps and the set of subgroups of index d in the free group of rank two.*

Proof. That the number of index-d subgroups of the rank-two free group satisfies the recursion 11.2.25 has been known since 1949 [17]. The proof of this relies on the fact that each such subgroup can be represented by a pair of d-permutations g_1 and g_2 which generate a group transitive on $[1 \dots d]$, and that any two such representations are isomorphic if they can be obtained by replacing the indices $(1, 2, \dots, d)$ with $(1, c_2, \dots, c_d)$ within g_1 and g_2 for some ordering $(c_2, \dots, c_d)$ of $(2, \dots, d)$.

Such a pair of permutations, given that they generate a transitive group, give us a 3-constellation $[g_1, g_2, (g_1 g_2)^{-1}]$, and the isomorphism given above is exactly the same isomorphism we gave for defining rooted hypermaps. Therefore, there is a direct equivalence between subgroups of the rank-two free group and rooted hypermaps, which equates subgroup index with hypermap degree. □

So this connection follows from the fact that the two types of object essentially share a representation as rooted 3-constellations. Given this close connection, it's not surprising that we can use Hall's method to re-derive our summation results from this section (specifically (11.2.24) – see Appendix G). Hall's method cannot reproduce $H_d(m_0, m_1, m_2)$ or its functional identities in full however. This is because his method only deals with the structure of the permutations g_1 and g_2, and cannot give any information about the cycles in the third permutation $(g_1 g_2)^{-1}$.

An important feature of Hall's proof is that it is applicable to higher-rank free groups as well as those of rank two, and, as we shall see in Theorem 12.2.2, it shows that there there is a correspondence between subgroups of the free group of rank k and rooted $(k+1)$-constellations.

Of course the same limitation still exists – this method cannot produce a full generating function which partitions constellations by the cycle structure of all of its permutations. However, we can derive properties of these full generating functions by generalising the method we used for rooted hypermaps to rooted constellations of higher length. We will look at this in the next chapter, along with doing the same for enumerating rooted maps.

Chapter 12

Further problems

The work of the previous chapters, concerning the enumeration of rooted hypermaps, covers all the fundamental principles of our matrix integration tools and how to use them to solve enumeration problems. In Chapter 10 we introduced the matrix integrals themselves, using a relatively simple case (one-face rooted hypermaps) to study their manipulation and evaluation. Then in Chapter 11 we moved on to general rooted hypermaps, and showed how to generalise the basic methods, in particular how to avoid overcounting due to invalid and isomorphic cases.

This therefore covers the most general hypermap case that our method can study, i.e. all hypermaps up to rooted isomorphism. We cannot go any further as the rooted isomorphism is tied directly to the construction of the matrix integrals we used, so unrooted hypermaps cannot be enumerated by this method.

This does not mean that we have reached the limit of what these methods can do, however. There are two directions in which we can still move while maintaining the rootedness condition: *specialisation* (enumerating sets of objects which are equivalent to subsets of the rooted hypermaps) and *generalisation* (enumerating supersets of the rooted hypermaps). In this chapter we will look at one example of each.

As an example of specialisation we will use the case of rooted maps. As we discussed in Section 9.1.4, hypermaps can be thought of as bipartite maps, a specialisation of maps, but conversely maps can be thought of as a specialisation of hypermaps where each of the edges contains exactly two darts. We will make use of this fact in Section 12.1.

For generalisation, we will start from the fact that hypermaps are equivalent to 3-constellations, and look at how to generalise our methods to rooted constellations of length greater than three in Section 12.2.

Together, these two problems will demonstrate some of the ways in which our basic method can be extended and modified for use in other contexts. As we will see, the methods we will use in these two examples are very similar to the hypermap case from the last chapter, the modifications we need to make being much less significant than those we made when moving from one-face hypermaps.

12.1 Rooted maps

A rooted map is simply a rooted hypermap where each edge contains exactly two darts. This fits with the familiar notion of a map as a set of vertices connected pairwise by "edge" lines – these edges correspond to the hypermap's edges, with the darts being the edges' two ends.

As we can describe maps as hypermaps, we can also describe them using ladder diagrams. In order to do so we need to make one change to the conventions we've used so far though. When we represented hypermaps as ladder diagrams in Chapters 10 and 11, we associated the permutation σ (which provides the ladder diagram's backbone) with the hypermaps' faces. There is no specific reason why it should be this way, though, as we have also established that the vertices, edges and faces of a hypermap are essentially interchangeable (see Lemma 9.1.1). If we had associated σ with the hypermaps' edges instead the resulting generating functions, apart from having their parameters reordered, would have been equivalent.

Now that we are looking at maps, however, we do have reasons to make a specific choice. We are constructing maps by taking hypermaps and applying a constraint to their edges, which means that we now need to associate σ with the maps' edges, as it is only σ which we have a means of constraining.

Aside from this, the method for constructing generating functions for rooted maps is very similar to the method we used for rooted hypermaps, with many of the steps being identical. Because of this we will only sketch the proof of the result here, highlighting the points where the two methods differ. As before, the process follows the following steps:

1. Construct the generating function $M(m_1, m_2; x)$ using the transitive ladder-diagram generating functions $\bar{P}_{d_1,\ldots,d_N}(m_1, m_2)$ from Lemma 11.2.2, taking into account any multiple counting.

2. Define a new function $G(m_1, m_2; x)$ in terms of the non-transitive counting functions $P_{d_1,\ldots,d_N}(m_1, m_2)$ (defined in (11.2.1)), and use Lemma 11.2.2 to find a functional relation linking this to $M(m_1, m_2; x)$.

3. Find a matrix-integral expression for $G(m_1, m_2; x)$, and use this to find the terms in its power series expansion in x.

Theorem 12.1.1. *Let $M(m_1, m_2; x)$ be the generating function (alternating in x) enumerating rooted maps by number of vertices, faces and edges. This satisfies*

$$M(m_1, m_2; x)G(m_1, m_2; x) = 2x\frac{\partial}{\partial x}G(m_1, m_2; x),$$

where

$$G(m_1, m_2; x) = \sum_{k_0=0}^{\infty} \cdots \sum_{k_{m_1-1}=0}^{\infty} \prod_{0 \le i < j < m_1} \left(1 - \frac{k_j - k_i}{j - i}\right) \prod_{a=0}^{m_1-1} \frac{(m_2 - a)_{2k_a}}{k_a! 2^{k_a}} (-x)^{k_a}.$$

Proof. A rooted map with k edges is represented by a ladder diagram with backbone $\sigma = (12)(34)\cdots((2k-1)(2k))$. These ladder diagrams are enumerated by the generating function

$$\bar{P}_{\underbrace{2,\ldots,2}_{k}}(m_1, m_2).$$

Isomorphisms under reordering/redirecting of the non-rooted edges mean that each rooted map is counted $(k-1)!2^{k-1}$ times, so when we sum over all possible k, taking this degeneracy into account, we get

$$M(m_1, m_2; x) = \sum_{k=1}^{\infty} \frac{(-x)^k}{(k-1)!2^{k-1}} \bar{P}_{\underbrace{2,\ldots,2}_{k}}(m_1, m_2).$$

We define $M(m_1, m_2; x)$ to be alternating in x, where x now partitions the maps by number of edges.

Let

$$G(m_1, m_2; x) = \sum_{k=0}^{\infty} \frac{(-x)^k}{k!2^k} P_{\underbrace{2,\ldots,2}_{k}}(m_1, m_2), \tag{12.1.1}$$

where $P_{d_1,\ldots,d_N}(m_1, m_2)$ is defined in (11.2.1). If we multiply $M(m_1, m_2; x)$ and $G(m_1, m_2; x)$ together, we get

$$\begin{aligned}
M(m_1, m_2; x)G(m_1, m_2; x) &= \sum_{d=0}^{\infty}\sum_{k=1}^{\infty} \frac{(-x)^{k+d}}{(k-1)!d!2^{k+d-1}} \\
&\quad \times \bar{P}_{\underbrace{2,\ldots,2}_{k}}(m_1, m_2) P_{\underbrace{2,\ldots,2}_{d}}(m_1, m_2) \\
&= \sum_{d=0}^{\infty}\sum_{k=d+1}^{\infty} \frac{(-x)^{k}}{(k-d-1)!d!2^{k-1}} \\
&\quad \times \bar{P}_{\underbrace{2,\ldots,2}_{k-d}}(m_1, m_2) P_{\underbrace{2,\ldots,2}_{d}}(m_1, m_2) \\
&= \sum_{k=1}^{\infty} \frac{(-x)^{k}}{(k-1)!2^{k-1}} \\
&\quad \times \sum_{d=0}^{k-1} \binom{k-1}{d} \bar{P}_{\underbrace{2,\ldots,2}_{k-d}}(m_1, m_2) P_{\underbrace{2,\ldots,2}_{d}}(m_1, m_2).
\end{aligned}$$

Substituting $N = k$ and $(d_1, \ldots, d_N) = (2, 2, \ldots, 2)$ into (11.2.3), we get

$$P_{\underbrace{2,\ldots,2}_{k}}(m_1, m_2) = \sum_{d=0}^{k-1} \binom{k-1}{d} \bar{P}_{\underbrace{2,\ldots,2}_{k-d}}(m_1, m_2) P_{\underbrace{2,\ldots,2}_{d}}(m_1, m_2),$$

so

$$M(m_1,m_2;x)G(m_1,m_2;x) = \sum_{k=1}^{\infty} \frac{(-x)^k}{(k-1)!2^{k-1}} P_{\underbrace{2,\ldots,2}_{k}}(m_1,m_2)$$
$$= 2x\frac{\partial}{\partial x}G(m_1,m_2;x).$$

Note the factor of two, which did not appear in the equivalent identity (11.2.9) from the rooted hypermap case. This is because x now partitions by edge count instead of dart count, and for maps these two values differ by a factor of two.

This gives our functional identity, so next we need to evaluate $G(m_1,m_2;x)$. Starting with (11.2.10) we have that

$$P_{\underbrace{2,\ldots,2}_{k}}(m_1,m_2) = \frac{1}{\Lambda}\int \prod_{k=1}^{m_1} q_k^{m_2-m_1}e^{-q_k}dq_k \prod_{1\le i<j\le m_1}(q_j-q_i)^2 \left(\sum_{a=1}^{m_1} q_a^2\right)^k$$

for some normalisation factor Λ which depends only on m_1 and m_2. We substitute this into (12.1.1) and sum over k, giving

$$G(m_1,m_2;x) = \frac{1}{\Lambda}\int \prod_{k=1}^{m_1} q_k^{m_2-m_1}e^{-q_k}dq_k \prod_{1\le i<j\le m_1}(q_j-q_i)^2 \exp\left(-\frac{x}{2}\sum_{a=1}^{m_1} q_a^2\right)$$
$$= \frac{1}{\Lambda}\int \prod_{k=1}^{m_1} q_k^{m_2-m_1}e^{-q_k-q_k^2x/2}dq_k \prod_{1\le i<j\le m_1}(q_j-q_i)^2.$$

We now need to use the same trick – breaking the Vandermonde product term into orthogonal polynomials – that we used in Theorem 11.2.3 to make this integral separable. This time our integrals resemble an inner product with weight function

$$w(q) = q^{m_2-m_1}e^{-q-q^2x/2},$$

which has moments

$$\begin{aligned}\mu_n &= \int_0^{\infty} w(q)q^n dq \\ &= \int_0^{\infty} q^{m_2-m_1+n}e^{-q}e^{-q^2x/2}dq \\ &= \sum_{k=0}^{\infty}\frac{(-x)^k}{k!2^k}\int_0^{\infty} q^{m_2-m_1+n+2k}e^{-q}dq \\ &= \sum_{k=0}^{\infty}\frac{(-x)^k}{k!2^k}(m_2-m_1+n+2k)!. \end{aligned} \tag{12.1.2}$$

If we define monic polynomials $K_n(q)$ orthogonal with respect to this weight function

then, paralleling (11.2.14) we have

$$\begin{aligned} G(m_1, m_2; x) &= \frac{m_1!}{\Lambda} \prod_{k=1}^{m_1} \int_0^\infty K_{k-1}(q) K_{k-1}(q) w(q) dq \\ &= \frac{m_1!}{\Lambda} \prod_{k=1}^{m_1} I_{k-1}. \end{aligned}$$

So, via Corollary F.0.1, we have that

$$G(m_1, m_2; x) = \frac{m_1!}{\Lambda} |\mathbf{M}_{m_1}|$$

where

$$\mathbf{M}_{m_1} = \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{m_1-1} \\ \mu_1 & \mu_2 & \cdots & \mu_{m_1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{m_1-1} & \mu_{m_1} & \cdots & \mu_{2m_1-2} \end{bmatrix}.$$

Therefore, by analogy with 11.2.15, when we substitute (12.1.2) into this we get

$$G(m_1, m_2; x) = \frac{m_1!}{\Lambda} \sum_{k_1=0}^{\infty} \frac{(-x)^{k_1}}{k_1! 2^{k_1}} \cdots \sum_{k_{m_1}=0}^{\infty} \frac{(-x)^{k_{m_1}}}{k_{m_1}! 2^{k_{m_1}}} \Phi(2k_1, \ldots, 2k_{m_1})$$

where $\Phi(k_1, \ldots, k_{m_1})$ is the same function as defined in (11.2.16). We substitute (11.2.17) in here, getting

$$\begin{aligned} G(m_1, m_2; x) = \frac{m_1!}{\Lambda} \sum_{k_1=0}^{\infty} \frac{(m_2 + 2k_1 - 1)!}{k_1! 2^{k_1}} (-x)^{k_1} \cdots \sum_{k_{m_1}=0}^{\infty} \frac{(m_2 + 2k_{m_1} - m_1)!}{k_{m_1}! 2^{k_{m_1}}} (-x)^{k_{m_1}} \\ \times \prod_{1 \le i < j \le m_1} (j - i + 2k_i - 2k_j). \end{aligned}$$

We use the fact that $G(m_1, m_2; 0) = 1$ (see (12.1.1)) to fix the normalisation. This gives us that

$$1 = \frac{m_1!}{\Lambda} \prod_{a=1}^{m_1} (m_2 - a)! \prod_{1 \le i < j \le m_1} (j - i),$$

and therefore

$$\begin{aligned} G(m_1, m_2; x) &= \sum_{k_1=0}^{\infty} \cdots \sum_{k_{m_1}=0}^{\infty} \prod_{1 \le i < j \le m_1} \left(1 - 2\frac{k_j - k_i}{j - i}\right) \prod_{a=1}^{m_1} \frac{(m_2 + 2k_a - a)!}{(m_2 - a)! 2^{k_a}} \frac{(-x)^{k_a}}{k_a!} \\ &= \sum_{k_1=0}^{\infty} \cdots \sum_{k_{m_1}=0}^{\infty} \prod_{1 \le i < j \le m_1} \left(1 - 2\frac{k_j - k_i}{j - i}\right) \prod_{a=1}^{m_1} \frac{(m_2 - a + 1)_{2k_a}}{k_a! 2^{k_a}} (-x)^{k_a} \\ &= \sum_{k_0=0}^{\infty} \cdots \sum_{k_{m_1-1}=0}^{\infty} \prod_{0 \le i < j < m_1} \left(1 - 2\frac{k_j - k_i}{j - i}\right) \prod_{a=0}^{m_1-1} \frac{(m_2 - a)_{2k_a}}{k_a! 2^{k_a}} (-x)^{k_a}. \end{aligned}$$

□

12.1.1 Evaluation

As in Section 11.2.4, we can use this to evaluate $M(m_1, m_2; x)$ by taking its power series expansion

$$M(m_1, m_2; x) = \sum_{e=1}^{\infty} M_e(m_1, m_2)(-x)^e,$$

with the $M_e(m_1, m_2)$ being polynomial generating functions which enumerate rooted maps with e edges by number of vertices and faces. $G(m_1, m_2; x)$ has a corresponding expansion

$$G(m_1, m_2; x) = \sum_{e=0}^{\infty} G_e(m_1, m_2)(-x)^e$$

with coefficients

$$G_e(m_1, m_2) = \sum_{\substack{k_0, \ldots, k_{m_1-1} \geq 0 \\ k_0 + \ldots + k_{m_1-1} = e}} \prod_{0 \leq i < j < m_1} \left(1 - 2\frac{k_j - k_i}{j - i}\right) \prod_{a=0}^{m_1-1} \frac{(m_2 - a)_{2k_a}}{k_a! 2^{k_a}}$$

for all $e \geq 0$, and these are related to $M_e(m_1, m_2)$ by the identity

$$\sum_{k=1}^{e} M_k(m_1, m_2) G_{e-k}(m_1, m_2) = 2e G_e(m_1, m_2).$$

As was the case for rooted hypermaps, we can compute all of these functions by evaluating at a number of specific values of m_1 and interpolating. $G_e(m_1, m_2)$ is of order at most $2e$ in m_2, and by symmetry will have the same order in m_1, so we need to evaluate it at $2e + 1$ different values of m_1 before interpolating.

The first few generating functions produced by this method are

$$\begin{aligned}
M_1(m_1, m_2) &= m_1^2 m_2 + m_1 m_2^2 \\
M_2(m_1, m_2) &= m_1 m_2 + 2m_1^3 m_2 + 5m_1^2 m_2^2 + 2m_1 m_2^3 \\
M_3(m_1, m_2) &= 10m_1^2 m_2 + 10m_1 m_2^2 + 5m_1^4 m_2 + 22m_1^3 m_2^2 + 22m_1^2 m_2^3 + 5m_1 m_2^4,
\end{aligned}$$

which we see reproduce the results of Walsh and Lehman [37].

As in the hypermap case we can refine these results further. If we set $m_1 = 1$ we get generating functions which enumerate rooted maps by number of vertices, summing over all face counts. After this substitution the summations and products in $G_e(m_1, m_2)$ all become trivial, leaving behind

$$G_e(1, m_2) = \frac{(m_2)_{2e}}{e! 2^e}.$$

Therefore,

$$\sum_{k=1}^{e} \frac{(m_2)_{2(e-k)}}{(e-k)! 2^{e-k}} M_k(1, m_2) = \frac{(m_2)_{2e}}{(e-1)! 2^{e-1}}.$$

Setting $m_2 = 1$ as well gives a recursion relation for the total number of rooted maps for a given edge count:

$$\sum_{k=1}^{e} \frac{[2(e-k)]!}{(e-k)!2^{e-k}} M_k = \frac{(2e)!}{(e-1)!2^{e-1}}, \tag{12.1.3}$$

or, simplifying further,

$$\sum_{k=1}^{e} [2(e-k)-1]!! M_k = 2e(2e-1)!!$$

where $(2e-1)!! = 1 \times 3 \times 5 \times \cdots \times (2e-1)$ is the double factorial. The sequence this recurrence generates is the same as [32], starting at $a(2)$.

This result in particular is interesting, as the intermediate terms

$$G_e(1,1) = (2e-1)!!$$

used in the recursion are already known to be the number of rooted maps with e edges and one vertex [37, Equation 4], or equally those with e edges and one face. This appearance of counts for one-face rooted maps is interesting as it is a close parallel to our study of one-face rooted hypermaps in Chapter 10; the equivalent recursion for counting all rooted hypermaps by dart count had intermediate expressions $F_d(1,1,1) = d!$, which is likewise the number of rooted hypermaps with d darts and one face. It is not clear how general a pattern this is, however, or in what other contexts it may apply, but we will see in Section 12.2.2 that it applies to general rooted constellations as well, not just hypermaps.

12.2 General constellations

The final example we will look at is that of general constellations. As noted in Section 9.1, hypermaps are equivalent to 3-constellations – constellations containing three permutations – but constellations can consist of any number of permutations greater than this.

As a reminder, an k-constellation is a sequence of d-permutations $[g_1, g_2, \ldots, g_k]$ for some integer d, such that the product $g_1 g_2 \cdots g_k$ is the identity and the group generated by $[g_1, g_2, \ldots, g_k]$ acts transitively on $[1 \ldots d]$ (Definition 9.1.4). When $k \leq 2$ these rules give rise to sets too trivial to be of interest to us, but when $k > 2$ the problem of enumeration becomes significantly more complicated. We have already done the $k = 3$ case with rooted hypermaps, and it should be no surprise that the methods we used, which were strongly centred around the permutation structure of the constellations, should also be applicable to rooted constellations of greater length.

It must be noted, however, that the usefulness of these methods diminishes significantly in this more general case. We saw when we generalised the one-faced rooted hypermap case to all rooted hypermaps that there was a significant decrease in

usefulness of the resulting expressions: we went from having closed-form expressions for individual generating functions to having a recursive procedure for evaluating generating functions algorithmically. As we move to an even more general case now we will find that the returns diminish even further; the best we will be able to get now is a matrix-integral expression for the intermediate functions in the recursive procedure (analogous to $F(m_0, m_1, m_2; x)$ from the hypermap case and $G(m_1, m_2; x)$ from the map case). We can't evaluate these integrals using the methods we've already looked at, because they involve integrals over multiple independent matrix spaces instead of just one.

However we will be able to extract some simple results from these integrals in certain special cases e.g. when one or more of the parameters of the function are set to unity, as we did in the hypermap case (Section 11.2.6).We will also find a connection between our results and subsets of free groups, as we did in Theorem 11.2.6.

12.2.1 Deriving the matrix integrals

Much like in the case with rooted maps, the solution for this problem very closely mirrors that of the hypermap case. The main modification which we need to make is in Lemma 11.2.1, where we established the ladder-diagram counting functions $P_{d_1,\ldots,d_N}(m_1, m_2)$. Specifically, let us consider (11.2.2) again:

$$\begin{aligned} P_{d_1,\ldots,d_N}(m_1, m_2) &= \sum_{g \in Sym_d} \delta[a_1, a_{\sigma g(1)}]\delta[b_1, b_{g(1)}] \cdots \delta[a_d, a_{\sigma g(d)}]\delta[b_d, b_{g(d)}] \\ &= \frac{\partial}{\partial \alpha_{a_1 b_1}} \cdots \frac{\partial}{\partial \alpha_{a_d b_d}} (\alpha_{a_{\sigma(1)} b_1} \cdots \alpha_{a_{\sigma(d)} b_d}) \Big|_{\alpha=1} . \end{aligned}$$

As a reminder, this connects to 3-constellations as follows: For a permutation σ with cycle lengths $d_1, \ldots, d_N$ (defined using the convention stated in Lemma 11.2.1) and arbitrary permutation g, define the permutation triple $[\sigma, g, (\sigma g)^{-1}]$. This is like a 3-constellation except that it is not necessarily transitive. For each g the Kronecker delta terms in (11.2.2) contract down to an expression of the form $m_1^{\mathrm{cyc}(\sigma g)} m_2^{\mathrm{cyc}(g)}$ (where m_1 and m_2 are the dimensions of the a- and b-indices respectively), so the sum over g then gives a generating function enumerating all triples with the same base permutation σ.

The generalisation to k-permutations is then reasonably simple: we write each k-constellation as $[\sigma, g_1, g_2, \ldots, g_{k-2}, (\sigma g_1 \cdots g_{k-2})^{-1}]$ (with σ defined the same way) and sum over all $g_1, \ldots, g_{k-2}$. This produces a new generating function analogous to $P_{d_1,\ldots,d_N}(m_1, m_2)$, again summing over all (potentially non-transitive) permutation sequences with a given σ, so the rest of the derivation runs exactly the same as before.

Our Kronecker-delta expression has to be somewhat more complicated to deal with

the new permutations, but we can construct a suitable expression as follows:

$$P^{(k)}_{d_1,\dots d_N}(m_1,\dots,m_{k-1}) = \sum_{g_1,\dots,g_{k-2}} \prod_{i=1}^{d} \delta[a_i, a_{\sigma g_1 \cdots g_{k-2}(i)}]\delta[b^{(1)}_i, b^{(1)}_{g_1(i)}]\cdots\delta[b^{(k-2)}_i, b^{(k-2)}_{g_{k-2}(i)}], \tag{12.2.1}$$

where the $b^{(j)}$-indices run from 1 to m_j respectively and a-indices run from 1 to m_{k-1}. We can expand this further, by noting that

$$\delta[a^{(1)}_i, a^{(1)}_{\sigma g_1\cdots g_{k-2}(i)}] = \delta[a^{(1)}_i, a^{(k-2)}_{g_{k-2}(i)}]\delta[a^{(k-2)}_{g_{k-2}(i)}, a^{(k-3)}_{g_{k-3}g_{k-2}(i)}]\cdots\delta[a^{(2)}_{g_2\cdots g_{k-2}(i)}, a^{(1)}_{\sigma g_1\cdots g_{k-2}(i)}],$$

where all of the $a^{(j)}$-indices run from 1 to m_{k-1}. Furthermore, when we put this into (12.2.1) the product over i allows us to freely substitute indices in each individual delta without changing the result. Replacing i with $g^{-1}_{k-2}(i)$ in the second, i with $g^{-1}_{k-2}g^{-1}_{k-3}(i)$ in the third etc. gives us

$$\begin{aligned} P^{(k)}_{d_1,\dots d_N}(m_1,\dots,m_{k-1}) = \sum_{g_1,\dots,g_{k-2}} \prod_{i=1}^{d} &\delta[a^{(1)}_i, a^{(k-2)}_{g_{k-2}(i)}]\delta[b^{(k-2)}_i, b^{(k-2)}_{g_{k-2}(i)}] \\ &\times \delta[a^{(k-2)}_i, a^{(k-3)}_{g_{k-3}(i)}]\delta[b^{(k-3)}_i, b^{(k-3)}_{g_{k-3}(i)}] \\ &\vdots \\ &\times \delta[a^{(2)}_i, a^{(1)}_{\sigma g_1(i)}]\delta[b^{(1)}_i, b^{(1)}_{g_1(i)}]. \end{aligned}$$

Now we can convert this into a multiderivative, as each Kronecker delta only includes one of the summed permutations. As before these derivatives will be in terms of matrix-valued dummy variables, but we now need to have more than one, one for each permutation. We will call these $\alpha^{(j)}$ for $1 \le j \le k-2$, where each one has dimensions $m_{k-1} \times m_j$. We get that

$$\begin{aligned} P^{(k)}_{d_1,\dots d_N}(m_1,\dots,m_{k-1}) = &\frac{\partial}{\partial \alpha^{(1)}_{a^{(2)}_1 b^{(1)}_1}} \cdots \frac{\partial}{\partial \alpha^{(1)}_{a^{(2)}_d b^{(1)}_d}} (\alpha^{(1)}_{a^{(1)}_{\sigma(1)} b^{(1)}_1} \cdots \alpha^{(1)}_{a^{(1)}_{\sigma(d)} b^{(1)}_d}) \\ &\times \frac{\partial}{\partial \alpha^{(2)}_{a^{(3)}_1 b^{(2)}_1}} \cdots \frac{\partial}{\partial \alpha^{(2)}_{a^{(3)}_d b^{(2)}_d}} (\alpha^{(2)}_{a^{(2)}_1 b^{(2)}_1} \cdots \alpha^{(2)}_{a^{(2)}_d b^{(2)}_d}) \\ &\vdots \\ &\times \frac{\partial}{\partial \alpha^{(k-2)}_{a^{(1)}_1 b^{(k-2)}_1}} \cdots \frac{\partial}{\partial \alpha^{(k-2)}_{a^{(1)}_d b^{(k-2)}_d}} (\alpha^{(k-2)}_{a^{(k-2)}_1 b^{(k-2)}_1} \cdots \alpha^{(k-2)}_{a^{(k-2)}_d b^{(k-2)}_d}) \Bigg|_{\alpha=0}, \end{aligned}$$

or equivalently

$$P^{(k)}_{d_1,\dots d_N}(m_1,\dots,m_{k-1}) = \frac{\partial}{\partial\alpha^{(1)}_{a_1^{(2)}b_1^{(1)}}}\frac{\partial}{\partial\beta^{(1)}_{a^{(1)}_{\sigma(1)}b_1^{(1)}}}\cdots\frac{\partial}{\partial\alpha^{(1)}_{a_d^{(2)}b_d^{(1)}}}\frac{\partial}{\partial\beta^{(1)}_{a^{(1)}_{\sigma(d)}b_d^{(1)}}}e^{\alpha^{(1)}\cdot\beta^{(1)}}$$
$$\times\frac{\partial}{\partial\alpha^{(2)}_{a_1^{(3)}b_1^{(2)}}}\frac{\partial}{\partial\beta^{(2)}_{a_1^{(2)}b_1^{(2)}}}\cdots\frac{\partial}{\partial\alpha^{(2)}_{a_d^{(3)}b_d^{(2)}}}\frac{\partial}{\partial\beta^{(2)}_{a_d^{(2)}b_d^{(2)}}}e^{\alpha^{(2)}\cdot\beta^{(2)}}$$
$$\vdots$$
$$\times\frac{\partial}{\partial\alpha^{(k-2)}_{a_1^{(1)}b_1^{(k-2)}}}\frac{\partial}{\partial\beta^{(k-2)}_{a_1^{(k-2)}b_1^{(k-2)}}}\cdots\frac{\partial}{\partial\alpha^{(k-2)}_{a_d^{(1)}b_d^{(k-2)}}}\frac{\partial}{\partial\beta^{(k-2)}_{a_d^{(k-2)}b_d^{(k-2)}}}$$
$$e^{\alpha^{(k-2)}\cdot\beta^{(k-2)}}\Big|_{\alpha,\beta=0},$$

where the $\beta^{(j)}$ again have dimensions $m_{k-1}\times m_j$ respectively. Note that all of the α and β get taken to zero.

We can now reverse-engineer a matrix integral expression. First, we replace each exponential by a Gaussian integral:

$$\begin{aligned}e^{\alpha^{(j)}\cdot\beta^{(j)}} &= \frac{1}{\pi^{m_{k-1}m_j}}\int_{\mathbb{C}^{m_{k-1}m_j}} d^{m_{k-1}m_j}z^{(j)}d^{m_{k-1}m_j}\bar{z}^{(j)}\exp(-z^{(j)}\cdot\bar{z}^{(j)}\\ &\qquad + \alpha^{(j)}\cdot z^{(j)} + \beta^{(j)}\cdot\bar{z}^{(j)})\\ &\equiv \int\mathcal{D}z^{(j)}\exp(-z^{(j)}\cdot\bar{z}^{(j)} + \alpha^{(j)}\cdot z^{(j)} + \beta^{(j)}\cdot\bar{z}^{(j)})\end{aligned}$$

(the $\int\mathcal{D}z^{(1)}$ notation, which incorporates the scale factor, is purely for convenience as we will be dealing with a lot of these integrals). Then we perform the derivatives, which results in the substitution

$$\begin{aligned}\frac{\partial}{\partial\alpha^{(j)}_{a,b}} &\to z^{(j)}_{a,b}\\ \frac{\partial}{\partial\beta^{(j)}_{a,b}} &\to \bar{z}^{(j)}_{a,b}.\end{aligned}$$

When we take all the α and β to zero, we get the result

$$P^{(k)}_{d_1,\dots d_N}(m_1,\dots,m_{k-1}) = \int\prod_{j=1}^{k-2}\mathcal{D}z^{(j)}\exp(-z^{(j)}\cdot\bar{z}^{(j)})\prod_{i=1}^{d}z^{(1)}_{a_i^{(2)}b_i^{(1)}}\bar{z}^{(1)}_{a^{(1)}_{\sigma(i)}b_i^{(1)}}z^{(2)}_{a_i^{(3)}b_i^{(2)}}\bar{z}^{(2)}_{a_i^{(2)}b_i^{(2)}}$$
$$\cdots z^{(k-2)}_{a_i^{(1)}b_i^{(k-2)}}\bar{z}^{(k-2)}_{a_i^{(k-2)}b_i^{(k-2)}}. \qquad (12.2.2)$$

Finally, by defining matrices $\rho^{(j)}$ such that $\rho^{(j)}_{a_1a_2} = z^{(j)}_{a_1b}z^{(j)}_{a_2b}$, we simplify the above to

$$\begin{aligned} P^{(k)}_{d_1,\dots d_N}(m_1,\dots,m_{k-1}) &= \int \prod_{j=1}^{k-2} \mathcal{D}z^{(j)} e^{-\mathrm{Tr}(\rho^{(j)})} \prod_{i=1}^{d} \rho^{(k-2)}_{a_i^{(1)}a_i^{(k-2)}} \rho^{(k-3)}_{a_i^{(k-2)}a_i^{(k-3)}} \cdots \rho^{(1)}_{a_i^{(2)}a^{(1)}_{\sigma(i)}} \\ &= \int \prod_{j=1}^{k-2} \mathcal{D}z^{(j)} e^{-\mathrm{Tr}(\rho^{(j)})} \prod_{i=1}^{d} [\rho^{(k-2)}\cdots\rho^{(1)}]_{a_i^{(1)}a^{(1)}_{\sigma(i)}} \\ &= \int \prod_{j=1}^{k-2} \mathcal{D}z^{(j)} e^{-\mathrm{Tr}(\rho^{(j)})} \prod_{r=1}^{N} \mathrm{Tr}[(\rho^{(k-2)}\cdots\rho^{(1)})^{d_r}]. \end{aligned} \tag{12.2.3}$$

This expression is very similar in form to $P_{d_1,\dots,d_N}(m_1,m_2)$ in (11.2.1) – which is of course just the $k=3$ case. The only differences are the inclusion of multiple integrals and the replacement of $\hat{\rho}$ with $\rho^{(k-2)}\cdots\rho^{(1)}$. As a result, the proof for the following theorem is almost exactly the same as that for Theorem 11.2.1. In particular the process for eliminating non-transitive cases and multiple counting remains unchanged. Given the similarities we will give only a sketch of the proof here, as many of the steps would be almost word-for-word reproductions of our earlier work. We will make clear where this happens, however.

Theorem 12.2.1. *Let $H^{(k)}(m_0,m_1,\dots,m_{k-1};x)$ be the generating function (alternating in x) enumerating all rooted k-constellations, partitioned by the cycle structures of the permutations and the permutations' degree. Then*

$$H^{(k)}(m_0,m_1,\dots,m_{k-1};x)F^{(k)}(m_0,m_1,\dots,m_{k-1};x) = x\frac{\partial}{\partial x}F^{(k)}(m_0,m_1,\dots,m_{k-1};x),$$

where

$$F^{(k)}(m_0,m_1,\dots,m_{k-1};x) = \int \prod_{j=1}^{k-2} \mathcal{D}z^{(j)} e^{-Tr(\rho^{(j)})} \det(1+x\rho^{(k-2)}\cdots\rho^{(1)})^{-m_0}.$$

Proof. As with (11.2.7), we have that

$$\begin{aligned} H^{(k)}(m_0,m_1,\dots,m_{k-1};x) = \sum_{N=1}^{\infty} \frac{m_0^N}{(N-1)!} \sum_{d_1=1}^{\infty} (-x)^{d_1} \\ \times \sum_{d_2=1}^{\infty} \frac{(-x)^{d_2}}{d_2} \cdots \sum_{d_N=1}^{\infty} \frac{(-x)^{d_N}}{d_N} \bar{P}^{(k)}_{d_1,\dots,d_N}(m_1,\dots,m_{k-1}), \end{aligned}$$

where $\bar{P}^{(k)}_{d_1,\dots,d_N}(m_1,\dots,m_{k-1})$ is the same summation over permutation sequences as $P^{(k)}_{d_1,\dots,d_N}(m_1,\dots,m_{k-1})$ in (12.2.1), but with only the transitive cases (the valid constellations) included. This satisfies

$$P^{(k)}_{d_1,\dots,d_N}(m_1,\dots,m_{k-1}) = \sum_{u\subset\{d_2,\dots,d_N\}} \bar{P}^{(k)}_{d_1,u_1,u_2,\dots}(m_1,\dots,m_{k-1}) P^{(k)}_{\bar{u}_1,\bar{u}_2,\dots}(m_1,\dots,m_{k-1}) \tag{12.2.4}$$

as in Lemma 11.2.2.

Thus, in clear parallel with the proof of Theorem 11.2.1, if we define

$$F^{(k)}(m_0, m_1, \ldots, m_{k-1}; x) = \sum_{N=0}^{\infty} \frac{m_0^N}{N!} \sum_{d_1=1}^{\infty} \frac{(-x)^{d_1}}{d_1} \cdots \sum_{d_N=1}^{\infty} \frac{(-x)^{d_N}}{d_N} \times P^{(k)}_{d_1,\ldots,d_N}(m_1, \ldots, m_{k-1}), \tag{12.2.5}$$

(12.2.4) gives us that

$$H^{(k)}(m_0, m_1, \ldots, m_{k-1}; x) F^{(k)}(m_0, m_1, \ldots, m_{k-1}; x) = x \frac{\partial}{\partial x} F^{(k)}(m_0, m_1, \ldots, m_{k-1}; x).$$

We have our integral expression for $P^{(k)}_{d_1,\ldots,d_N}(m_1, \ldots, m_{k-1})$ in (12.2.3), and when we substitute this into (12.2.5) we get

$$\begin{aligned} F^{(k)}(m_0, m_1, \ldots, m_{k-1}; x) &= \int \prod_{j=1}^{k-2} \mathcal{D}z^{(j)} e^{-\mathrm{Tr}(\rho^{(j)})} \sum_{N=0}^{\infty} \frac{m_0^N}{N!} \\ &\quad \times \prod_{r=1}^{N} \sum_{d_r=1}^{\infty} \frac{(-x)^{d_r}}{d_r} \mathrm{Tr}[(\rho^{(k-2)} \cdots \rho^{(1)})^{d_r}] \\ &= \int \prod_{j=1}^{k-2} \mathcal{D}z^{(j)} e^{-\mathrm{Tr}(\rho^{(j)})} \sum_{N=0}^{\infty} \frac{m_0^N}{N!} \\ &\quad \times \left(\sum_{d=1}^{\infty} \frac{(-x)^d}{d} \mathrm{Tr}[(\rho^{(k-2)} \cdots \rho^{(1)})^d] \right)^N . \end{aligned} \tag{12.2.6}$$

We simplify this in much the same manner as in Lemma (A.0.3); the summations in the integrand converge anywhere where the absolute values of the eigenvalues of $\rho^{(k-2)} \cdots \rho^{(1)}$ are all less than $1/x$. In this domain the summations simplify to

$$\begin{aligned} \sum_{N=0}^{\infty} \frac{m_0^N}{N!} \left(\sum_{d=1}^{\infty} \frac{(-x)^d}{d} \mathrm{Tr}[(\rho^{(k-2)} \cdots \rho^{(1)})^d] \right)^N &= e^{-m_0 \mathrm{Tr}[\ln(1 + x\rho^{(k-2)} \cdots \rho^{(1)})]} \\ &= \det(1 + x\rho^{(k-2)} \cdots \rho^{(1)})^{-m_0}. \end{aligned}$$

If we treat both sides of this equation as defining a formal power series in x, the fact that they are equal over a finite neighbourhood of zero means that the series are equivalent. Thus we can substitute this into (12.2.6), and we get

$$F^{(k)}(m_0, m_1, \ldots, m_{k-1}; x) = \int \prod_{j=1}^{k-2} \mathcal{D}z^{(j)} e^{-\mathrm{Tr}(\rho^{(j)})} \det(1 + x\rho^{(k-2)} \cdots \rho^{(1)})^{-m_0}.$$

□

We can't evaluate this expression any further, however, at least not using our current

methods. This is due to the fact that the determinant cannot in general be written purely as a function of the matrices' eigenvalues, as it could when $k = 3$. What we can do, however, is perform the same sub-summation trick from Section 11.2.6, this time by setting $m_{k-1} = 1$. When we do this, each $z^{(j)}$ is reduced to a complex m_j-dimensional vector, with $\rho^{(j)}$ becoming its (scalar) squared modulus. The integrals become vector integrals, which become radial integrals due to the integrand depending only on the length of the vectors. Thus, if we denote $|z^{(j)}|$ as r_j, we get

$$\begin{aligned}\int \mathcal{D}z^{(j)} &\equiv \frac{1}{\pi^{m_j}} \int_{\mathbb{C}^{m_j}} d^{m_j} z^{(j)} d^{m_j} \bar{z}^{(j)} \\ &\to \frac{2}{\Gamma(m_j)} \int_0^\infty r_j^{2m_j - 1} dr_j,\end{aligned}$$

and the integral becomes

$$\begin{aligned}F^{(k)}(m_0, \ldots, m_{k-2}, 1; x) &= \int \prod_{j=1}^{k-2} \mathcal{D}z^{(j)} e^{-r_j^2} \frac{1}{(1 + x r_1^2 r_2^2 \cdots r_{k-2}^2)^{m_0}} \\ &= \sum_{d=0}^{\infty} \frac{\Gamma(m_0 + d)}{d! \Gamma(m_0)} (-x)^d \prod_{j=1}^{k-2} \frac{2}{\Gamma(m_j)} \int_0^\infty r_j^{2m_j + 2d - 1} e^{-r_j^2} dr_j \\ &= \sum_{d=0}^{\infty} \frac{(-x)^d}{d!} \prod_{j=0}^{k-2} \frac{\Gamma(m_j + d)}{\Gamma(m_j)} \\ &= \sum_{d=0}^{\infty} \frac{(-x)^d}{d!} \prod_{j=0}^{k-2} (m_j)_d.\end{aligned}$$

Thus, splitting $H^{(k)}(m_0, \ldots, m_{k-1}; x)$ up into degree-specific generating functions as

$$H^{(k)}(m_0, \ldots, m_{k-1}; x) = \sum_{d=1}^{\infty} (-x)^d H_d^{(k)}(m_0, \ldots, m_{k-1}),$$

we get the recursion relation

$$H_d^{(k)}(m_0, \ldots, m_{k-2}, 1) = \frac{1}{(d-1)!} \prod_{j=0}^{k-2} (m_j)_d - \sum_{i=1}^{d-1} \frac{1}{(d-i)!} \prod_{j=0}^{k-2} (m_j)_{d-i} H_i^{(k)}(m_0, \ldots, m_{k-2}, 1).$$

Furthermore, if we set all the parameters to unity, defining $H_d^{(k)} = H_d^{(k)}(1, \ldots, 1)$, we get

$$H_d^{(k)} = d \cdot (d!)^{k-2} - \sum_{i=1}^{d-1} [(d-i)!]^{k-2} H_i^{(k)}. \tag{12.2.7}$$

This once more highlights the link between rooted constellations and subsets of free groups [17], which we already noted in the $k = 3$ case in Theorem 11.2.6, and which we can state more generally as follows:

Theorem 12.2.2. *There is a bijection between the set of rooted k-constellations with degree d and the set of subgroups of index d in the free group of rank $k-1$.*

Proof. An index-d subgroup of the free group of rank $k-1$ is represented by a set of d-permutations $P_1, \ldots, P_{k-1}$ which generate a transitive group, and any two such representations are isomorphic if they can be obtained from one another by replacing the indices $(1, 2, \ldots, d)$ with $(1, c_2, \ldots, c_d)$ within the permutations for some ordering $(c_2, \ldots, c_d)$ of $(2, \ldots, d)$. [17]. Such a set of permutations, when combined with the inverse of their product, form a k-constellation with degree d, and the subgroup isomorphism given is identical to the isomorphism for rooted constellations, so each such subgroup corresponds uniquely to a rooted k-constellation with degree d. □

12.2.2 One-face constellations

In Section 12.1.1 we noted an intriguing pattern. At that point we had two recursion relations – (11.2.25) for giving H_d, the number of rooted hypermaps with d darts, and (12.1.3) for giving M_e, the number of rooted maps with e edges – and both of these featured intermediate expressions, namely $F_d(1,1,1) = d!$ and $G_e(1,1,1) = (2e-1)!!$ respectively, which turn out to be respectively the number of rooted (hyper)maps with one face. No reason for this was apparent, but the question remains of how general a pattern this is. Here we will show that it at least applies to all other lengths of rooted constellation as well.

We first need to define what a constellation with one face actually is, as the geometric interpretation associated with hypermaps does not readily extend to other constellations. There is a natural choice in this case, however: for hypermaps, having one face is equivalent to the first permutation in its constellation representation being a 1-cycle. We therefore also define a one-face constellation as one where its first permutation is a one-cycle.

Enumerating only one-face rooted constellations is easy; just as was the case for hypermaps, we already have their generating functions in $P_d^{(k)}(m_1, \ldots, m_{k-1})$. When we set $m_1 = \ldots = m_{k-1} = 1$ in (12.2.2) we get

$$\begin{aligned} P_d^{(k)}(1, \ldots, 1) &= \int_0^\infty 2r_1 dr_1 e^{-r_1^2} r_1^{2d} \cdots \int_0^\infty 2r_{k-1} dr_{k-1} e^{-r_{k-1}^2} r_{k-1}^{2d} \\ &= (d!)^{k-1}, \end{aligned}$$

where $r_i = |z^{(i)}|$.

Therefore, there are a total of $(d!)^{k-2}$ rooted one-face k-constellations with d darts. As predicted, these values are equal to the intermediate values in the recursion (12.2.7) for computing $H_d^{(k)}$, arising from the fact that

$$F^{(k)}(1, \ldots, 1; x) = \sum_{d=0}^{\infty} (d!)^{k-2} (-x)^d.$$

Strangely, however, there is no obvious reason for this to be the case. Evaluating

$F^{(k)}(1,\ldots,1;x)$ does not just trivially lead to the expression

$$\sum_{d=0}^{\infty} P_d^{(k)}(1,\ldots,1)(-x)^d,$$

the values arise by some other route instead. The form of the recursions (12.2.7) and (12.1.3) suggests that there could be some combinatorial interpretation of this fact, but we can only give a very rough idea of what form this may take.

Proposition 12.2.1. *There exists a mapping from pairs of rooted k-constellations (A, B) – where A has exactly one face and the degrees of A and B sum to d – to rooted k-constellations of degree d with one face, such that each such constellation is mapped to in exactly d ways.*

Proposition 12.2.2. *There exists a mapping from pairs of rooted maps (A, B) – where A has exactly one face and the edge counts of A and B sum to e – to rooted constellations with e edges and one face, such that each such map is mapped to in exactly $2e$ ways.*

These say nothing of what these mappings would be; they just assert their existence. Given the way the counts match up, such a mapping is guaranteed to exist, even if it is entirely arbitrary how the results correspond to the inputs. One can only hope that there is a mapping with some clear, meaningful structure.

Chapter 13

Enumeration discussion

In summary, what we have presented here is a new method of enumerating certain subsets of the rooted constellations, specifically rooted hypermaps and maps, as well as the general rooted constellations themselves. The end result in each of these cases is a matrix-integral expression in terms of which a generating function can be defined, and these generating functions partition the objects being enumerated both by their degree and by the cycle counts of their constituent permutations. By looking at these three examples we have gained some idea of how broadly applicable the method is, and have also encountered some of its limitations.

This method works by writing constellations in the form $[\sigma, g_1, g_2, (\sigma g_1 g_2)^{-1}]$ (using 4-constellations as an example), where σ, g_1 and g_2 are permutations. For a given σ we then construct an integral expression for a generating function enumerating all such constellations, using the fact that Gaussian matrix integrals can be evaluated as sums over permutations, as we demonstrated in Chapter 10. We then sum these generating functions over all possible cycle structures of σ.

In this final step we have the option of applying restrictions to the structure of σ. For example, in Chapter 10 we set σ as a one-cycle to enumerate only rooted hypermaps with one face, and in Chapter 12 we looked at cases where σ only had cycles of length two in order to enumerate rooted maps. If other objects can be similarly expressed as constrained rooted constellations, then our methods could also be applied to them.

The important feature of the resulting generating functions is that, as well as partitioning the objects being enumerated by the number of cycles in σ and in the various g_i, it also partitions by the number of cycles in the final permutation, $(\sigma g_1 \cdots)^{-1}$. In the example of rooted hypermaps, this means that for each partition we know how many vertices, edges and faces they have, as well as the number of darts. Finding this information is not obviously easy to do, given that the number of cycles in a product of permutations cannot be computed from the cycle counts of the constituent permutations. However, it turns out that this information is already encoded in our matrix integrals. Take, for example, the functions

$$P_{d_1,\ldots,d_N}(m_1, m_2) = \frac{\Gamma(m_1 m_2 + d)}{2\pi^{m_1 m_2}} \int_{S^{2m_1 m_2 - 1}} d\Omega \prod_{r=1}^{N} \mathrm{Tr}[(\hat{\rho}_1^{12})^{d_r}]$$

used in the rooted hypermap/map cases. The integrand depends only on the cycle structure of σ, given by the cycle lengths $(d_1, \ldots, d_N)$, and the summation over g then arises from the process of evaluating the integral, but the interactions between the various matrix elements in the integrand also introduces dependence on the product σg, from which we obtain its cycle count. This fact applies to all the cases we have looked at here, and it is clear in that any situation in which this method applies we will again be able to extract all available cycle counts in the same way.

Once we have these generating functions, it is also possible to then sum over one or more of the cycle counts, simply by setting parameters in the generating functions to unity. Indeed, doing so often results in expressions which are much simpler than the overall generating function, as we saw in Chapter 12.

This brings us on to one of the method's limitations, however. While we are in general able to find matrix integral expressions for our generating functions, our ability to compute them is usually limited. The case of one-face rooted hypermaps is the only case where we were able to find a simple closed-form expression for the generating functions, allowing them to be directly expanded as polynomial functions. In the general rooted hypermap and map cases, we had to define the generating functions using a recursive procedure, in terms of other functions written as matrix integrals. Furthermore, while we were able to evaluate these integrals in summation form, we could not do so in a way that allowed us to directly expand them as polynomials, and we had to use interpolation to gain their polynomial forms instead.

The worst offender, however, was the general constellation case, where we couldn't convert the matrix integrals to summations at all. This was due to the fact that the integrals were now over *multiple* independent spaces of matrices, the number of matrices involved being equal to the number of g-permutations. This then rendered all the tools we had used previously ineffective. It did not prevent us from gaining any useful information, however, as by setting one of the parameters in the generating function to unity (summing over one of the cycle counts) we could reduce the dimensions of the matrices to one, turning the matrix integrals into scalar integrals which could be evaluated.

Another limitation is that there are some specific properties which the objects being enumerated must have for these methods to apply. For one thing, in the case of maps and hypermaps, they had to be orientable, as the construction of their constellation representations is very dependent on them being embedded on orientable surfaces. The other, broader point, which applies to general constellations as well, is that they have to be rooted (i.e. isomorphism classes under transformations which preserve the root). This ties in with the construction of the integral generating functions, and is needed to make sure the summations over the various g-permutations resulting from said integrals has a one-to-one correspondence to the isomorphism classes being enumerated. This is not necessarily a problem even if one wishes to study a different type of isomorphism class, as i.e. the existence of any rooted maps with a given set of properties implies, and is implied by the existence of an unrooted map with the same properties. So

these methods could be used for questions of existence, where the exact counts are not important beyond their being non-zero. The counts themselves are only applicable when specifically rooted objects are being considered though.

Beyond the question of the general applicability of these methods, we have also found a number of specific results in the cases studied. One of the advantages of enumerating via generating functions is that they can be used to prove general trends, and we have managed to derive a number of such trends, such as calculating the total number of objects of each type with a given number of darts (or equivalently by number of edges for maps). In the one-face rooted hypermap case this comes out to there being $d!$ such hypermaps, reflecting their equivalence to permutations, and in the general cases of hypermaps, maps and constellations, we were able to find recursion relations for computing the overall counts. Many of these trends are proven in ways which circumvent the limitations of our methods (i.e. by setting parameters in the generating functions to unity, as mentioned earlier, such that the dimensions of sums and integrals are reduced to one). It is this, and other such uses of generating functions, which give these methods their power.

Appendices

Appendix A

Miscellaneous identities

Lemma A.0.1.

$$\sum_{i=0}^{a} \binom{d}{i} (-1)^i = \binom{d-1}{a} (-1)^a \tag{A.0.1}$$

for any integer d and $a \geq 0$.

Proof. (A.0.1) is true for $a = 0$, as both sides equal unity. This holds for $d \leq 0$ as well as the binomial coefficients can be generalised using the reflection identity

$$\binom{-d}{i} = (-1)^i \binom{d+i-1}{i}.$$

If we assume (A.0.1) is true for $a = k$, then when $a = k + 1$,

$$\begin{aligned}
\sum_{i=0}^{k+1} \binom{d}{i} (-1)^i &= \sum_{i=0}^{k} \binom{d}{i} (-1)^i + \binom{d}{k+1} (-1)^{k+1} \\
&= \binom{d-1}{k} (-1)^k - \binom{d}{k+1} (-1)^k \\
&= \binom{d-1}{k+1} (-1)^{k+1}.
\end{aligned}$$

Therefore, by induction, (A.0.1) holds for all $a \geq 0$. □

Lemma A.0.2.

$$\int_0^\infty a e^{-at} \ln t \, dt = -\gamma - \ln a$$

for $a \geq 0$, where γ is the Euler-Mascheroni constant.

Proof. This integral is related to the integral definition of the gamma function:

$$\begin{aligned}\int_0^\infty ae^{-at}\ln t dt &= \frac{d}{dz}\int_0^\infty ae^{-at}t^{z-1}dt\bigg|_{z=1}\\ &= \frac{d}{dz}\left(\frac{1}{a^{z-1}}\int_0^\infty e^{-t}t^{z-1}dt\right)\bigg|_{z=1}\\ &= \frac{d}{dz}\frac{\Gamma(z)}{a^{z-1}}\bigg|_{z=1}\\ &= \frac{\Gamma(z)}{a^{z-1}}(\psi(z)-\ln a)\bigg|_{z=1}\end{aligned}$$

where $\psi(z) = \Gamma'(z)/\Gamma(z)$ is the digamma function. When we set $z = 1$, we use the fact that $\Gamma(1) = 1$ and $\psi(1) = -\gamma$ to get

$$\int_0^\infty ae^{-at}\ln t dt = -\gamma - \ln a.$$

□

Lemma A.0.3. *The two formal power series in x*

$$A(q_1,\ldots,q_N;m_0;x) = \sum_{N=0}^{\infty}\frac{m_0^N}{N!}\left(\sum_{d=1}^{\infty}\frac{(-x)^d}{d}\sum_{a=1}^{m_1}q_a^d\right)^N \tag{A.0.2}$$

and

$$B(q_1,\ldots,q_N;m_0;x) = \prod_{a=1}^{m_1}\frac{1}{(1+q_ax)^{m_0}} \tag{A.0.3}$$

are equivalent.

Proof. (A.0.2) does not converge for all values of its parameters, but there is a domain for which it does converge absolutely, namely when $|q_ax| < 1$ for all $1 \le a \le m_1$. To show that these two formal power series are equivalent, it therefore suffices to show that they equal each other in the range where they both converge absolutely.

When (A.0.2) does converge, we can simplify it as follows:

$$\begin{aligned}A(q_1,\ldots,q_N;m_0;x) &= \sum_{N=0}^{\infty}\frac{m_0^N}{N!}\left(-\sum_{a=1}^{m_1}\ln(1+q_ax)\right)^N\\ &= \sum_{N=0}^{\infty}\frac{m_0^N}{N!}\left[-\ln\left(\prod_{a=1}^{m}(1+q_ax)\right)\right]^N\\ &= \exp\left[-m_0\ln\left(\prod_{a=1}^{m}(1+q_ax)\right)\right]\\ &= \prod_{a=1}^{m_1}\frac{1}{(1+q_ax)^{m_0}},\end{aligned}$$

which is exactly (A.0.3). □

Appendix B

Exact evaluation of $\langle S_{2,2} \rangle$

In this appendix we will exactly evaluate the summation

$$\langle S_{2,2} \rangle = \ln 2 - \sum_{k=1}^{\infty} \frac{3[1-(-1)^k]}{2k(k+1)(k+4)}$$

derived in Section 6.1.1 of Part II.

We begin by expanding the summand out using partial fractions:

$$\langle S_{2,2} \rangle = \ln 2 - \sum_{k=1}^{\infty} 3[1-(-1)^k] \left(\frac{1}{8k} - \frac{1}{6(k+1)} + \frac{1}{24(k+4)} \right).$$

This essentially splits the expression into three separate infinite summations. We cannot separate them though, as they would be individually divergent. However, if we shift the sums relative to each other, then many of the terms can be cancelled out:

$$\begin{aligned} \langle S_{2,2} \rangle &= \ln 2 - \sum_{k=1}^{\infty} \left(\frac{3[1-(-1)^k]}{8k} - \frac{1+(-1)^k}{2k} + \frac{1-(-1)^k}{8k} \right) \\ &\quad + \sum_{k=1}^{4} \frac{1-(-1)^k}{8k} \\ &= \frac{1}{3} + \ln 2 - \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k}. \end{aligned} \tag{B.0.1}$$

This shift does not affect the value of the summation by the following argument. If we take the partial sums of the original and shifted series up to K terms, given by

$$\begin{aligned} \Sigma_1 &= \ln 2 - \sum_{k=1}^{K} 3[1-(-1)^k] \left(\frac{1}{8k} - \frac{1}{6(k+1)} + \frac{1}{24(k+4)} \right) \\ \Sigma_2 &= \frac{1}{3} + \ln 2 - \sum_{k=1}^{K} \frac{(-1)^{k-1}}{k}, \end{aligned}$$

then the difference between them is

$$\Sigma_2 - \Sigma_1 = \sum_{k=K+1}^{K+1} \left(-\frac{1}{2k} - \frac{(-1)^k}{2k} \right) + \sum_{k=K+1}^{K+4} \left(\frac{1}{8k} - \frac{(-1)^k}{8k} \right) = \mathcal{O}(K^{-1}),$$

so in the limit as $K \to \infty$, the partial sums tend to the same value.

The remaining summation in (B.0.1) is now recognisable as the Taylor expansion of $\ln(1+z)$ at $z = 1$, which converges to $\ln 2$, so

$$\langle S_{2,2} \rangle = \frac{1}{3}.$$

Appendix C

Confirmation of Page's entropy formula

In this appendix we re-prove Page's exact formula (6.2.1) for the mean entropy of entanglement of a bipartite quantum system in a pure state $\langle S_{m_1 m_2} \rangle$. This proof stands in addition to existing ones [16, 27, 30], but does not evaluate the entropy integral directly as they did.

Theorem C.0.3. *For any $m_1, m_2 \geq 2$,*

$$\langle S_{m_1 m_2} \rangle = \sum_{k=m_2+1}^{m_1 m_2} \frac{1}{k} - \frac{m_1 - 1}{2m_2}.$$

Proof. For our proof here, we use the fact that

$$\langle S_{m_1 m_2} \rangle = -\langle \mathrm{Tr}[\hat{\rho}_1^{12} \ln \hat{\rho}_1^{12}] \rangle = -\lim_{d \to 1} \frac{\partial}{\partial d} \langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle.$$

We already know from Theorem 5.2.1 in Section 5.2.1 of Part II that

$$\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = \frac{\Gamma(m_1 m_2)}{d\Gamma(m_1 m_2 + d)} \sum_{r=0}^{m_1 - 1} \frac{(-1)^r}{r!\Gamma(d-r)} \frac{\Gamma(m_1 + d - r)}{\Gamma(m_1 - r)} \frac{\Gamma(m_2 + d - r)}{\Gamma(m_2 - r)}.$$

for any non-integer $d > 0$, and we also know the trivial special case that $\langle \mathrm{Tr}[(\hat{\rho}_1^{12})^1] \rangle = 1$. Differentiating the above gives us that

$$\begin{aligned} -\frac{\partial}{\partial d} \langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle = & \left(\frac{1}{d} + \psi(m_1 m_2 + d) \right) \langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle \\ & - \frac{\Gamma(m_1 m_2)}{d\Gamma(m_1 m_2 + d)} \sum_{r=0}^{m_1 - 1} \frac{(-1)^r}{r!\Gamma(d-r)} \frac{\Gamma(m_1 + d - r)}{\Gamma(m_1 - r)} \\ & \times \frac{\Gamma(m_2 + d - r)}{\Gamma(m_2 - r)} \left(\psi(m_1 + d - r) \right. \\ & \left. + \psi(m_2 + d - r) - \psi(d - r) \right) \end{aligned}$$

where $\psi(z) = \frac{\partial}{\partial z} \ln \Gamma(z)$ is the digamma function.

We must now be careful as we take the limit $d \to 1$. The $\Gamma(d-r)$ term in the summand means that most terms in the summation will vanish when $r > 0$, with the exception of the term proportional to $\psi(d-r)/\Gamma(d-r)$, which is indeterminate at $d = 1$ for integer $r > 0$. In these cases,

$$\lim_{d\to 1} \frac{\psi(d-r)}{\Gamma(d-r)} = -\lim_{d\to 1} \frac{\partial}{\partial d} \frac{1}{\Gamma(d-r)} = -\left[\mathrm{res}_{z=1-r}\Gamma(z)\right]^{-1} = (-1)^r \Gamma(r).$$

Therefore,

$$\begin{aligned}
\langle S_{m_1 m_2} \rangle &= -\lim_{d\to 1} \frac{\partial}{\partial d} \langle \mathrm{Tr}[(\hat{\rho}_1^{12})^d] \rangle \\
&= 1 + \psi(m_1 m_2 + 1) - \psi(m_1 + 1) - \psi(m_2 + 1) + \psi(1) \\
&\quad + \frac{1}{m_1 m_2} \sum_{r=1}^{m_1 - 1} \frac{1}{r} (m_1 - r)(m_2 - r) \\
&= \psi(m_1 m_2 + 1) - \psi(m_1 + 1) - \psi(m_2 + 1) + \psi(1) \\
&\quad + \sum_{r=1}^{m_1} \frac{1}{r} - \frac{m_1 - 1}{2m_2}.
\end{aligned}$$

Given also the fact that [1, p 258]

$$\psi(N) = -\gamma + \sum_{k=1}^{N-1} \frac{1}{k}$$

for integers N,

$$\begin{aligned}
\langle S_{m_1 m_2} \rangle &= \sum_{k=1}^{m_1 m_2} \frac{1}{k} - \sum_{k=1}^{m_1} \frac{1}{k} - \sum_{k=1}^{m_2} \frac{1}{k} + \sum_{k=1}^{m_1} \frac{1}{k} - \frac{m_1 - 1}{2m_2} \\
&= \sum_{k=m_2+1}^{m_1 m_2} \frac{1}{k} - \frac{m_1 - 1}{2m_2}.
\end{aligned}$$

□

Appendix D

Some counts of rooted hypermaps

$d = 1$

	1
1	1

$d = 2$

	1	**2**
1		1
2	1	

$d = 3$

	1	**2**	**3**
1	1		1
2		3	
3	1		

$d = 4$

	1	**2**	**3**	**4**
1		5		1
2	5		6	
3		6		
4	1			

$d = 5$

	1	**2**	**3**	**4**	**5**
1	8		15		1
2		40		10	
3	15		20		
4		10			
5	1				

$d = 6$

	1	**2**	**3**	**4**	**5**	**6**
1		84		35		1
2	84		175		15	
3		175		50		
4	35		50			
5		15				
6	1					

$d = 12$

	1	**2**	**3**	**4**	**5**	**6**	$\cdots$
1		19 056 960		18 128 396		2 641 925	
2	19 056 960		75 220 860		24 656 775		$\ddots$
3		75 220 860		66 805 310		8 654 646	
4	18 128 396		66 805 310		19 324 305		$\ddots$
5		24 656 775		19 324 305		1 981 980	
6	2 641 925		8 654 646		1 981 980		$\ddots$
7		1 585 584		990 990		60 984	
8	88 803		235 950		32 670		
9		23 595		9 075			
10	715		1 210				
11		66					
12	1						

$d = 49$

	1	$\cdots$
1	4965566237014429068344915618949350015408594541871104000000000	$\cdots$
3	4577891503208884472952882199105457309783499129634160640000000	$\cdots$
5	5202310457084388884858448388683318495756691504704887193600000	$\cdots$
$\vdots$	$\vdots$	$\ddots$

Table D.0.1: Sample coefficients of various $P_d(m_1, m_2)$, indexed by the exponents of $m_1$and m_2 (which exponent is which is unimportant, as all the functions are symmetric). The $d = 12$ case in particular is given to aid comparison with Walsh's enumeration [35], while a few values from $d = 49$ are given to show the method's usability even at high degree.

Appendix E

Multiple counting of two-face rooted hypermaps

Here we prove that, using the normal form for rooted hypermaps associated with ladder diagrams, each rooted two-face hypermap can be represented in exactly j distinct ways, where j is the length of the second (non-rooted) cycle in the permutation σ.

Consider two-face rooted hypermaps of the form $[\sigma, g, (\sigma g)^{-1}]$, where $\sigma = (1 \dots i)(i+1 \dots i+j)$. For example, if $i = 2$ and $j = 4$ then $\sigma = (12)(3456)$. As we are restricting ourselves to this normal form, the number of isomorphisms we need to consider is also restricted. Specifically we only need to look at isomorphisms given by permutations p such that $p(1) = 1$ and $p\sigma p^{-1} = \sigma$, and the only permutations which satisfy both of these properties are integer powers of $(1)(2)\dots(i)(i+1\dots i+j)$, i.e. permutations which perform a cyclic reordering of the indices in the second cycle only. There are of course j such permutations.

Now consider some given p of this form, and a particular rooted hypermap $[\sigma, g, (\sigma g)^{-1}]$. Given that this hypermap is transitive, there must be a cycle in g (which we will denote χ) which contains indices from both cycles of σ. When we apply the isomorphism p to g and look at the effect it has on χ, we see that the indices from the first cycle are preserved, while the indices from the second cycle are preserved if and only if p is the identity, being mapped to a distinct index in the second cycle otherwise. As a result, the image of χ after applying p will match to a cycle in g if and only if p is the identity. Thus any non-trivial p maps the hypermap to a distinct representation of an isomorphic rooted hypermap. The hypermap's isomorphism class therefore contains exactly j distinct representations, one for each possible p.

Appendix F

Inner products of general orthogonal polynomials

Here we will look at a result relating to orthogonal polynomials for arbitrary inner products.

Consider the inner product

$$\langle a, b\rangle = \int_0^\infty a(q)b(q)w(q)dq,$$

defined on the space of polynomial functions, for some weight function $w(q)$. This weight function has an associated set of moments

$$\mu_n = \int_0^\infty q^n w(q)dq.$$

We then define an associated set of monic polynomials $K_n(q)$ each of order n, which are orthogonal in the sense that $\langle K_i, K_j\rangle = 0$ whenever $i \neq j$.

We are interested here in the constants $I_n = \langle K_n, K_n\rangle$. These are dependent only on $w(q)$, and we show here that they can be computed just from the moments μ_n without having to evaluate the polynomials themselves.

Theorem F.0.4. *The inner products*

$$I_n = \langle K_n, K_n\rangle = \begin{cases} \mu_0, & n = 0 \\ |\mathbf{M}_{n+1}|/|\mathbf{M}_n|, & n > 0 \end{cases}$$

where

$$\mathbf{M}_n = \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{n-1} \\ \mu_1 & \mu_2 & \cdots & \mu_n \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \cdots & \mu_{2n-2} \end{bmatrix}.$$

Equivalently,

$$I_n = \frac{M(n+1)}{M(n)}$$

for all $n \geq 0$, where we define

$$M(n) = \begin{cases} 1, & n = 0 \\ |\mathbf{M}_n|, & n > 0 \end{cases}.$$

Proof. The case when $n = 0$ is simple to prove; as the $K_n(q)$ are monic, $K_0(q) = 1$, so

$$I_0 = \int_0^\infty 1 \cdot w(q) dq = \mu_0.$$

When $n > 0$, the orthogonal polynomials can be written in terms of the moments:

$$K_n(q) = \begin{vmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_n \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \mu_{n+1} & \cdots & \mu_{2n-1} \\ 1 & q & q^2 & \cdots & q^n \end{vmatrix} \Bigg/ \begin{vmatrix} \mu_0 & \mu_1 & \cdots & \mu_{n-1} \\ \mu_1 & \mu_2 & \cdots & \mu_n \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \cdots & \mu_{2n-2} \end{vmatrix}$$

$$= \frac{1}{|\mathbf{M}_n|} \begin{vmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_n \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \mu_{n+1} & \cdots & \mu_{2n-1} \\ 1 & q & q^2 & \cdots & q^n \end{vmatrix}.$$

To see that this is the correct polynomial, evaluate $\langle q^i, K_j \rangle$ for any $0 \leq i < j$. For example,

$$\langle q^i, K_j \rangle \propto \int_0^\infty \begin{vmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_n \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \mu_{n+1} & \cdots & \mu_{2n-1} \\ q^{i+1} & q^{i+2} & q^{i+3} & \cdots & q^{i+n} \end{vmatrix} w(q) dq$$

$$= \begin{vmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_n \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \mu_{n+1} & \cdots & \mu_{2n-1} \\ \mu_{i+1} & \mu_{i+2} & \mu_{i+3} & \cdots & \mu_{i+n} \end{vmatrix}.$$

For any $0 \leq i < j$ two of the rows in this matrix are equal, so the determinant is zero. Thus $K_j(q)$ is orthogonal to all polynomials of order less than j, as expected.

Now consider $I_n = \langle K_n, K_n \rangle$. As K_n is orthogonal to all polynomials of order less than n we can subtract away all lower-order terms from the left hand side until only

the leading term remains without changing the value. This means that

$$
\begin{aligned}
I_n &= \langle q^n, K_n \rangle \\
&= \frac{1}{|\mathbf{M}_n|} \int_0^\infty \begin{vmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_n \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \mu_{n+1} & \cdots & \mu_{2n-1} \\ q^n & q^{n+1} & q^{n+2} & \cdots & q^{2n} \end{vmatrix} w(q) dq \\
&= \frac{1}{|\mathbf{M}_n|} \begin{vmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_n \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \mu_{n+1} & \cdots & \mu_{2n-1} \\ \mu_n & \mu_{n+1} & \mu_{n+2} & \cdots & \mu_{2n} \end{vmatrix} \\
&= \frac{|\mathbf{M}_{n+1}|}{|\mathbf{M}_n|}.
\end{aligned}
$$

□

Corollary F.0.1. *For any $n > 0$,*

$$\prod_{a=1}^{n} I_{a-1} = |\mathbf{M}_n|.$$

Proof. This follows from Theorem F.0.4:

$$
\begin{aligned}
\prod_{a=1}^{n} I_{a-1} &= I_0 I_1 \cdots I_{n-2} I_{n-1} \\
&= \frac{M(1)}{M(0)} \frac{M(2)}{M(1)} \cdots \frac{M(n-1)}{M(n-2)} \frac{M(n)}{M(n-1)} \\
&= \frac{M(n)}{M(0)} \\
&= |\mathbf{M}_n|.
\end{aligned}
$$

□

Appendix G

Confirmation of hypermap subgroup counting

Here we confirm the result (11.2.24), by using the bijection proven in Theorem 11.2.6 between rooted hypermaps and subgroups of the rank 2 free group. This proof parallels the proof of Theorem 5.2 in [17], but while Hall looked at the total number of subgroups of a given index, we will now look at a generating function which partitions the subgroups by the cycle structure of their permutation representations (i.e. by the number of vertices and faces in the equivalent rooted hypermaps.

Theorem G.0.5. *The set of rooted hypermaps with d darts is enumerated by the generating function $H_d(m_0, 1, m_2)$, where*

$$H_d(m_0, 1, m_2) = \frac{(m_0)_d (m_2)_d}{(d-1)!} - \sum_{k=1}^{d-1} \frac{(m_0)_{d-k} (m_2)_{d-k}}{(d-k)!} H_k(m_0, 1, m_2)$$

and $H_1(m_0, 1, m_2) = m_0 m_2$, and the generating function partitions the hypermaps by number of faces and vertices.

Proof. We have that each rooted d-hypermap is a 3-constellation $[g_1, (g_2 g_1)^{-1}, g_2]$, and that two such rooted hypermaps are isomorphic if they are equivalent under rearrangement of the last $d-1$ indices in the permutations g_1 and g_2. The number of faces in each hypermap is the number of cycles in g_1, and the number of vertices is g_2. For the sake of this proof we will just think of rooted hypermaps as the pair consisting of g_1 and g_2.

The generating function

$$\frac{\Gamma(m_0 + d)}{\Gamma(m_0)} \frac{\Gamma(m_2 + d)}{\Gamma(m_2)} \equiv (m_0)_d (m_2)_d$$

enumerates all pairs of d-permutations by cycle structure (see Theorem 10.1.1). This counts cases where the group they generate is not transitive, however, so we need to subtract all these cases to leave only those which are valid hypermaps.

For a particular permutation pair (g_1, g_2), let $(1, b_2, \ldots, b_k)$ be the transitive constituent which includes the identity (i.e. the actions of both g_1 and g_2 on $(1, b_2, \ldots, b_k)$ are themselves permutations on that set, which generate a transitive group). These transitive sub-permutation pairs are by definition enumerated by $H_k(m_0, 1, m_2)$, while the remaining part, which are just a pair of unconstrained permutations on $d - k$ elements, are enumerated by $(m_0)_{d-k}(m_2)_{d-k}$.

For any given k there are $(d-1)!/(k-d)!$ possible choices for $(1, b_2, \ldots, b_k)$ i.e. $k-1$ elements chosen from $d-1$ elements where their order matters. Thus, summing over all possible variants, we get

$$(m_0)_d(m_2)_d = \sum_{k=1}^{d} \frac{(d-1)!}{(d-k)!}(m_0)_{d-k}(m_2)_{d-k}H_k(m_0, 1, m_2). \tag{G.0.1}$$

Rearranging and dividing by $(d-1)!$ gives

$$H_d(m_0, 1, m_2) = \frac{(m_0)_d(m_2)_d}{(d-1)!} - \sum_{k=1}^{d-1} \frac{(m_0)_{d-k}(m_2)_{d-k}}{(d-k)!} H_k(m_0, 1, m_2).$$

Finally, the special case $H_1(m_0, 1, m_2) = m_0 m_2$ follows from substituting $d = 1$ into (G.0.1). □

Bibliography

[1] M. Abramowitz, I. Stegun, Handbook of Mathematical Functions, Dover Publications, 1965.

[2] M. Aguiar, F. Sottile, Structure of the Malvenuto-Reutenauer Hopf algebra of permutations, Adv. Math. (N. Y). 191 (2005) 225–275.

[3] D. Arquès, Hypercartes pointées sur le tore: Décompositions et dénombrements, J. Comb. Theory, Ser. B 43 (1987) 275–286.

[4] D. Arquès, Relations fonctionnelles et dénombrement des cartes pointées sur le tore, J. Comb. Theory, Ser. B 43 (1987) 253–274.

[5] I. Bengtsson, K. Zyczkowski, Geometry of quantum states: an introduction to quantum entanglement, 1st Edition, Cambridge University Press, 2006.

[6] C. Bennett, D. DiVincenzo, J. Smolin, W. Wootters, Mixed-state entanglement and quantum error correction, Phys. Rev. A 54 (1996) 3824–3851.

[7] D. Bessis, C. Itzykson, J. B. Zuber, Quantum field theory techniques in graphical enumeration, Adv. Appl. Math. 1 (1980) 109–157.

[8] T. J. Bromwich, An Introduction to the Theory of Infinite Series, American Mathematical Society, 1991.

[9] L. Comtet, Sur les coefficients de l'inverse de la serie formelle Sum n! t^n, Comptes Rend. Acad. Sci. Paris, A 275 (1972) 569–572.

[10] L. Comtet, Advanced Combinatorics, D. Reidel Publishing Company, 1974.

[11] A. Erdélyi, M. Wyman, The asymptotic evaluation of certain integrals, Arch. Ration. Mech. Anal. 14 (1963) 217–260.

[12] H. Everett, The theory of the universal wave function, in: many-worlds Interpret. quantum Mech., Princeton University Press, 1973.

[13] R. P. Feynman, QED: The strange theory of light and matter, Princeton University Press, 1985.

[14] R. P. Feynman, A. R. Hibbs, Quantum mechanics and path integrals, Dover Publications, 1997.

[15] G. B. Folland, How to Integrate a Polynomial over a Sphere, Am. Math. Mon. 108 (2001) 446.

[16] S. Foong, S. Kanno, Proof of Page's conjecture on the average entropy of a subsystem, Phys. Rev. Lett. 72 (1994) 1148–1151.

[17] M. Hall, Subgroups of finite index in free groups, Canad. J. Math 1 (1949) 187–190.

[18] B. R. Heap, Permutations by Interchanges, Comput. J. 6 (1963) 293–298.

[19] M. Klazar, Irreducible and connected permutations, Inst. Teor. Inform. 122.

[20] S. K. Lando, A. K. Zvonkin, Graphs on surfaces and their applications, Springer, 2004.

[21] S. Lloyd, H. Pagels, Complexity as thermodynamic depth, Ann. Phys. (N. Y). 188 (1988) 186–213.

[22] E. Lubkin, Entropy of an n-system from its correlation with a k-reservoir, J. Math. Phys. 19 (1978) 1028.

[23] M. L. Mehta, Random Matrices, 2nd Edition, Academic Press Inc., 1991.

[24] D. Page, Average entropy of a subsystem, Phys. Rev. Lett. 71 (1993) 1291–1294.

[25] R. Penco, D. Mauro, Perturbation theory via Feynman diagrams in classical mechanics, Eur. J. Phys. 27 (2006) 1241–1249.

[26] M. Petkovšek, H. S. Wilf, D. Zeilberger, A= B, A K Peters Ltd, 1996.

[27] J. Sánchez-Ruiz, Simple proof of Page's conjecture on the average entropy of a subsystem, Phys. Rev. E 52 (1995) 5653.

[28] G. Schaeffer, E. Vassilieva, A bijective proof of Jackson's formula for the number of factorizations of a cycle, J. Comb. Theory, Ser. A 115 (6) (2008) 903–924.

[29] R. Sedgewick, Permutation Generation Methods, ACM Comput. Surv. 9 (1977) 137–164.

[30] S. Sen, Average Entropy of a Quantum Subsystem, Phys. Rev. Lett. 77 (1996) 1–3.

[31] N. J. A. Sloane, Number of connected permutations of [1..n].
URL `https://oeis.org/A003319`

[32] N. J. A. Sloane, R. Ehrenborg, G. Hetyei, A problem of configurations.
URL `https://oeis.org/A000698`

[33] G. 't Hooft, A planar diagram theory for strong interactions, Nucl. Phys. B 72 (1974) 461–473.

[34] T. R. Walsh, Generating nonisomorphic maps without storing them, SIAM J. Algebr. Discret. Methods.

[35] T. R. Walsh, Generating nonisomorphic maps and hypermaps without storing them (2012).
URL `http://accueil.labunix.uqam.ca/{~}walsh{_}t/papers/GENERATINGNONISOMORPHIC.pdf`

[36] T. R. S. Walsh, Hypermaps versus bipartite maps, J. Comb. Theory, Ser. B 18 (1975) 155–163.

[37] T. R. S. Walsh, A. B. Lehman, Counting rooted maps by Genus. I, J. Comb. Theory, Ser. B 13 (1972) 192–218.

[38] D. Zeilberger, Maple Programs Related to A=B.
URL `http://www.math.rutgers.edu/{~}zeilberg/programsAB.html`

[39] A. Zvonkin, Matrix integrals and map enumeration: An accessible introduction, Math. Comput. Model. 26 (1997) 281–304.