

The Squish Map and the SL_2 Double Dimer Model

Leigh Foster

Benjamin Young

Submitted: Oct 10, 2023; Accepted: Feb 21, 2024; Published: Mar 22, 2024

© The authors. Released under the CC BY-SA license (International 4.0).

Abstract

A plane partition, whose 3D Young diagram is made of unit cubes, can be approximated by a “coarser” plane partition, made of cubes of side length 2. Indeed, there are two such approximations obtained by “rounding up” or “rounding down” to the nearest cube. We relate this coarsening (or downsampling) operation to the squish map introduced by the second author in earlier work. We exhibit a related measure-preserving map between the dimer model on the honeycomb graph, and the SL_2 double dimer model on a coarser honeycomb graph; we compute the most interesting special case of this map, related to plane partition q -enumeration with 2-periodic weights. As an application, we specialize the weights to be certain roots of unity, obtain novel generating functions (some known, some new, and some conjectural) that (-1) -enumerate certain classes of pairs of plane partitions according to how their dimer configurations interact.

Mathematics Subject Classifications: 05A15, 05C50, 82B20

1 Introduction

Anyone who has played with cubical building blocks in their youth has, at some point, constructed a $2 \times 2 \times 2$ cube out of eight $1 \times 1 \times 1$ cubes. Later in life, some of us (including the authors) went on to study *plane partitions*, which are nothing more than stable piles of cubes in the corner of a large room. These two experiences tell us that there ought to be *some* relation between plane partitions made out of the little cubes, and plane partitions made out of the bigger ones. Or, going the other way: what if we take a picture of a plane partition and “downsample” it, by approximating its $1 \times 1 \times 1$ cubes by roughly one eighth as many $2 \times 2 \times 2$ cubes, as best we can (see Figure 1)? How much information do we lose by doing this?

We propose to answer this question through an analysis of the *squish map* – a map originally studied in [8], as a means of proving a combinatorial theorem about plane partition enumeration. Here, we prove that the squish map is a measure-preserving transformation

Department of Mathematics, University of Oregon, Eugene, Oregon, U.S.A. (leighf@uoregon.edu, bjj@uoregon.edu).

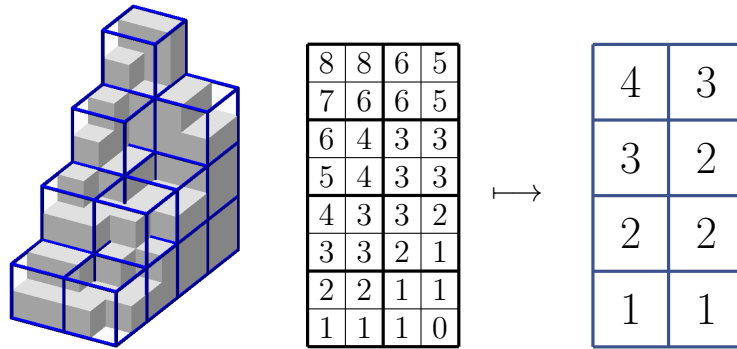


Figure 1: A plane partition, viewed as a stack of boxes - and then “downsampled” by a factor of two, rounding up to the nearest full box. The plane partitions are π and π_{\max} , respectively, from Example 8 in Section 3.3.

between instances of the single and double dimer models on honeycomb graphs. Thus, the loops of the double dimer model indicate where information is lost in the “downsampling” process.

1.1 Definitions and literature survey

A *plane partition* is an infinite matrix of nonnegative integers, all of which are zero sufficiently far from the origin, which are weakly decreasing both in rows or in columns (we do not typically draw the zeros). Equivalently, interpreting the numbers in a plane partition as z coordinates, one can represent a plane partition as a stack of unit cubes in the corner of a room - this is precisely the relationship between ordinary integer partitions and their Young diagrams, but one dimension higher.

If the plane partition’s cubes fit inside an $x \times y \times z$ box, then it is said to be a *boxed $x \times y \times z$ plane partition*. MacMahon [6] proved that the generating function for boxed $x \times y \times z$ plane partitions is

$$\prod_{i=1}^x \prod_{j=1}^y \frac{1 - q^{i+j+z-1}}{1 - q^{i+j-1}}$$

which, in the limit $x, y, z \rightarrow \infty$ gives the famous generating function for plane partitions,

$$\prod_{i \geq 1} \left(\frac{1}{1 - q^i} \right)^i.$$

Boxed plane partitions are in bijection with perfect matchings on an $x \times y \times z$ hexagon graph, and hence with the dimer model on an certain graph. This result (as far as we know) is folklore, and we don’t know a good reference; see Figure 6 for an illustration. We call the graph in question $H_{x,y,z}$, the $x \times y \times z$ honeycomb graph (see Figure 2). Let $D_{x,y,z}$ denote the set of perfect matchings (otherwise known as *dimer configurations*) on $H_{x,y,z}$. Furthermore, let $DD_{x,y,z}$ be the set of *double dimer configurations* on $H_{x,y,z}$. That

is, an element $m \in D_{x,y,z}$ is an induced 1-regular subgraph of $H_{x,y,z}$ – every vertex of $H_{x,y,z}$ is in exactly one edge of $D_{x,y,z}$, whereas an element $m \in DD_{x,y,z}$ is an induced 2-regular subgraph (with the slightly unusual convention that doubled edges are allowed). See Figure 4.

We require Kenyon’s $SL_2(\mathbb{C})$ -weighted double dimer model [4]: in addition to an edge weight, one assigns a 2×2 matrix $M_e \in SL_2(\mathbb{C})$ to each edge e ; the partition function involves a product of traces of such matrices taken around a loop. This collection of matrices is called a *connection* on the graph, by analogy with differential geometry, since its contribution to the partition function is the monodromy around closed paths in the graph. When all M_e are the identity matrix, all loops contribute $\text{Tr}(I) = 2$, so the model reduces to two independent copies of the ordinary dimer model, but this is typically a much more general and subtle model.

There are standard, determinantal tools for computing single and double dimer partition functions. For the single dimer model, one uses [3] and [7]; for lattice paths, the references are [5] and [2]. Kenyon [4] introduces a matrix, analogous to Kasteleyn’s matrix, whose determinant computes the partition function of this model. We do not need either determinant for this paper.

The map we introduce here, *the squish map*, is a map from $D_{2x,2y,2z}$ to $DD_{x,y,z}$. One can (in principle) discover everything there is to know about it by drawing $H_{2x,2y,2z}$ with distorted edge lengths (see Figure 5). It was introduced in [8]. Our result here is that the squish map is measure-preserving, for particular choices of the parameters of the single and double dimer models. Indeed, we are even able to prove a subtle result about the enumeration of down-sampled plane partitions, hinted at in [8]: If we let $x, y, z \rightarrow \infty$, then there is a closed-form generating function which is preserved by the squish map. It is a certain 4-variable generating function for “ $\mathbb{Z}_2 \times \mathbb{Z}_2$ - colored plane partitions” studied in [9]; in it, a cube at position (i, j, k) gets a color according to the parities of $(i - j)$ and $(i - k)$; the four variables keep track of how many cubes of each color there are. The squish map allows us to see this generating function as marking certain statistics on a downsampled plane partition. This generating function was studied in [9] under a certain specialization; we both study it in full generality here, including the original specialization and a few intriguing others.

2 The single and double dimer models

Definition 1. The *weight* of a graph is an assignment $\nu : E \rightarrow \mathbb{R}_{\geq 0}$ of real numbers onto each edge of the graph, where E is the set of edges in a graph G .

Definition 2. Consider a perfect matching m on G . The *weight of m* is defined to be $w(m) = \prod_{e \in m} w(e)$.

Example 3. The following weighting reproduces the $Q^{(\text{number of boxes})}$ statistic on plane partitions, up to an overall power of Q . This, as far as the authors know, is a “folklore” idea for which we do not know a good reference. Assign a weight of 1 to one horizontal edge in each column of hexagons. Above each horizontal edge with weight 1, assign

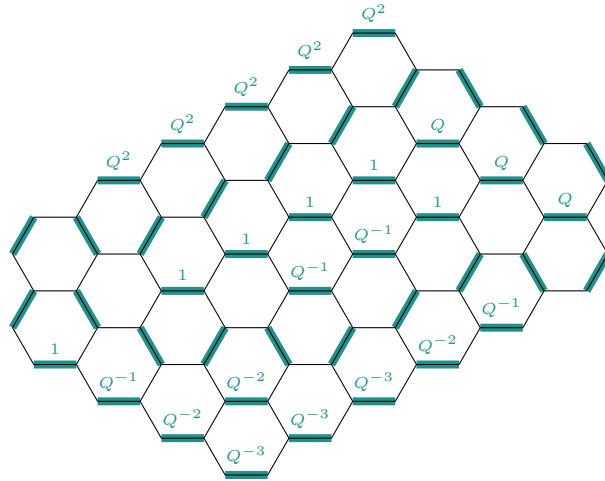


Figure 2: A perfect matching (dimer configuration) on $H_{6,2,4}$, with a monochromatic weighting on the horizontal edges. Note that although the edges labeled 1 are lined up along a particular diagonal, this does not have to be this case.

weights Q, Q^2, Q^3, \dots , and below weights Q^{-1}, Q^{-2}, \dots . Non-horizontal edges get weight 1. Then the overall weight of the perfect matching in Figure 2 is Q^{-6} .

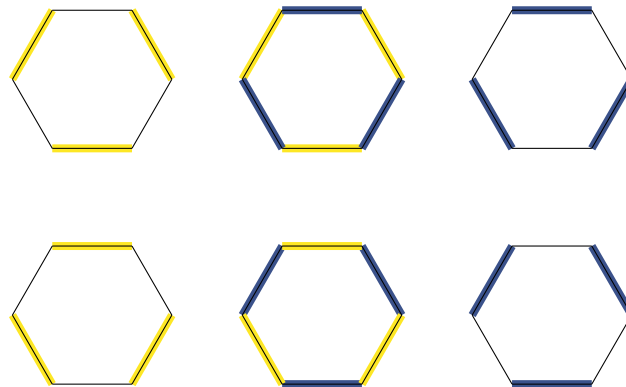


Figure 3: The two possible configurations of a single hexagon loop in the double dimer model.

The following definition of the $SL_2(\mathbb{C})$ double dimer model is due to Kenyon [4]:

Definition 4. Let $G = (V, E)$ be a bipartite graph with a *scalar weight* $w : E \rightarrow \mathbb{C}$ as well as an SL_2 *connection*: a map $\Gamma : E \rightarrow SL_2(\mathbb{C})$. Then the contribution of a double dimer configuration DD is defined to be

$$\left(\prod_{e \in m} w(e) \right) \times \prod_{\text{closed loops } L} \text{Tr} \left(\prod_{e \in L} \Gamma(e) \right).$$

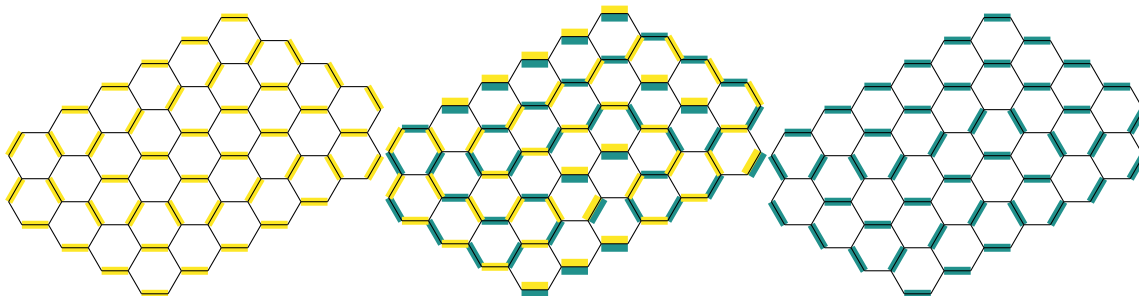


Figure 4: A double dimer configuration – an overlaying of the right and the left single dimer configurations, that consists of closed loops and doubled edges.

3 The squish map

3.1 Coordinates on the honeycomb grid

The honeycomb graph is the tiling of the plane by hexagons; the center of each hexagon is a point in the dual triangular lattice. A convenient way to give coordinates to the triangular lattice is to draw it on the plane normal to $(1, 1, 1) \in \mathbb{R}^3$. This plane has an orthonormal basis

$$\vec{x} = \frac{1}{\sqrt{2}}(-1, 1, 0) \quad \vec{y} = \frac{1}{\sqrt{6}}(-1, -1, 2)$$

which we won't really use, but we *will* orient our pictures according to it, and we will use words like up, left, bottom, horizontal, etc... in the conventional way with respect to this basis.

The set $\mathbb{Z}^3 \subseteq \mathbb{R}^3$ projects onto a copy of the triangular lattice in this plane. Draw the honeycomb graph G in such a way that the lattice points are the centers of the hexagons, then label each lattice point (and thus each hexagon) with any of the lattice points that project to its center. For instance, the hexagon at the origin has the labels $(0, 0, 0), \pm(1, 1, 1), \pm(2, 2, 2)$, etc. Call this particular embedding of the honeycomb graph H . In the \vec{x}, \vec{y} coordinates above, the edge common to the hexagon at $(0, 0, 0)$ and the hexagon at $(0, 0, 1)$ is horizontal.

There is also a second embedding of the honeycomb graph onto the plane which is of interest to us, which we shall glibly call $2H$, and it is obtained by projecting the double-sized lattice $(2\mathbb{Z}) \times (2\mathbb{Z}) \times (2\mathbb{Z})$ to a (bigger) triangular lattice in the plane, and taking the planar dual. The hexagons of $2H$ have even coordinates $(2i, 2j, 2k)$.

3.2 Degenerating H to $2H$

The squish map can be defined entirely combinatorially, but for visualization purposes it is extremely helpful to first define a continuous degeneration $H(t) : [0, 1] \rightarrow \mathbb{R}^2$ such that $H(0) = H$ and $H(1) = 2H$. This degeneration is shown in Figure 5 and first appeared in [8].

To define $H(t)$, write $H = (V, E)$, and let $H^{ev} = (V^{ev}, E^{ev})$ be the subgraph of H consisting of all hexagons whose centers have even coordinates; we say that H^{ev} are the “even hexagons”. Let $P = H \setminus H^{ev}$. Then the graph (V, P) is a disconnected union of “propellers” (or $K_{1,3}$ ’s or “claws”, depending on what dialect of graph theory you speak). Each propeller has a central vertex v and three leaves x, y, z . For $t \in [0, 1]$, define $x(t) = tv + (1 - t)x$, so that $x(t)$ is a point on the edge joining v to x . Define $y(t)$ and $z(t)$ similarly, and make the same definitions at each other propeller.

The graph embedding $H(t)$ is obtained by drawing the vertex a at position $a(t)$. For $0 \leq t < 1$, $H(t)$ is an embedding of H ; indeed, $H(t)$ is an explicit homotopy equivalence. At $t = 1$, however, we have $v = x = y = z$, and thus $H(1)$ degenerates to an embedding of $2H$.

3.3 The squish map

$H(t)$ defines a 2-to-1 map Sq from E^{ev} to the edges of $2H$: given an edge e of an even hexagon in H , find the corresponding edge in $H(0)$, and let $Sq(e)$ be the corresponding edge in $H(1)$.

Definition 5. The squish map $Sq : D(H) \rightarrow DD(2H)$ sends a dimer configuration m on H to a double dimer configuration $Sq(m)$ on $2H$ as follows: let $m^{ev} = m \cap E^{ev}$; then $Sq(m) = Sq(m^{ev})$.

Indeed, when drawing m on the graph $H(t)$ for $t < 1$, visualize what the squish map is doing: propeller edges of m get shorter and shorter, while the doubled edges get longer and closer together. This process is shown in Figure 5.

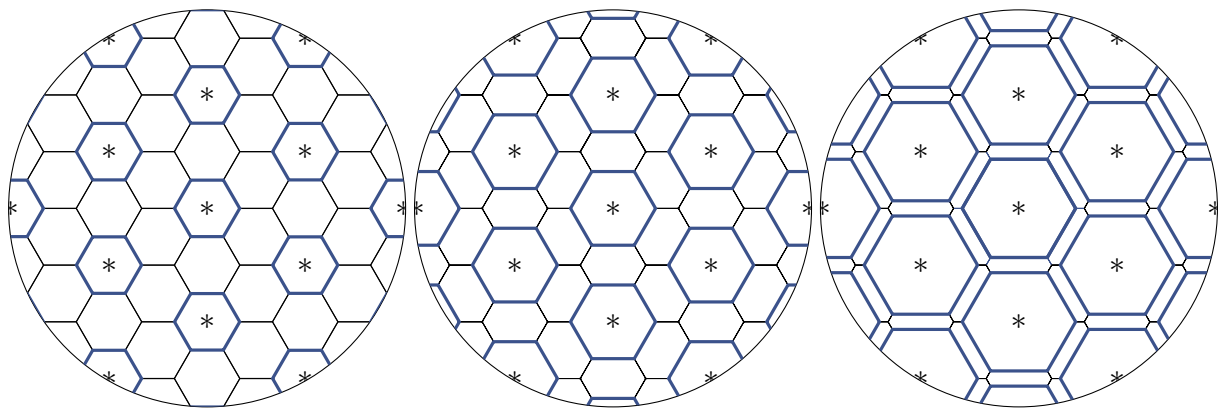


Figure 5: The degeneration $H(t)$.

We now relate the squish map to the $1 \times 1 \times 1$ and $2 \times 2 \times 2$ stacked cubes mentioned in the introduction. We can use the language of plane partitions to visualize a perfect matching in the single dimer model as a stack of cubes in the corner of a room, as in Figure 6. Then putting our matching through the squish map ‘downsamples’ the plane partition: we would use $1/8$ as many $2 \times 2 \times 2$ bricks instead. For example, to build a $2 \times 2 \times 4$ prism we can use 16 single cubes, or two larger $2 \times 2 \times 2$ cubes.

Theorem 6. Consider a single dimer configuration \mathcal{S} sent through the squish map resulting in a double dimer configuration \mathcal{D} made of loops and doubled edges. In each 2×2 section of the plane partition for \mathcal{S} , let $\pi_{\min} \begin{pmatrix} m & n \\ o & p \end{pmatrix} = \lfloor \frac{\min(m,n,o,p)}{2} \rfloor$, and $\pi_{\max} \begin{pmatrix} m & n \\ o & p \end{pmatrix} = \lceil \frac{\max(m,n,o,p)}{2} \rceil$.

These are the minimal and maximal plane partitions that, when overlayed, give us the double dimer configuration \mathcal{D} .

A few natural questions one might ask at this point would be: what single dimer configurations squish to a given double dimer configuration? In the language of building blocks, how can we tell which larger $2 \times 2 \times 2$ brick configurations overlay with one another to give us a particular double dimer model? How do we recover that information if we were only given the $1 \times 1 \times 1$ smaller blocks?

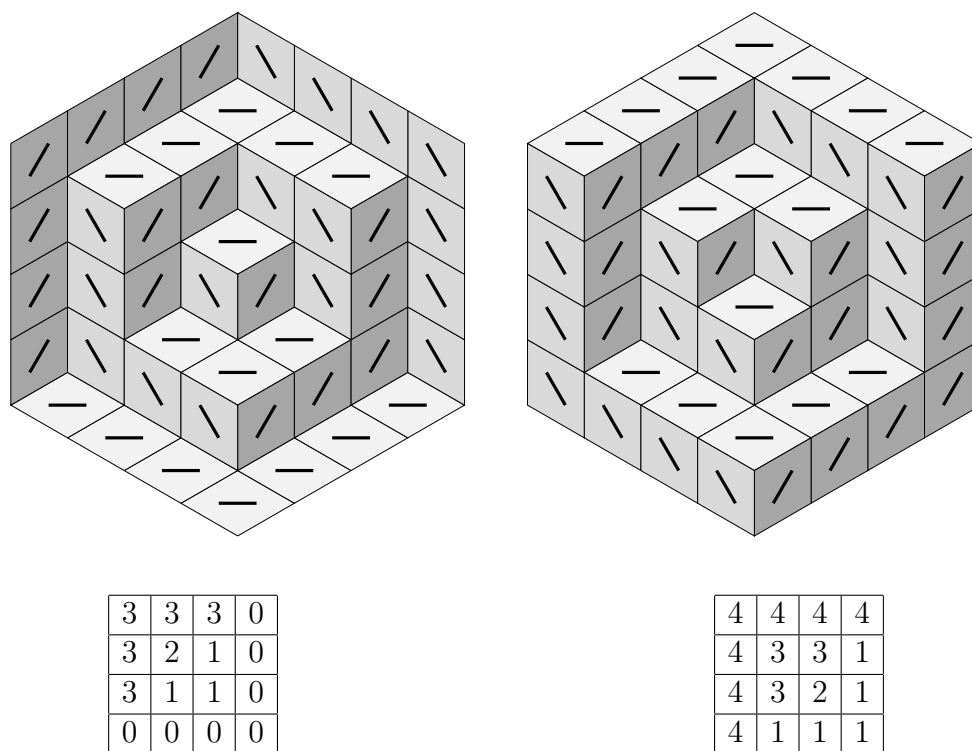


Figure 6: Two single dimer configurations shown as boxed plane partitions.

Example 7. Consider the two plane partitions given by Figure 6. Both of these configurations become a loop within a loop (as in Figure 7) when sent through the squish map. If we picture the plane partitions as boxes in a room, the minimal configuration that squishes to the loop within a loop is the diagram on the left of Figure 6, and the maximal one is the diagram on the right. (Note that these are just two out of a possible 23,364 configurations that squish to the same loop-within-a-loop. See Theorem 20.) To

determine which possible two overlaid $2 \times 2 \times 2$ single dimer configurations give the same double dimer configuration, we start by downsampling either plane partition.

For this example, we will work through the process on the minimum diagram and partition above, though the process will work on either partition (or any of the other 23,362 in between).

We want to round down to get the first single dimer configuration, π_{min} , and round up to get the second single dimer configuration, π_{max} . To do this rounding process, start by sectioning the plane partition into 2×2 grids.

3	3	3	0
3	2	1	0
3	1	1	0
0	0	0	0

Each 2×2 section will become one entry in the downsampled plane partition. When we round down, we want to count how many *complete* $2 \times 2 \times 2$ blocks exist in each section. (You can also think about this as removing smaller $1 \times 1 \times 1$ cubes one-at-a-time until you are only left with $2 \times 2 \times 2$ blocks.) Thus, we have $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ as the first single dimer configuration.

Now to find the second single dimer configuration, we want to round up to the nearest larger cube. (You can also think about this as adding smaller cubes until we fill the $2 \times 2 \times 2$ cube.) So the maximal configuration in its entirety becomes rewritten as $\begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix}$.

Now we have a pair of plane partitions that, when overlaid, become the same double dimer configuration we had as the result under the squish map. In Figure 7, the yellow (lighter colored) configuration corresponds to $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, and the blue (darker colored)

configuration corresponds to $\begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix}$.

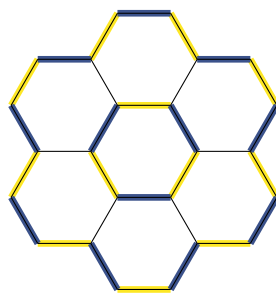


Figure 7: A loop within a loop in the double dimer model.

Example 8. For the plane partition

8	8	6	5
7	6	6	5
6	4	3	3
5	4	3	3
4	3	3	2
3	3	2	1
2	2	1	1
1	1	1	0

then

$$\pi_{min} = \begin{array}{|c|c|} \hline 3 & 2 \\ \hline 2 & 1 \\ \hline 1 & 0 \\ \hline 0 & 0 \\ \hline \end{array} \quad \text{and} \quad \pi_{max} = \begin{array}{|c|c|} \hline 4 & 3 \\ \hline 3 & 2 \\ \hline 2 & 2 \\ \hline 1 & 1 \\ \hline \end{array}$$

Remark 9. Consider a plane partition that could already have been made out of $2 \times 2 \times 2$ boxes. We can downsample in a straightforward manner. Each 2×2 region in the plane partition contains all the same even entry, so the entry in π_{min} would be half that number, as it would be for π_{max} . Since all four entries were the same, then we have no rounding to do, so $\pi_{min} = \pi_{max}$. Thus when we overlay the single dimer configurations corresponding to π_{min} and π_{max} , we get all doubled edges in the double dimer configuration.

Proof of Theorem 6. Consider a given plane partition π that may not break down nicely into $2 \times 2 \times 2$ cubes, where π has k boxes. Then $Sq(\pi) = \pi_{min} \sqcup \pi_{max}$ as double dimer configurations. We proceed by induction on k . The base case is $k = 0$, which is straightforward by the remarks above: $\pi, \pi_{min}, \pi_{max}$ and all the corresponding single or double dimer configurations are minimal.

Suppose that π has $k + 1$ boxes. Delete a box from π to create a π' that has k boxes. Then via the induction hypothesis we have that $Sq(\pi') = \pi'_{min} \sqcup \pi'_{max}$. If we add the $(k + 1)$ th box back in, we land in one of three cases.

In the first case, we have started a new $2 \times 2 \times 2$ cube by adding that cube in the $(2i, 2j, 2k)$ position. Then, when viewed as a plane partition, $\pi_{max}(i, j) = \pi'_{max}(i, j) + 1$, while the minimum configuration stays the same ($\pi_{min}(i, j) = \pi'_{min}(i, j)$). Under the squish map we get that $Sq(\pi) = Sq(\pi')$, except at the hexagon located at (i, j, k) , where the perfect matching around said hexagon changes from the left of Figure 8 to the right.

In the second case, we have added the cube into the last empty slot of a $2 \times 2 \times 2$ larger cube, so the new cube has gone into position $(2i + 1, 2j + 1, 2k + 1)$. Then the maximum plane partition remains the same, so $\pi_{max}(i, j) = \pi'_{max}(i, j)$, but $\pi_{min}(i, j) = \pi'_{min}(i, j) + 1$.

So under the squish map we have that $Sq(\pi) = Sq(\pi')$, except at (i, j, k) , where the matching around the hexagon changes from the right of Figure 8 to the left.

Finally, the last case occurs when we add a cube anywhere else; i.e. adding this smaller cube neither completes a $2 \times 2 \times 2$ box nor is the first cube in an otherwise empty $2 \times 2 \times 2$ box. Here we have that $\pi_{min} = \pi'_{min}$ and $\pi_{max} = \pi'_{max}$. In this case we have no change under the squish map, so $Sq(\pi) = Sq(\pi')$ exactly. \square

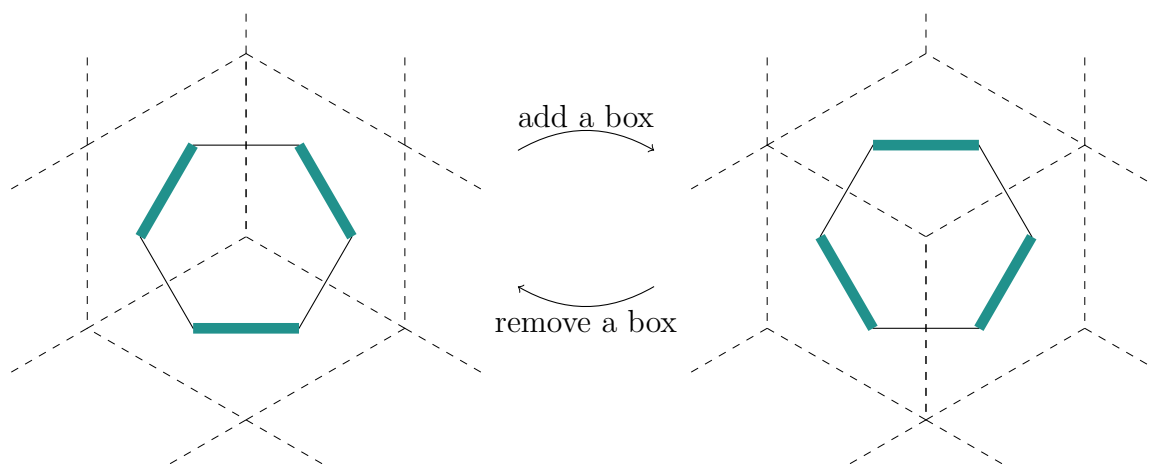


Figure 8: Adding or removing a box.

3.4 Transfer matrix approach

Now that we have the squish map defined on graphs and plane partitions, we want to be able to take a weight function in the single dimer model and push it through the squish map to give us an SL_2 connection and a scalar weight function in the double dimer model. To do this, we first consider a perfect matching on the single dimer model. Once the graph has been squished, we can consider walking along a path around a given loop in the now-double dimer model. Given a starting vertex, each path consists of a series of right and left turns at each new vertex encountered until we once again reach the starting vertex. These turns are given labels L and R for left and right assigned to each vertex in the path. We want to interpret L and R as 2×2 transfer matrices for keeping track of the loop's contribution to the dimer model. This was the strategy of [8], which used different matrices.

To determine what particular matrices L and R should be, we consider a walk along the path snippet given by Figure 9. We begin at the bottom edge and then turn left at the next vertex, so we could step onto either edge z or w . If x is an edge in the perfect matching, then z cannot also be a matched edge, so the only next step could be w . Similarly, if y is in the perfect matching, then a left turn onto the next matched edge could include either z or w .

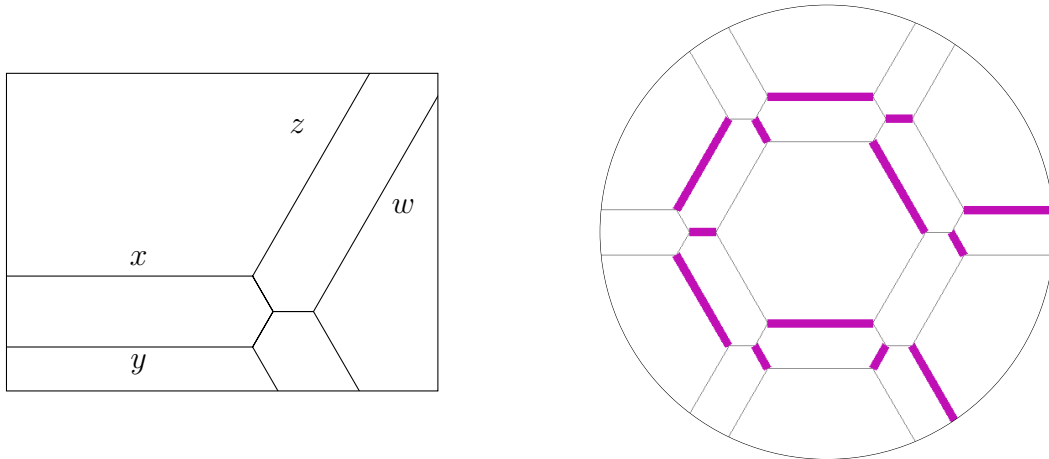


Figure 9: Left: the possible x, y, z , and w edge weights over a left-hand turn. Right: A perfect matching on a squished hexagon lattice.

We represent this with the vectors

$$\begin{bmatrix} x \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ y \end{bmatrix}$$

as our two possible starting locations. Then we need the first vector to map only to edge w , and the second vector to map to both edges z and w , so we get the following maps

$$\begin{bmatrix} x \\ 0 \end{bmatrix} \mapsto \begin{bmatrix} 0 \\ w \end{bmatrix} \quad \begin{bmatrix} 0 \\ y \end{bmatrix} \mapsto \begin{bmatrix} z \\ w \end{bmatrix}.$$

A similar scenario happens for R , so we then find the 2×2 matrix to make the above maps hold, getting that

$$L = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

Remark 10. Note that instead of defining L and R as above, we could also have defined L to be R^{-1} , and similar for R . So instead we would have had

$$L = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}.$$

Sometimes it may be more convenient to define L and R this way (with the negative signs), such as if we had specialized the edge weights to be ± 1 . We ultimately wanted to use the version with only positive entries, however, to (hopefully) make it more clear to the reader how things are working.

3.5 From transfer matrices to the SL_2 connection

We can now use the matrices L and R to compute the contribution of a closed loop under the squish map from the single to the double dimer model by taking the trace of the product. However, we're supposed to have 2×2 matrices, with determinant 1 associated to the *edges* of $H(i, j, k)$, not the vertices. We use the following process to include the left and right turn information in the edges of the graph as matrix weights.

To begin with, we associate general 2×2 edge-weight matrices A, B , and C to be placed on each horizontal, north-east, and north-west matched edge, respectively, as in Figure 10.

$$A = \begin{bmatrix} a & 0 \\ 0 & a^{-1} \end{bmatrix}, \quad B = \begin{bmatrix} b^{-1} & 0 \\ 0 & b \end{bmatrix}, \quad C = \begin{bmatrix} c & 0 \\ 0 & c^{-1} \end{bmatrix}$$

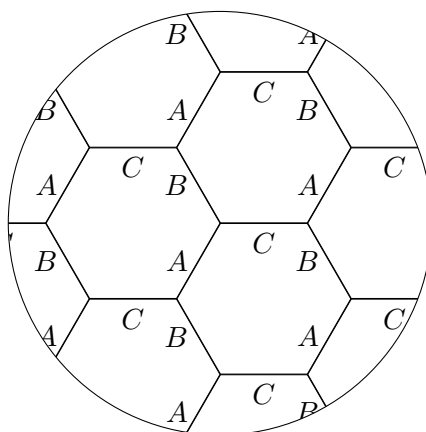


Figure 10: Matrix weights A, B , and C on the single dimer model.

To handle the process of moving L and R from the vertices to the edges, we need some way to include the information for turns in an unknown path. Then we want to come up with new matrices α, β , and γ that include the information from A, B , and C , but that also encode the information in the turns. For example, the move $A \rightarrow B$ is always a left turn, and the move $A \rightarrow C$ is always a right turn. So we somehow want the new $\beta\alpha$ to include the same information as BLA and $\gamma\alpha$ to encode the information from CRA (reading paths from right-to-left).

Definition 11. Let $J = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, and $i = \sqrt{-1}$.

Then α, β , and γ are defined as follows:

$$\alpha := iAJ = \begin{bmatrix} ia & 0 \\ 0 & \frac{1}{ia} \end{bmatrix}$$

$$\beta := iRBLJ = \begin{bmatrix} ib & -ib - \frac{i}{b} \\ 0 & \frac{1}{ib} \end{bmatrix}$$

$$\gamma := -iLCRJ = \begin{bmatrix} \frac{1}{ic} & 0 \\ -ic - \frac{i}{c} & ic \end{bmatrix}.$$

Theorem 12. *The single dimer model on the $2x \times 2y \times 2z$ hexagon lattice with A , B , and C weights (as in Figure 10) gives rise to the same partition function as that on the double dimer model on the $x \times y \times z$ hexagon graph with scalar weight of 1 everywhere and connection given by α , β , and γ (as in Figure 11).*

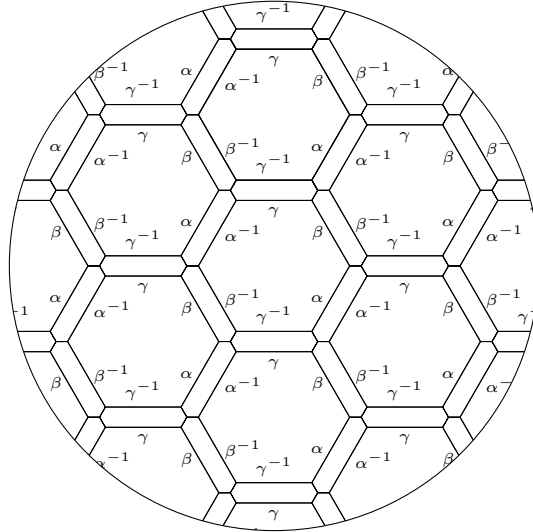


Figure 11: The α , β , and γ connection on the squished hexagon lattice.

Proof. From the examples above, Let's consider the right turn $A \rightarrow C$. In our matrices, we write this as $\gamma\alpha$, which is $(-iLCRJ)(iAJ) = LCRA$, which does indeed include the information CRA , as we desired. There are 12 total ways to step from one edge to another, whose equivalent products are given below.

Left Turns	Right Turns
$\beta\alpha = -RBLA$	$\alpha\beta = -JARBLJ$
$\gamma\beta = LCLBLJ$	$\beta\gamma = -RBRCRJ$
$\alpha^{-1}\gamma = -JA^{-1}LCRJ$	$\gamma\alpha^{-1} = -LCRA^{-1}$
$\beta^{-1}\alpha^{-1} = -RB^{-1}LA^{-1}$	$\alpha^{-1}\beta^{-1} = -JA^{-1}RB^{-1}LJ$
$\gamma^{-1}\beta^{-1} = LC^{-1}LB^{-1}LJ$	$\beta^{-1}\gamma^{-1} = RB^{-1}RC^{-1}RJ$
$\alpha\gamma^{-1} = -JALC^{-1}RJ$	$\gamma^{-1}\alpha = -LC^{-1}RA$

Assume we have a valid path of length n with matrix string $\chi_n\chi_{n-1}\dots\chi_1$. Then this string is equivalent to one of the form

$$X_n M_n T_{n-1} M_{n-1} \cdots T_2 M_2 T_1 M_1 X_1,$$

where each $M \in \{A, B, C, A^{-1}, B^{-1}, C^{-1}\}$ is one of the original edge matrices, $T_i \in \{R, L\}$ is a turning matrix L or R , and the X_i -terms on either end are the extras above used for bookkeeping (the LJ , RJ , L , R , or J ‘bookkeeping’ terms on the left or right of each full string). If we were to take another step along this path, then that $(n + 1)$ th step would involve left-multiplying our string by χ_{n+1} , so we now have $\chi_{n+1}\chi_n \cdots \chi_1 = (\chi_{n+1}\chi_n)\chi_{n-1} \cdots \chi_1$. But the $\chi_{n+1}\chi_n$ is a valid construction of two edge-weight matrices with a left or right turn in the middle (possibly with an X_{n+1} or X_n bookkeeping term), which means it is one of the 12 turns explicitly computed above. So $\chi_{n+1}\chi_n \cdots \chi_1$ gives us a string that is equivalent to using A , B , and C with left- and right-turn matrices. \square

Example 13. Consider the path around a single hexagon with these new weights. If we begin at the bottom edge and then travel counterclockwise around the loop, then we perform matrix multiplication (reading right to left) in the order we reach the edges, getting $\gamma^{-1}\beta^{-1}\alpha^{-1}\gamma\beta\alpha$. This product gives us the total connection for the path.

Now if we compute this matrix out for the left-turn version, we get

$$\begin{aligned} & \gamma^{-1}\beta^{-1}\alpha^{-1}\gamma\beta\alpha \\ &= (iLC^{-1}RJ)(-iRB^{-1}LJ)(iA^{-1}J)(-iLCRJ)(iRBLJ)(iAJ) \\ &= i^6LC^{-1}RJR B^{-1}LJA^{-1}JLCRJ RBLJAJ \\ &= -LC^{-1}LB^{-1}LA^{-1}iLCLBLA \\ &= \begin{bmatrix} a^2b^2c^2 + a^2b^2 + a^2c^2 + a^2 + 1 & \frac{(a^2b^2c^2 + a^2b^2 + a^2 + 1)(b^2 + 1)}{a^2b^2} \\ \frac{(a^2b^2c^2 + a^2c^2 + a^2 + 1)(c^2 + 1)}{c^2} & \frac{a^2b^4c^4 + a^2b^4c^2 + a^2b^2c^4 + 3a^2b^2c^2 + a^2b^2 + a^2c^2 + b^2c^2 + a^2 + b^2 + c^2 + 1}{a^2b^2c^2} \end{bmatrix} \end{aligned}$$

and the trace of this matrix is $a^2b^2c^2 + a^2b^2 + a^2c^2 + b^2c^2 + a^2 + b^2 + c^2 + \frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2} + \frac{1}{a^2b^2} + \frac{1}{a^2c^2} + \frac{1}{b^2c^2} + \frac{1}{a^2b^2c^2} + 4$ which has 18 terms (including repeated terms).

So, in particular, we know that there are only 20 $2 \times 2 \times 2$ boxed plane partitions: the minimal one, the maximal one, and 18 others. Then using the trace of our matrix, we have accounted for all 18 terms that correspond to the 18 perfect matchings which get squished by the squish map to a single loop. These are the correct weights for plane partitions with 2-periodic weights.

4 Generating functions and specializations

Note that the previous sections have used periodic edge weights and connections. Here we generalize to cover several natural weight functions for plane partition enumeration.

4.1 Arbitrary weights

We first consider an *arbitrary* nonzero weight function $w : G \rightarrow \mathbb{C}^*$. Note that every vertex v of G is a part of some propeller: either v is the center vertex of a propeller, or not (in which case, only one of the edges incident to v is part of the propeller).

We modify the weight function w by performing a so-called *gauge transformation*: for each vertex v which is not the center of a propeller; suppose that the edge e connects v to

the center of the propeller. Divide the weights of all of v 's incident edges by $w(e)$. This operation changes the dimer model partition function only by an overall constant, which we can ignore at least in the case where G is a finite subgraph of the honeycomb graph. Assume without loss of generality that this has been done.

Then, given two edges e_1, e_2 which get sent to e under the squish map, let $w(e) = \sqrt{w(e_1)w(e_2)}$, the geometric mean of the weights of e_1 and e_2 . Define a new weighting on G by

$$\tilde{w}(e_1) = \frac{w(e_1)}{w(e)}, \quad \tilde{w}(e_2) = \frac{w(e_2)}{w(e)}.$$

This weighting \tilde{w} has the property that $\tilde{w}(e_1) = \tilde{w}(e_2)^{-1}$, so we can push it through the squish map as before.

4.2 Periodic weights on plane partitions

One has to be careful with computing the weight of a perfect matching m on any infinite graph, such as H . There are infinitely many edges in such a graph, so one would need to consider issues of convergence before multiplying all of the weights together to find the weight of a perfect matching. There is then a separate convergence issue when trying to sum over m .

However, our main interest is in plane partitions, which are in bijection with certain perfect matchings on H . Plane partitions come with a natural generating function, computed by MacMahon, as stated in Section 1.1. We now define

$$M(a, q) = \prod_{i \geq 1} \left(\frac{1}{1 - aq^i} \right)^i$$

for weighted plane partitions [6]. Indeed, there is a particular weighting which is uniquely relevant on the entire honeycomb lattice: the $\mathbb{Z}_2 \times \mathbb{Z}_2$ periodic weighting.

We assign the weights *to the lattice points* as follows:

$$w(i, j, k) = \begin{cases} q & \text{if } i - j \equiv 0, i - k \equiv 0 \pmod{2}, \\ r & \text{if } i - j \equiv 1, i - k \equiv 0 \pmod{2}, \\ s & \text{if } i - j \equiv 0, i - k \equiv 1 \pmod{2}, \\ t & \text{if } i - j \equiv 1, i - k \equiv 1 \pmod{2}. \end{cases} \quad (1)$$

There are a variety of edge weight functions on the dimer model which correspond to this; one which is well behaved with respect to the squish map is (partially) shown in Figure 12. We previously defined the weight function on a finite portion of the honeycomb graph (as in the monochromatic weighting on Figure 2). To extend this weighting to the entire plane, set $p^3 = q$ and include constants k_1, k_2 , and k_3 .

Choose each of k_1, k_2 , and k_3 so that the desired region of the plane has the appropriate edge weights. (Generally we choose this in such a way that the edge weights near the center of the region in question are all of low degree.) This is comparable to choosing

the weight “1” on a particular place in a given column of horizontal edges from Example 3. The result will be that each ‘strip’ of hexagons (in each vertical, northeast/southwest, or northwest/southeast diagonal, as in figure 12) will have a weight of 1 in the desired location, with increasing and decreasing powers traveling either direction away from the center.

We can decompose this weighting into two pieces, as shown on the bottom of Figure 12. The first piece becomes the SL_2 connection, as described above and seen in the lower right of the figure. The second piece becomes the scalar weights, which is shown in the lower left. For further details on various weight functions of the honeycomb graph, see [9].

Theorem 14. *Using the weight function given in Figure 2 and a monodromy given by placing α on every northeast/southwest edge, β on every northwest/southeast edge, and γ on every horizontal (east/west) edge, then the squish map is measure preserving.*

Corollary 15. *Using α , β , and γ as described in Theorem 14, we can find the generating function for a given $m \times n \times o$ boxed double dimer plane partition.*

Remark 16. As seen in [4], we can compute the probability of a given edge being present in the single dimer model by taking the determinant of a certain Kasteleyn matrix. When we use this procedure in conjunction with the squish map, we can now determine the probability that a given edge will appear as a doubled edge in the double dimer model. If said doubled edge is present, then its preimage under the squish map would have two edges mapping to the doubled edge, so the overall probability would be the product of the two individual probabilities that each edge would be present in the single dimer model.

Define $Q := qrst$, and $\widetilde{M}(x, y) := M(x, y)M(x^{-1}, y)$. The generating function for plane partitions in which q, r, s, t mark the boxes in as in equation 1 was computed in [9] (where it was denoted $Z_{\mathbb{Z}_2 \times \mathbb{Z}_2}$); it is

$$Z_Q = M(1, Q)^4 \frac{\widetilde{M}(rs, Q)\widetilde{M}(st, Q)\widetilde{M}(tr, Q)}{\widetilde{M}(-r, Q)\widetilde{M}(-s, Q)\widetilde{M}(-t, Q)\widetilde{M}(-rst, Q)}. \quad (2)$$

Under the specialization $r = s = t = q$ (and hence $Q = q^4$) we recover MacMahon’s generating function for plane partitions by a slightly delicate manipulation of formal power series. In [8] it is observed that under the specialization $r = s = t = -1$, the above generating function specializes to $M(1, -q)$, and this latter statement is proven using the squish map (without the SL_2 double dimer model).

Corollary 17 (to Theorem 14). *The partition function for the SL_2 double dimer model is Z_Q from equation 2.*

5 Examples

We now work out two examples in which we squish boxed $2 \times 2 \times 2$ and $2 \times 2 \times 4$ plane partitions, with boxes weighted by $w(i, j, k)$. These do not have nice closed-form

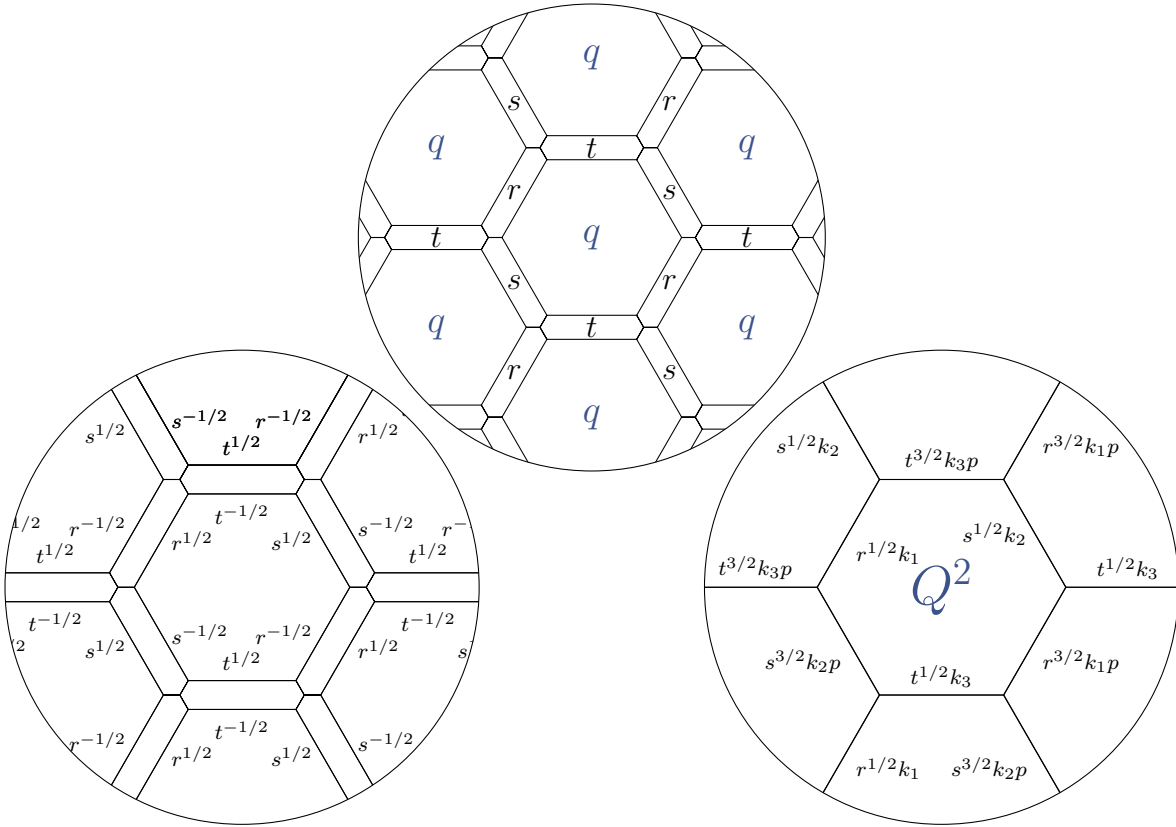


Figure 12: Top: 2-periodic weights for the single dimer model. Bottom: SL_2 connection and scalar weight for the corresponding double dimer model.

generating functions akin to MacMahon's generating function, but we can nonetheless evaluate them with our techniques. We also investigate the specializations of our formula at primitive roots of unity, recovering and extending the results of [9], as well as a new conjecture.

Example 18 ($2 \times 2 \times 2$). Consider the path around a single hexagon that is the result of a particular single dimer configuration sent through the squish map, or a $2 \times 2 \times 2$ boxed plane partition in the double dimer model. The two configurations on $H_{2,2,2}$ that do not give rise to a loop under the squish map are pictured in Figure 13. The right graphic has a total weight of 1, and the right graphic has a total weight of Q^2 . Then the remaining 18 partitions are represented by the polynomial $1 + q + qr + qs + qt + qrs + qrt + qst + 4qrst + qr^2st + qrs^2t + qrst^2 + qr^2s^2t + qr^2st^2 + qrs^2t^2 + qr^2s^2t^2 + q^2r^2s^2t^2$.

We want to draw a comparison between Z_Q and the matrix model under the squish map, so consider the trace of the product of terms corresponding to the single hexagon loop, $\frac{\eta}{a^2b^2c^2}$, where $\eta := a^4b^4c^4 + a^4b^4c^2 + a^4b^2c^4 + a^2b^4c^4 + a^4b^2c^2 + a^2b^4c^2 + a^2b^2c^4 + 4a^2b^2c^2 + a^2b^2 + a^2c^2 + b^2c^2 + a^2 + b^2 + c^2 + 1$. Note that this is only the connection in the $SL_2(\mathbb{C})$ double dimer model, not also the weights. So let the weight on every horizontal

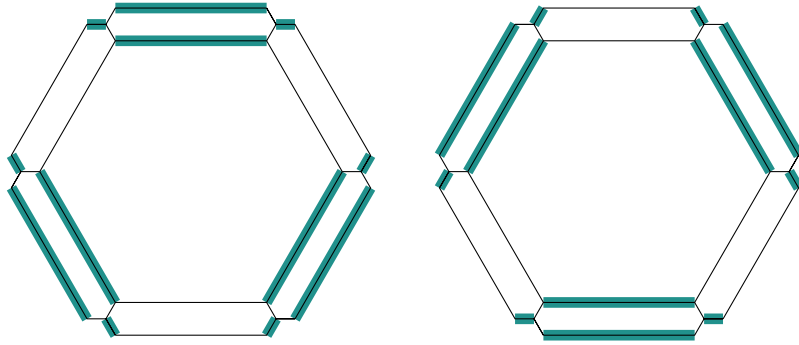


Figure 13: The only two configurations of a $2 \times 2 \times 2$ single dimer model that do not squish to a single hexagon loop.

edge be

$$Q^{(\text{height of that edge})} = (qa^2b^2c^2)^{(\text{height of that edge})},$$

as in Figure 2. Then the total contribution (the connection and the weight) of that single hexagon is

$$qa^2b^2c^2 \left(\frac{\eta}{a^2b^2c^2} \right) = q \cdot \eta.$$

Now consider a mapping from Z_Q to the matrices, where $a^2 = r$, $b^2 = s$, and $c^2 = t$. Then $Q =qrst = qa^2b^2c^2$. Then making the replacements in the statement above, we get

$$q(r^2s^2t^2 + r^2s^2t + r^2st^2 + rs^2t^2 + r^2st + rs^2t + rst^2 + 4rst + rs + st + rt + r + s + t + 1).$$

Then this matches what we got for Z_Q when we include the 1 and Q^2 terms from the two non-loop configurations.

Example 19 ($2 \times 2 \times 4$). For another example, we consider the case with two hexagons, or a $2 \times 2 \times 4$ boxed plane partition. Our generating function is

$$\begin{aligned} GF_{2 \times 2 \times 4} = & q^4r^4s^4t^4 + q^3r^4s^4t^4 + q^3r^4s^4t^3 + q^3r^4s^3t^4 + q^3r^3s^4t^4 + q^3r^4s^3t^3 + q^3r^3s^4t^3 \\ & + q^2r^4s^4t^3 + q^3r^3s^3t^4 + q^2r^4s^4t^2 + 4q^3r^3s^3t^3 + q^2r^4s^3t^3 + q^2r^3s^4t^3 + q^3r^3s^3t^2 \\ & + q^2r^4s^3t^2 + q^2r^3s^4t^2 + q^3r^3s^2t^3 + q^3r^2s^3t^3 + 3q^2r^3s^3t^3 + q^3r^3s^2t^2 + q^3r^2s^3t^2 \\ & + 4q^2r^3s^3t^2 + q^3r^2s^2t^3 + 2q^2r^3s^2t^3 + 2q^2r^2s^3t^3 + q^2r^3s^3t + q^3r^2s^2t^2 + 3q^2r^3s^2t^2 \\ & + 3q^2r^2s^3t^2 + 3q^2r^2s^2t^3 + q^2r^3s^2t + q^2r^2s^3t + 9q^2r^2s^2t^2 + q^2r^2st^3 + q^2rs^2t^3 \\ & + 3q^2r^2s^2t + 3q^2r^2st^2 + 3q^2rs^2t^2 + qr^2s^2t^2 + q^2rst^3 + 2q^2r^2st + 2q^2rs^2t + qr^2s^2t \\ & + 4q^2rst^2 + qr^2st^2 + qrs^2t^2 + 3q^2rst + qr^2st + qrs^2t + q^2rt^2 + q^2st^2 + qrst^2 \\ & + q^2rt + q^2st + 4qrst + q^2t^2 + qrs + q^2t + qrt + qst + qr + qs + qt + q + 1. \end{aligned}$$

This generating function, however, contains all 105 of the terms given by a $2 \times 2 \times 4$ four-colored boxed plane partition. To compare with the matrix version, we need to be careful which terms squish to which configurations. See Figure 14.

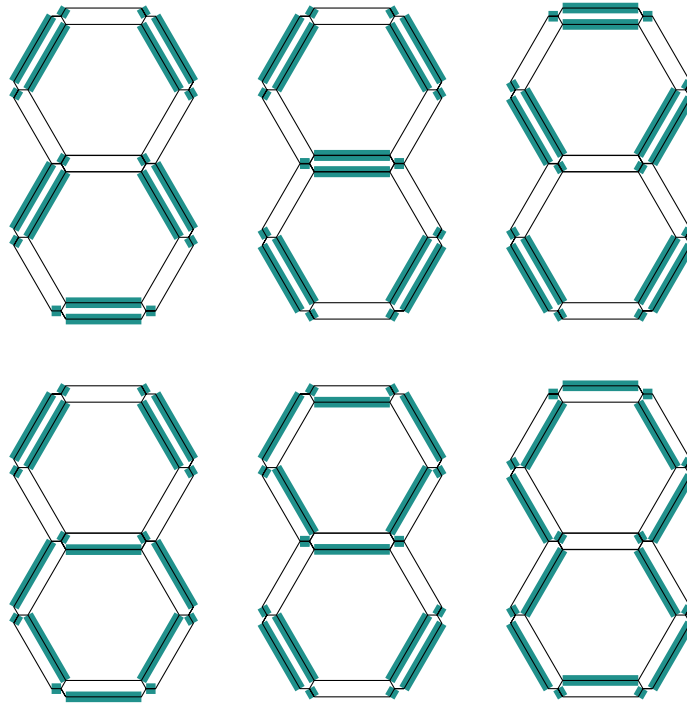


Figure 14: A representation of of each possible result of pushing the $2 \times 2 \times 4$ single dimer configuration through the squish map.

Note that the top row of configurations in Figure 14 has no loops (only doubled edges), so there is no contribution from the $SL_2(\mathbb{C})$ connection. So we get a contribution only from the weight, and we get 1 for the configuration on the left, Q^2 for the configuration in the middle, and Q^4 for the configuration on the right. Since there is only one configuration in the single dimer model that squishes to each of these they must each account for only one term each in the generating function $GF_{2 \times 2 \times 4}$, namely 1, $q^2 r^2 s^2 t^2$, and $q^4 r^4 s^4 t^4$.

Now we examine the second line of Figure 14. From the $2 \times 2 \times 2$ (Example 18) case, we see that we get 18 terms in the generating function for a single loop. So each of the first two figures in the second line correspond to 18 terms in the generating function, but to determine which terms we need to include the Q -weights. So the lower left figure has two horizontal edges, the lower getting a weight of 1 and the upper getting a weight of Q . So this diagram corresponds with the terms Q (the single hexagon generating function seen previously) $= Q (r^2 s^2 t^2 + r^2 s^2 t + r^2 s t^2 + r s^2 t^2 + r^2 s t + r s^2 t + r s t^2 + 4 r s t + r s + s t + r t + r + s + t + 1)$. Then the lower-middle diagram gets a weight of $Q^2 \cdot Q = Q^3$, so its generating function is $Q^3 (r^2 s^2 t^2 + r^2 s^2 t + r^2 s t^2 + r s^2 t^2 + r^2 s t + r s^2 t + r s t^2 + 4 r s t + r s + s t + r t + r + s + t + 1)$. Then we have the 105 monomials from $GF_{2 \times 2 \times 4}$, and subtract off the three monomials corresponding to the no-loop configurations, and the $2 \cdot 18$ monomials just described corresponding to single hexagon loop configurations, and we are left with the following 66 terms: $(r^4 s^4 t^2 + r^4 s^4 t + r^4 s^3 t^2 + r^3 s^4 t^2 + r^4 s^3 t + r^3 s^4 t + 3r^3 s^3 t^2 + 4r^3 s^3 t + 2r^3 s^2 t^2 + 2r^2 s^3 t^2 + r^3 s^3 + 3r^3 s^2 t + 3r^2 s^3 t + 3r^2 s^2 t^2 + r^3 s^2 + r^2 s^3 + 8r^2 s^2 t + r^2 s t^2 + r s^2 t^2 + 3r^2 s^2 +$

$3r^2st + 3rs^2t + rst^2 + 2r^2s + 2rs^2 + 4rst + 3rs + rt + st + r + s + t + 1)q^2t$, which corresponds to a loop around two hexagons, as in the lower right of Figure 14.

To compare with the corresponding contribution of the SL_2 double dimer model, we first take the matrix product of a monodromy around both hexagons, then take the trace and multiply by the weight $(qa^2b^2c^2)^2$.

$$(qa^2b^2c^2)^2 \cdot \left(\frac{1}{a^4b^4c^2} \cdot [a^8b^8c^4 + a^8b^8c^2 + a^8b^6c^4 + a^6b^8c^4 + a^8b^6c^2 + a^6b^8c^2 + 3a^6b^6c^4 + 4a^6b^6c^2 + 2a^6b^4c^4 + 2a^4b^6c^4 + a^6b^6 + 3a^6b^4c^2 + 3a^4b^6c^2 + 3a^4b^4c^4 + a^6b^4 + a^4b^6 + 8a^4b^4c^2 + a^4b^2c^4 + a^2b^4c^4 + 3a^4b^4 + 3a^4b^2c^2 + 3a^2b^4c^2 + a^2b^2c^4 + 2a^4b^2 + 2a^2b^4 + 4a^2b^2c^2 + 3a^2b^2 + a^2c^2 + b^2c^2 + a^2 + b^2 + c^2 + 1] \right)$$

Then make the replacements using $a^2 = r$, $b^2 = s$, and $c^2 = t$. The resulting polynomial is $(r^4s^4t^2 + r^4s^4t + r^4s^3t^2 + r^3s^4t^2 + r^4s^3t + r^3s^4t + 3r^3s^3t^2 + 4r^3s^3t + 2r^3s^2t^2 + 2r^2s^3t^2 + r^3s^3 + 3r^3s^2t + 3r^2s^3t + 3r^2s^2t^2 + r^3s^2 + r^2s^3 + 8r^2s^2t + r^2st^2 + rs^2t^2 + 3r^2s^2 + 3r^2st + 3rs^2t + rst^2 + 2r^2s + 2rs^2 + 4rst + 3rs + rt + st + r + s + t + 1)q^2t$, which matches exactly with the 66 terms from naively enumerating the plane partitions with the same weights.

5.1 Specializations

In an attempt to find a simple application of these techniques to plane partition enumeration, we have noticed the following curious phenomenon. Consider the specialization $a = b = c = \omega$, a primitive n th root of unity for various n , and let L be a loop which appears in the SL_2 double dimer model, after performing the squish map.

Theorem 20. *If $n = 1$ or $n = 2$, then the contribution of L is the number of double dimer configurations that contribute to that loop.*

Example 21. Consider the shape from Figure 15 (one of the six snake tiles). The path around this tile is $\gamma^{-1}(\beta^{-1}\alpha^{-1}\gamma\alpha^{-1})^2\gamma(\beta\alpha\gamma^{-1}\alpha)^2$, and if we calculate out that matrix product and then specialize to $a = b = c = \omega$, a first or second root of unity, we get the matrix

$$\begin{bmatrix} 1393 & 576 \\ 2208 & 913 \end{bmatrix}.$$

The trace of this matrix is 2306, corresponding with the 2306 possible double dimer configurations that squish to that particular snake tile.

Proof. The first and second roots of unity case essentially acts as a reduction from the α , β , and γ version of the matrices to the original L and R vertex turn matrices. If we calculate the path around a shape as *turns* instead of using edge connections α , β , and γ , then we get the same resulting trace of the matrix product. In example 21, we would have the turn sequence (reading right-to-left since it corresponds to a path) $LLLRLLRLLLLRLLRL$, and if we use the originally-defined L and R matrices, we would get $\begin{bmatrix} 337 & 1152 \\ 576 & 1969 \end{bmatrix}$, which has a trace of 2306, as in the example. \square

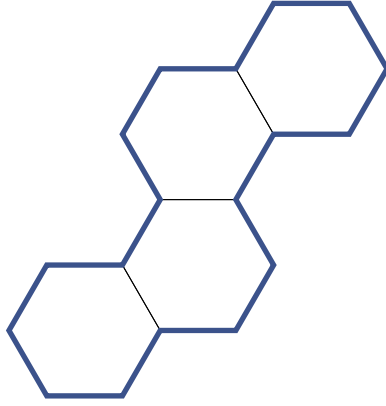


Figure 15: $\gamma^{-1}(\beta^{-1}\alpha^{-1}\gamma\alpha^{-1})^2\gamma(\beta\alpha\gamma^{-1}\alpha)^2$.

Theorem 22. *If $n = 4$ then the contribution of L is 2.*

Proof. Note that this is the same contribution as if the connection were trivial (the matrix associated to every edge of G being identity matrix). When $n = 4$, then each matrix α , β , and γ is equivalent to $-I$, where I is the 2×2 identity matrix. Then since any possible loop in the graph must have even length, the resulting product around a loop is I . \square

Theorem 23. *If $n = 8$, then the result of multiplying the connection around L is $\pm \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, where the sign tracks the parity of the number of hexagons contained in L . Note that this is still true regardless of our choice of basepoint.*

Proof. The trace of a loop around a single hexagon when specialized to eighth roots of unity is $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = -1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = -I$. Then assume that the path around a region containing n hexagons is $(-1)^n \cdot I$.

Consider a region R with n hexagons, as in our assumption above. Attach one additional hexagon on the boundary of R in a way that does not affect the simply-connectedness of R , and call this new region R' . Choose your basepoint to be the spot right after the added hexagon. Then the boundary of R is the same until the new hexagon. Upon reaching the new hexagon, take the old path of R , then loop all the way around the new hexagon (still in the counterclockwise direction). Note that this will traverse the most recent one to five edges from R , but now in the opposite direction. Finally, the boundary of R' also includes the boundary of the new hexagon. So our whole path is that of $R \cdot \{\text{path around a single hexagon}\}$, which is $(-1)^n \cdot I \cdot (-1) \cdot I = (-1)^{n+1} \cdot I$. \square

For the next conjecture we need some terminology from Conway-Lagarias [1] about tiling regions in H with tiles made of unions of hexagons. A *bone* is the union of three collinear adjacent hexagons in H ; a *stone* is the union of three hexagons in H which all share a common vertex. A *signed tiling of L* is a collection of tiles, each with a weight of $+1$ or -1 , covering L in such a way that the total contribution at each hexagon inside

L is 1, and the total contribution of each hexagon outside L is zero. We also define one more tile, the *snake*, which is a union of four hexagons in an “S” shape (see Figure 15).

Conjecture 24. If $n = 3$ or $n = 6$, then the contribution of L is 0 unless there exists a signed tiling of L by stones, bones and snakes – in which case, the contribution is $(-1)^s \cdot I$, with s being the number of stones used in the signed tiling.

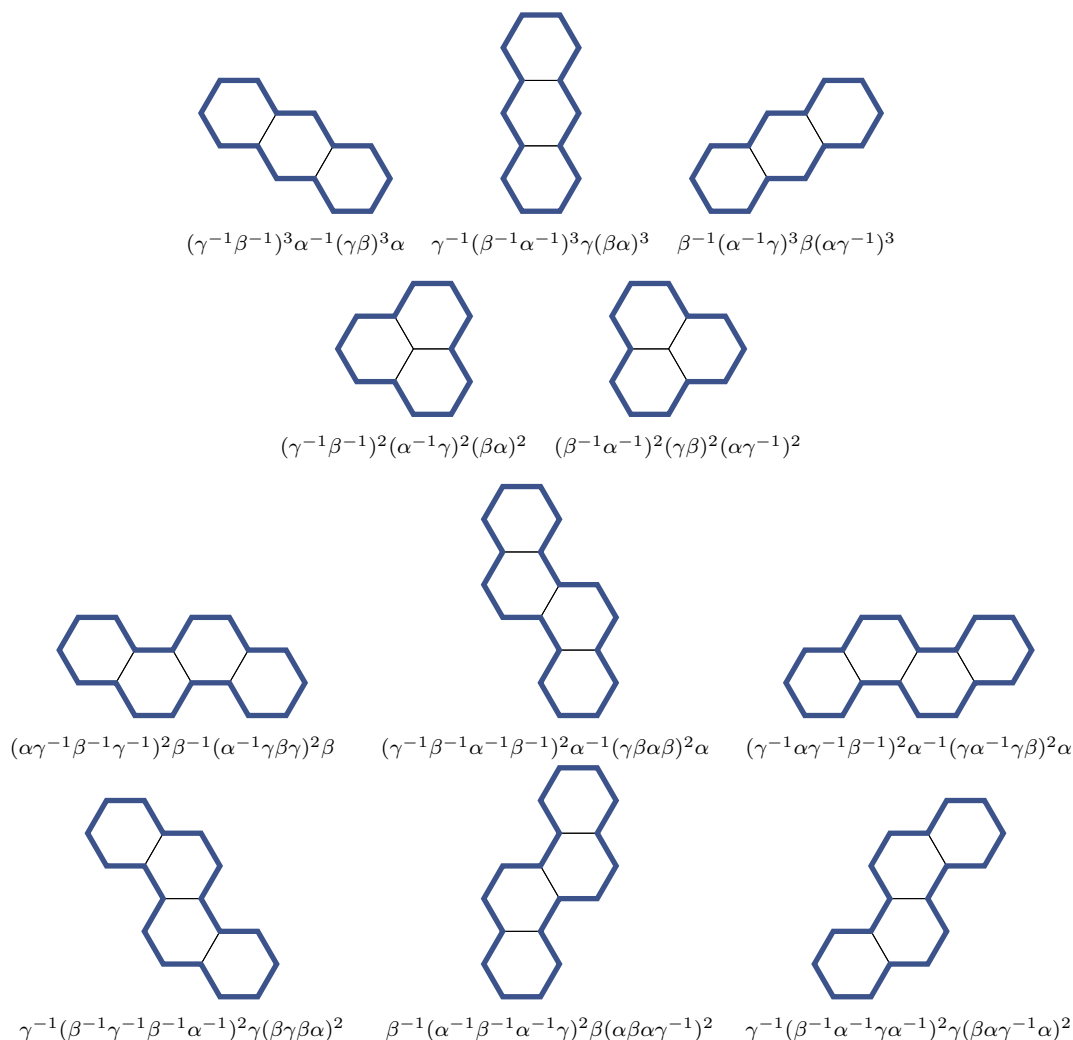


Figure 16: All of the tiles we use for ω a third or sixth root of unity. The top row is all three orientation of bones, the second row both orientation of stones, and the bottom two rows all six orientations of the snake tile.

When $n = 3$ or $n = 6$, then the contribution of a loop around a bone or snake is I . So if we inductively add tiles to a region (as we did in the proof of Theorem 23 for individual hexagons), we get $\pm I$, where the parity depends only on the number of stone tiles used in the tiling. So to prove the conjecture, one needs only show that the monodromy of all non-signed-tilable L is 0.

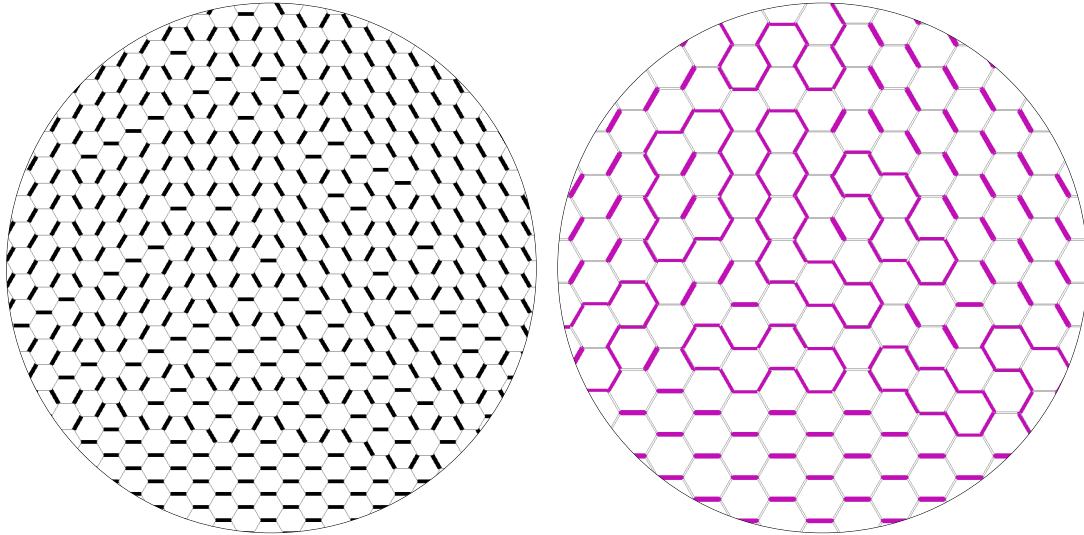


Figure 17: The single dimer model before and after the squish map showing various closed loops, including two snakes, a bone, and some other loops with positive contributions under Conjecture 24.

We expect that after this specialization our SL_2 connection is strongly related to the character of one of the Conway-Lagarias tiling groups, so the proof will involve computing this character, as well as a map from said tiling group to a subgroup generated by the matrices α , β , and γ . Note that a contribution of 0 means that a double dimer configuration will not contribute at all to the partition function, so this should give an interesting generating function for “good pairs” of plane partitions.

6 Concluding remarks

Here we explain the process of finding α , β , and γ . Perhaps our discovery process is of interest to the reader that has made it this far.

Let M (for move) be the matrix $M = L^{-1}R$. Then $M^{-1} = R^{-1}L$. Our first attempt to create matrices α , β , and γ involved simply conjugating A, B, C with different powers of M , noting that $M^3 = -I$. Based on empirical studies of the shortest possible loop, six left turns around a single hexagon, this looked close to the correct definition. We defined $\alpha_1 := A$, $\beta_1 := M^{-1}BM$, and $\gamma_1 := M^{-2}BM^2$. These matrices worked for our main goal – terms in the trace of the resulting matrix count how many perfect matchings squish to a particular loop, with the correct weights. However, that polynomial had alternating signs depending on the total degree of each term.

Our next step was to correct for the signs by replacing a with ai , b with bi , and c with ci in the original A, B and C matrices. Let A', B' , and C' be the corresponding A, B , and C matrices after making the above replacements. We wanted all of the signs to be positive because the matrix product of a path around a loop should give rise to a

probability measure on the graph, as in [4]. After including the is we had a trace that has all positive values. Then, $\alpha_2 := A'$, $\beta_2 := M^{-1}B'M$, and $\gamma_2 := M^{-2}C'M^2$.

We had one final problem, however: we needed to accomplish the specialization $a \mapsto ai$, etc, using linear algebra. We did this using the matrix J as defined above. We can now rewrite $A' = iJA$, $B' = -iJB$, and $C' = iJC$. Finally, to correct for the signs we mentioned, we conjugate each α_2 , β_2 , and γ_2 with J . So (finally) we have that $\alpha_3 := JA'J$, $\beta_3 := JM^{-1}B'MJ$, and $\gamma_3 := JM^{-2}C'M^2J$. After rewriting and simplifying, we arrive at the final definition of α , β , and γ as stated in Definition 11.

Acknowledgements

We would like to thank Richard Kenyon and Sunil Chhita for helpful conversations. Leigh Foster received support from NSF grant DMS-2039316.

References

- [1] J. H. Conway and J. C. Lagarias. Tiling with polyominoes and combinatorial group theory. *J. Combin. Theory Ser. A*, 53(2):183–208, 1990.
- [2] Ira Gessel and Gérard Viennot. Binomial determinants, paths, and hook length formulae. *Adv. in Math.*, 58(3):300–321, 1985.
- [3] PW Kasteleyn. The statistics of dimers on a lattice. *Physica*, 27:1209–1225, 1961.
- [4] Richard Kenyon. Conformal invariance of loops in the double-dimer model. *Comm. Math. Phys.*, 326(2):477–497, 2014.
- [5] Bernt Lindström. On the vector representations of induced matroids. *Bull. London Math. Soc.*, 5:85–90, 1973.
- [6] Percy A. MacMahon. *Combinatory analysis. Vol. I, II (bound in one volume)*. Dover Phoenix Editions. Dover Publications Inc., Mineola, NY, 2004.
- [7] H. N. V. Temperley and Michael E. Fisher. Dimer problem in statistical mechanics—an exact result. *Philos. Mag. (8)*, 6:1061–1063, 1961.
- [8] Benjamin Young. Squishing dimers on the hexagon lattice. *Electron. J. Combin.*, 16(1):#R86, 20, 2009.
- [9] Benjamin Young. Generating functions for colored 3D Young diagrams and the Donaldson-Thomas invariants of orbifolds. *Duke Math. J.*, 152(1):115–153, 2010.