An introduction to the algebraic Bethe ansatz

James Plunkett
Department of Mathematics and Statistics,
The University of Melbourne,
Victoria 3010, Australia.

SUPERVISOR: DR OMAR FODA

Honours Thesis, November 2006.

Abstract

The algebraic Bethe ansatz is one of the main approaches to solving integrable models in statistical mechanics. This thesis is a detailed introduction to aspects of this technique.

In chapter 1, following Takhtajan[1], we fully explain all preliminaries that are required by this technique in the context of the 8-vertex model, including the solution of the Yang-Baxter equations in terms of Boltzmann weights that are parameterized by elliptic function.

In chapter 2, also following Takhtajan, we restrict our attention to the special case of the 6-vertex model, with Boltzmann weights parameterized by hyperbolic trigonometric functions, formulate the model in detail on a finite lattice with periodic boundary conditions, and show how the algebraic Bethe ansatz can be used to characterize the eigenvectors and eigenvalues, in the homogeneous case (all vertical rapidities are equal, and all horizontal rapidities are also equal).

In chapter 3, following Bogoliubov *et al*[2], we consider the 6-vertex model in inhomogeneous case (all rapidities are now independenti variables) but with domain wall boundary conditions.

Using the algebraic Bethe ansatz, and Izergin's result for the partition function, we calculate boundary 1-point functions of the six-vertex model on a finite lattice and represent them in determinant form.

[1] Takhatajan, L. J. Introduction to Agebraic Bethe Ansatz, *Lecture Notes in Physics, Exactly solvable problems in condensed matter and relativistic field theory*

[2] Boundary correlation functions of the six-vertex model *J. Phys. A: Math. Gen. 35, 5525-5541 (2002)*

Contents

Acknowledgments

I would very much like to thank my supervisor, Dr Omar Foda. Thank you for introducing me to this fascinating field of statistical mechanics. Despite being the busiest person I have ever met, you always managed to be encouraging and accommodating whenever I had issues.

I would also like to thank Alex Caradoc, Omar's PhD student. You probably would have finished by now if I wasn't harassing you all the time.

Lastly, I would like to thank the other Honours students for making this the most enjoyable of my seven years at Melbourne University by far.

Chapter 1

Monodromy matrix and the Quantum R-matrix

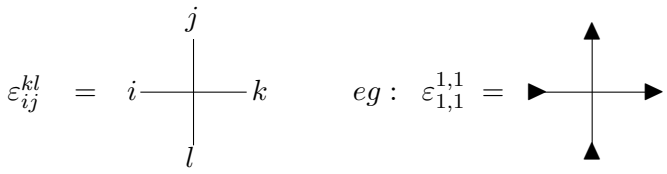
1.1 Vertex Models

In many applications of statistical mechanics particles are fixed or localized, which means that kinetic energy can essentially be neglected. Instead, a material's magnetic properties are determined by the angular momentum of the atoms' electrons, also known as 'spin'. [1] We can represent these fixed models grahically. Accordingly, this paper will use pictures in many proofs and to demonstrate ideas. It is therefore important to have an understanding of what the pictures represent.

Vertices

The energy of an atom or a particle is often represented by a vertex. The energy of these will depend on their spins, which we represent by 4 directed arrows, one on each edge. Denoting an up or right arrow as $+1$ and a down or left arrow as -1, we can represent the energy of an atom:

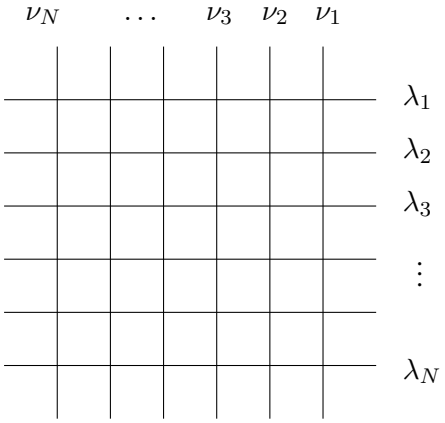
[1] Baxter, Rodney. J, *Exactly Solved Models in Statistical Mechanics,*



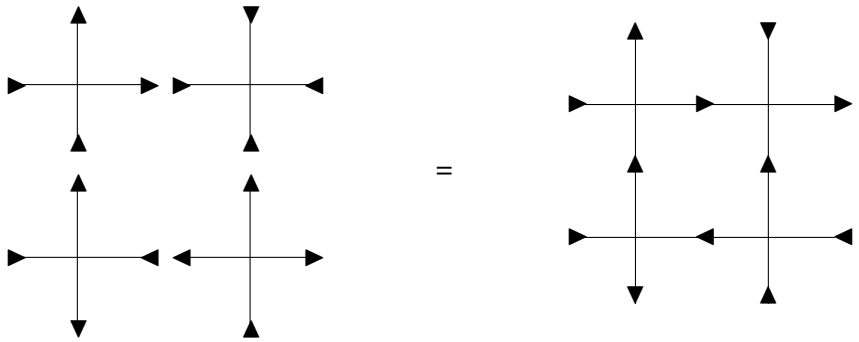
As there are 4 edges, and each edge can adopt two possible states, there are $2^4 = 16$ possible configurations / energies.

The Lattice

A lattice is similar to a vertex, but with more than one horizontal or vertical line, and represents a group of atoms or particles bonded together. When arrows are placed on the bonds, the lattice represents the total energy of that particular configuration of vertices, where each vertex on the lattice represents the energy of an atom. The total energy of the lattice is given by the sum of the energies of all the vertices. A 2-Dimensional rectangular lattice would look like:

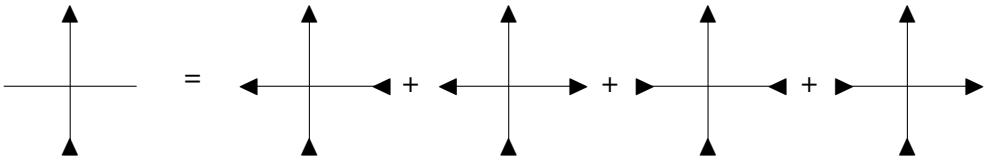


Where the labels ν_κ and λ_α can be seen, for now, as 'co-ordinates'. For the atoms to bond together, they must share a similar bond/arrow. For example:



Blank bonds and Combinations

As we often deal with many different configurations of arrows, rather than explicitly write them all out, we represent a sum over all configurations with blank bonds. Each blank bond is a sum over a right and left arrow (or an up and down arrow). For example:



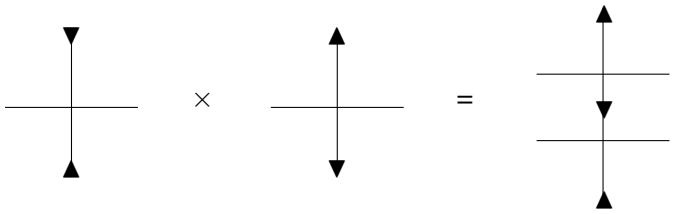
Clearly, for N blank bonds, there will be 2^N configurations.

Mutiplication of Vertices / Lattices

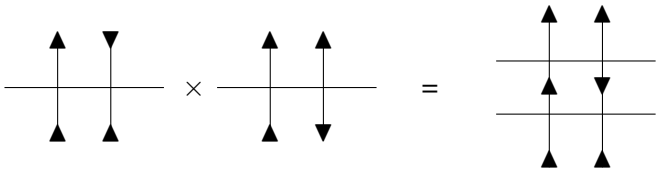
There are two possibilities for multiplication: where the vertical arrows specified, and where the horizontal arrows specified. The idea is to connect similar arrows and join them together.

Vertical Multiplication:
Multiplying vertices whose vertical bonds have been specified requires us to place

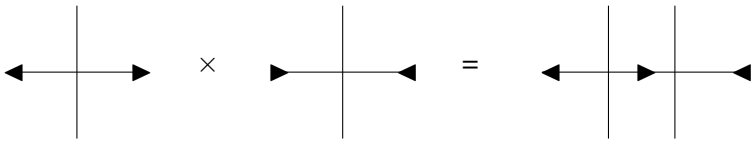
the right vertex on top of the left vertex, where the arrows on the top of the left vertex are the same as the arrows on the bottom of the right vertex (so the first vertex in the product lies on the bottom of the resulting lattice). For example



(the blank horizontal bonds still represent a sum over all configurations.) This can be extended to lattices, provided the arrows on the top of the left lattice align with the arrows at the bottom of the right lattice. For example:



Horizontal Multiplication:
Multiplication of horizontal bonds is much easier: we simply join them together, just as they join together in the lattice: For example:



If a product of vertices cannot be joined by vertical or horizontal multiplication, we must leave them as two separate sublattices. As a general rule, vertices/lattices are not commutative under multiplication.

1.2 The Partition Function

Let us consider an $M \times N$ rectangular lattice with periodic boundary conditions. We can think of the edges in the lattice as forming a circle, where the arrows at the top must point in the same direction as the corresponding arrows on the other side of the lattice (similarly for the right and left). The $M \times N$ rectangular lattice, can be thought of as a two dimensional taurus. If each vertex configuration represents a different energy ε_j, $(j = 1 \ldots 16)$, then each lattice configuration has a total energy of:

$$E = \sum_{j=1}^{16} N_j \varepsilon_j$$

Where N_j is the number of vertices with energy ε_j

The *partition function*, Z, is an important quantity in statistical mechanics. It encodes various properties of a system in thermodynamic equilibrium. Here, it can be represented:

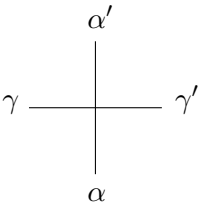
$$Z = \sum e^{-\beta E}. \tag{1.2.1}$$

where we are summing over all possible configurations of the lattice. Given each bond can have two arrow configurations, and there $M \times N$ bonds, the partition function is the sum of the energies (E) of $2^{M \times N}$ terms. (Note that $\beta = 1/kT$, where T is the absolute temperature of the system in Kelvin $(273K = 0^oC)$ and $k = 1.3805 \times 10^{-23}$ is Boltzman's constant.)

1.3 The Transfer Matrix

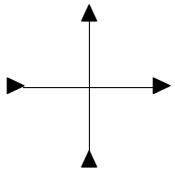
We can simplify Equation 1.2.1 by considering the *Botlzmann weight* of each vertex configuration, $v_j = e^{-\beta \varepsilon_j}$, and arranging them in a matrix $\{L_\alpha^{\alpha'}(\gamma, \gamma')\}_n$, where each entry of L_n represents the Boltzmann weight of a particular arrow configuration, and the subscript n represents the horizontal position (ν_n) that we want L_n to represent. Specify each entry of L_n as:

$L_\alpha^{\alpha'}(\gamma, \gamma') \quad =$



and, as we did before, assign spin variable $+1$ to the arrows pointing up and to the right and -1 to arrows down and to the left, so that:

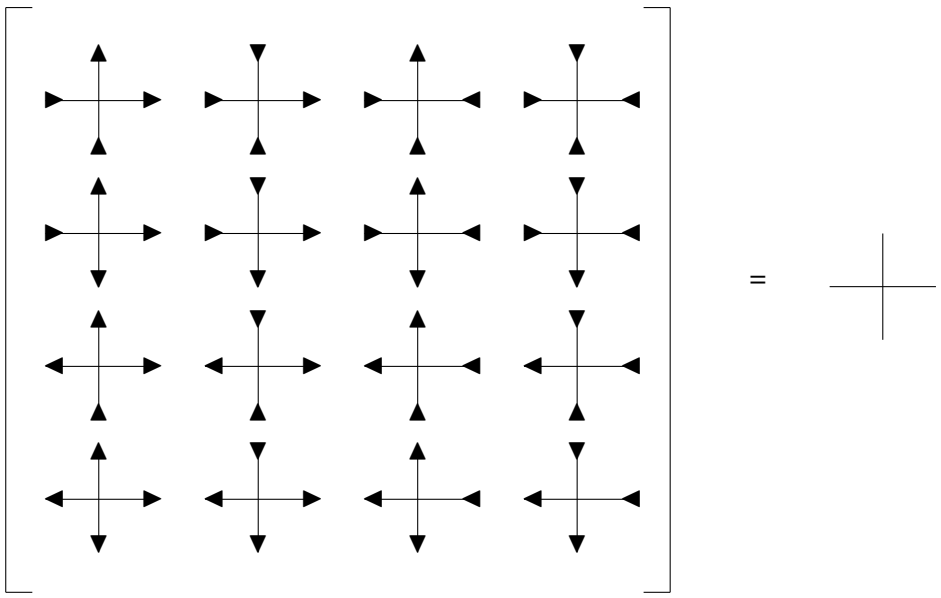
$L_1^1(1,1) \quad = e^{-\beta\varepsilon_{1,1}^{1,1}} \; =$



Then matrix L_n is:

$$\begin{bmatrix} L_1^1(+1,+1) & L_{-1}^1(+1,+1) & L_1^{-1}(+1,+1) & L_{-1}^{-1}(+1,+1) \\ L_1^1(-1,+1) & L_{-1}^1(-1,+1) & L_1^{-1}(-1,+1) & L_{-1}^{-1}(-1,+1) \\ L_1^1(+1,-1) & L_{-1}^1(+1,-1) & L_1^{-1}(+1,-1) & L_{-1}^{-1}(+1,-1) \\ L_1^1(-1,-1) & L_{-1}^1(-1,-1) & L_1^{-1}(-1,-1) & L_{-1}^{-1}(-1,-1) \end{bmatrix}_n$$

We can represent L_n graphically as:

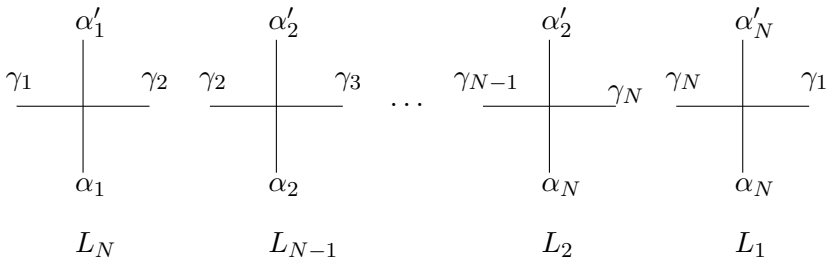


(1.3.1)

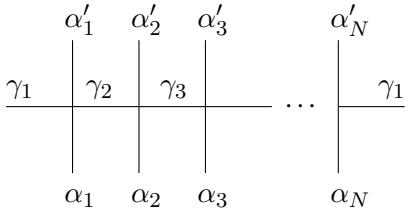
L_n is an operator that acts in the space $h_n \approx \mathbb{C}^2$, where h_n represents the n-th 'column' of the lattice. This is explained in further detail in Section 2.1. For simplicity, we will consider the *homogeneous* case where all L_n are equivalent but act in different spaces. The inhomogenous case is considered in in Chapter 3. Introducing $\{\alpha\} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ and $\{\alpha'\} = \{\alpha'_1, \alpha'_2, \ldots, \alpha'_N\}$, we can define matrix T_m, with elements:

$$T_{\{\alpha\},\{\alpha'\}} = \sum_{\gamma_1} \ldots \sum_{\gamma_N} L^{\alpha'_1}_{\alpha_1}(\gamma_1, \gamma_2) L^{\alpha'_2}_{\alpha_2}(\gamma_2, \gamma_3) \ldots L^{\alpha'_N}_{\alpha_N}(\gamma_N, \gamma_1) \tag{1.3.2}$$

Where m represents the λ_m that T corresponds to. As we are working in the homogeneous case we are also assuming that all rows are equivalent: $T_m \approx T$. We we can represent T graphically as:

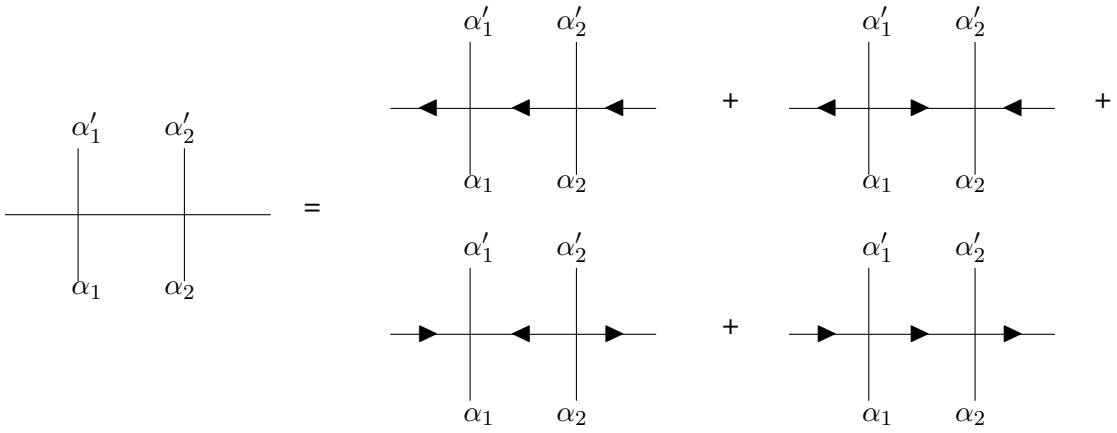


Which we can represent as one lattice, that is equivalent to a row of the lattice at λ_m.



It is important to note the periodicity of T: the last horizontal arrow (γ_1) is the same as the first, ensuring that the $N \times 1$ lattice is periodic in the horizontal direction.

Each entry of matrix T corresponds to a set of fixed $\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ and also $\{\alpha'_1, \alpha'_2, \ldots, \alpha'_N\}$, but a sum over all possible periodic configurations of γ_j. The entry represents the Boltzmann weight of that $1 \times N$ lattice configuration. T can be thought of as one‘ row’ from the original $M \times N$ lattice. All ‘rows’ are equivalent in the homogeneous case, as $T_m \approx T$. As each of the N blank edges can have a spin of ± 1, matrix T will be $2^N \times 2^N$. For example, if $N = 2$, Matrix T would be:



Specifying the values of $\{\alpha_1, \alpha_2\}, \{\alpha'_1, \alpha'_2\}$ would give is the different entries of

matrix T.

Now that we have defined T, we can express equation 1.2.1 as:

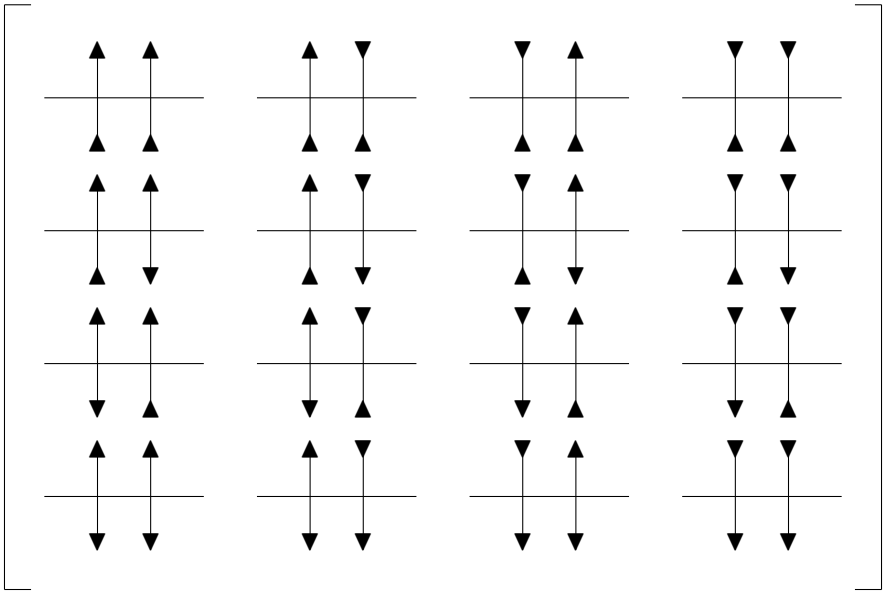
$$Z = Tr(T^M) \tag{1.3.3}$$

Where Tr is the matrix trace.

Proof: Given that N is arbitrary, we will consider the case $N = 2$ for simplicity. The proof can easily be generalized for larger values of N. The Transfer matrix, T, for $N = 2$ is:

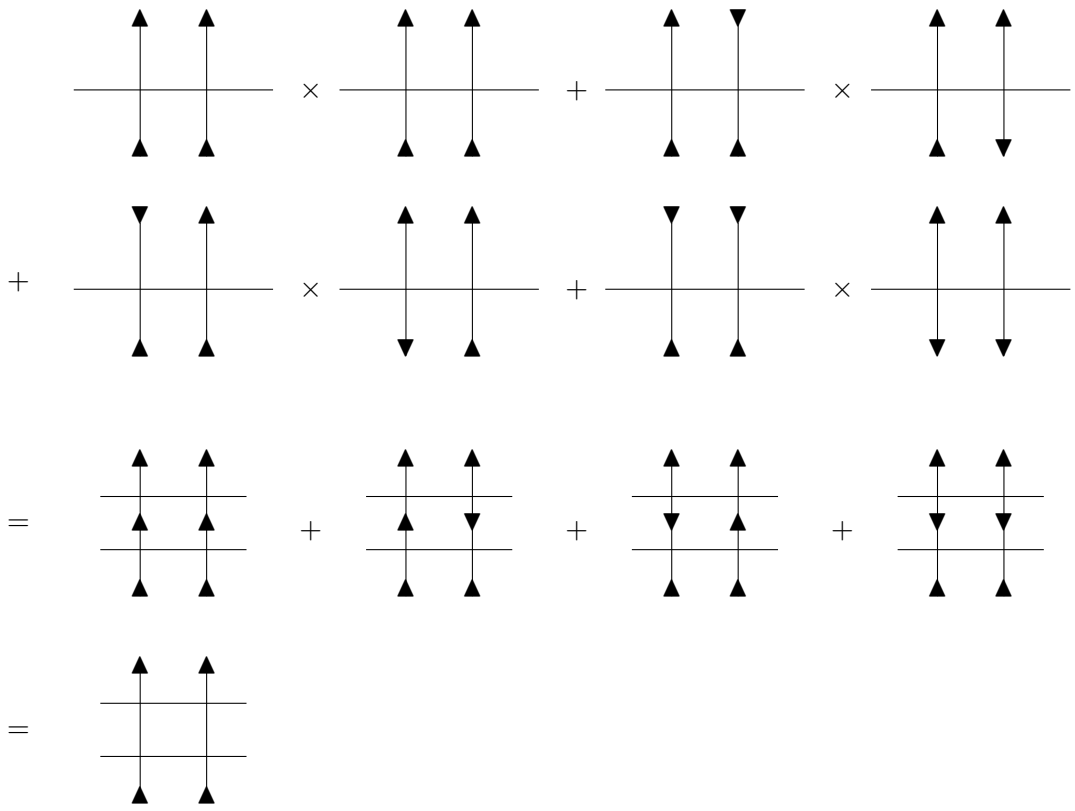
$$\begin{bmatrix} T_{\{+1,+1\},\{+1,+1\}} & T_{\{+1,+1\},\{+1,-1\}} & T_{\{+1,+1\},\{-1,+1\}} & T_{\{+1,+1\},\{-1,-1\}} \\ T_{\{+1,-1\},\{+1,+1\}} & T_{\{+1,-1\},\{+1,-1\}} & T_{\{+1,-1\},\{-1,+1\}} & T_{\{+1,-1\},\{-1,-1\}} \\ T_{\{-1,+1\},\{+1,+1\}} & T_{\{-1,+1\},\{+1,-1\}} & T_{\{-1,+1\},\{-1,+1\}} & T_{\{-1,+1\},\{-1,-1\}} \\ T_{\{-1,-1\},\{+1,+1\}} & T_{\{-1,-1\},\{+1,-1\}} & T_{\{-1,-1\},\{-1,+1\}} & T_{\{-1,-1\},\{-1,-1\}} \end{bmatrix} \tag{1.3.4}$$

Which can be represented graphically as:

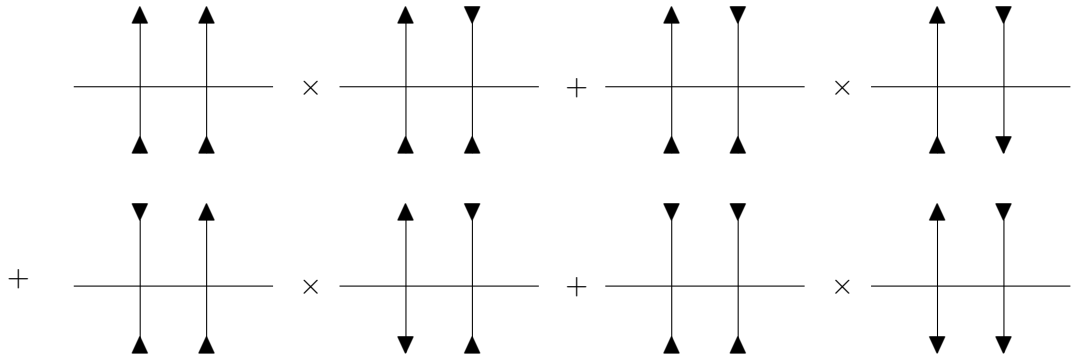


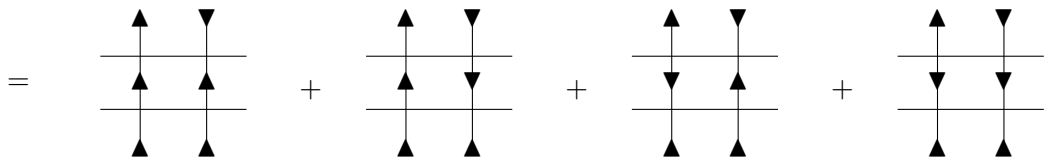
(Note: given the periodicity requirement of T, we are really only summing over periodic configurations of the horizontal bonds.) We eventually want T^M, so we

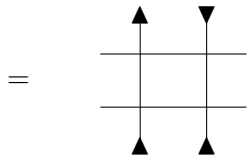
first need T^2. If we let $T_{\alpha,\beta}$ represent the α-th row and β-th column of T, then $T^2_{1,1}$, is:



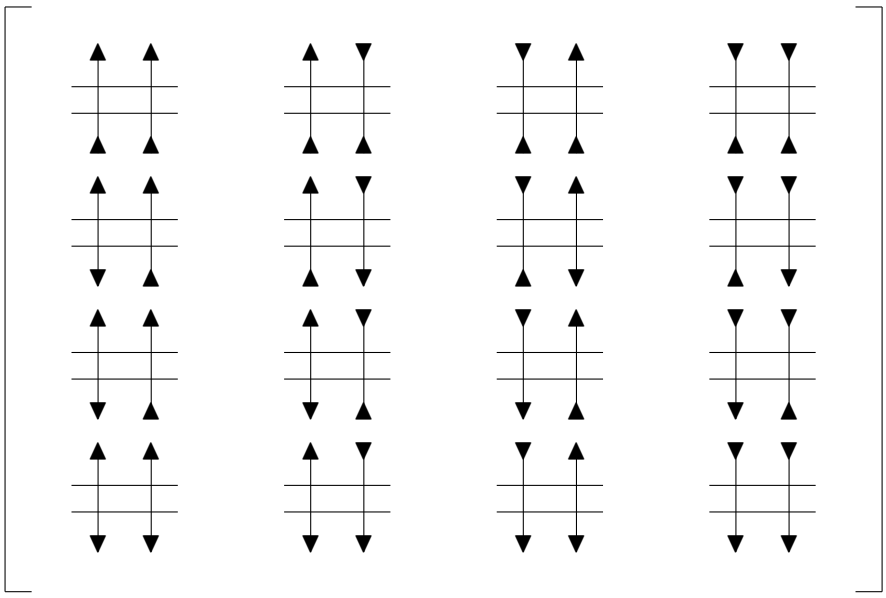
We can find the other entries of T^2 the same way. $T^2_{1,2}$ is:





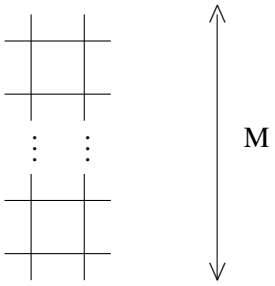


Doing this for all entries of T^2 gives:

$T^2 =$ 

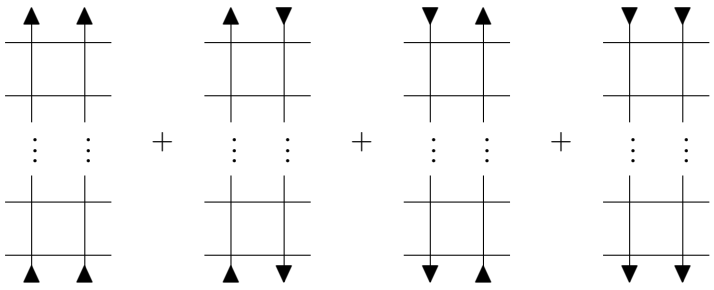
Notice that all the arrows are configured identically to the original T matrix, only each entry is a 2×2 lattice rather than a 2×1 lattice. Repeating this process M times will give us a matrix with $2 \times M$ lattices as entries and the configuration of the arrows will be identical to T and T^2. Thus, matrix T^M can be represented:

T^M $=$



The periodic entries of T^M will occur along the diagonal (as in T^2), which is simply the trace of T^M:

Trace(T^M) $=$



Which is precisely what we wanted: the partition function, Z, for $N = 2$. This is just the Boltmann weights of all periodic configurations of the $2 \times M$ lattice. The same process can be easily generalized to cases of $N > 2$

■

Clearly, for computational purposes, the above is not an efficient process of finding T^M for large M or N. It would be much simpler if we could diagonalize T. The way to do this is to introduce repidities in T which would allow us to diagonalize T using the Yang-Baxter equation, as it requires non-trivial dependance of Boltzman weights on rapidity variables.

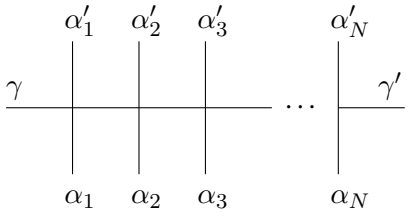
1.4 The Monodromy Matrix M

Suppose we have two sets of Boltzman weights, v_j and v'_j, and the corresponding transfer matrices T and T'. We would like to know when T and T' com-

mute:

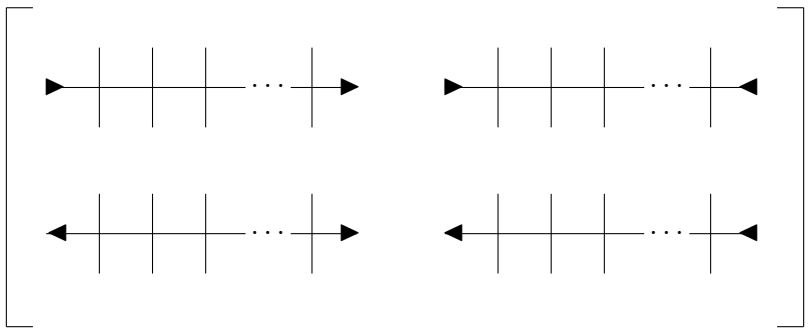
$$[T, T'] = 0 \tag{1.4.1}$$

To answer this question we need to introduce a 2×2 matrix M, as an operator on quantum space, with elements $M_{\gamma,\gamma'}$:



We can see that this is precisely the T matrix without the periodicity requirement. M can be represented graphically as:

M =



For convenience we will represent M as:

$$M = \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} \tag{1.4.2}$$

Given that only the diagonal entries of M are periodic, we can see that the transfer martix, T, is simply the trace of M.

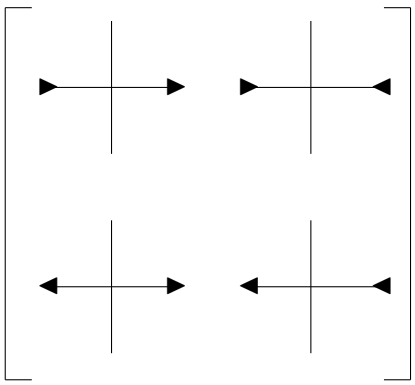
$$T = Tr(M) = \mathcal{A} + \mathcal{D}$$

If we think of h_n as a horizontal line in the n^{th} 'column' of the lattice, a product of all N columns will give a 'row' of the lattice. As T and M represent 'rows' of the lattice and therefore 'act' over all horizontal spaces:

$$\mathcal{V}_N = \prod_{n=1}^{N} \otimes h_n \quad , \quad h_n \approx \mathbb{C}^2$$

Where the $\mathbb{C}$ means that the Boltzman weights may take complex values. (What 'acts in its own space' means is discussed in Section 2.1.)

Recall that we represent the the Boltzmann weights of all 16 configurations in matrix (1.3.1). We can represent that 4×4 L_n matrix as a 2×2 matrix:

$L_n \ =$ 

(1.4.3)

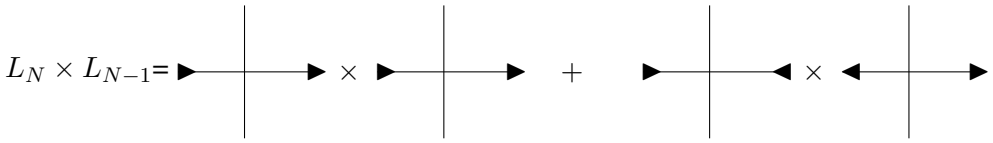
For convenience we will represent this as:

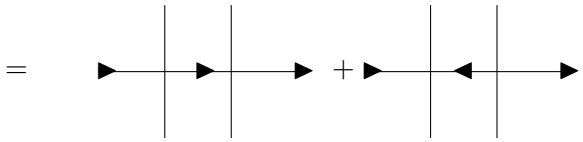
$$L_n = \begin{bmatrix} \alpha_n & \beta_n \\ \gamma_n & \delta_n \end{bmatrix} \tag{1.4.4}$$

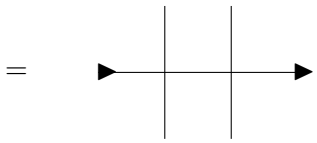
As with L_n, $\alpha_n, \beta_n, \gamma_n, \delta_n$ are operators in $\mathcal{V}$ which act non-trivially in h_n. The matrix L_n is called the *Local L-Operator*, and

$$M = L_N \dots L_1 = \prod_{n=1}^{N} L_n \tag{1.4.5}$$

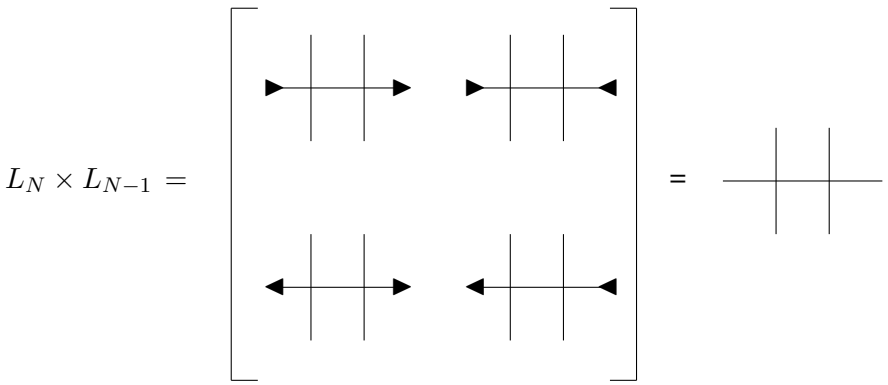
Proof: If we consider $L_N \times L_{N-1}$:







Repeating this for the other vertices, $L_N \times L_{N-1}$ becomes:



Note that the arrows are fixed in precisely the same way as they were in(1.4.3), only the 1×1 entries have become 1×2. This is very similar to the T matrix proof. Repeating this N times, the entries of $L_N \ldots L_1$ will become $1 \times N$ lattices, and all arrow configurations will remain the same. This is precisely matrix M, which we call the *Monodromy Matrix*.

Note that while (1.4.5) appears to be identical to (1.3.2), the former has no restriction on the boundaries of L_n. So M is identical to the T matrix only it does not

have any boundary conditions.

■

Getting back to the original problem: under what conditions will 1.4.1 be true? It turns out that it is sufficient that:

$$R(M \otimes M') = (M' \otimes M)R \tag{1.4.6}$$

Where R is some 4×4 matrix over complex numbers (like L_n).

Proof: If we define tensor product as:

$$M \otimes M' = \begin{bmatrix} \mathcal{A}M' & \mathcal{B}M' \\ \mathcal{C}M' & \mathcal{D}M' \end{bmatrix}$$

Then (1.4.6) can be rewritten as:

$$M \otimes M' = R^{-1}(M' \otimes M)R$$

Taking the trace of both sides gives:

$$Tr(M \otimes M') = Tr(R^{-1}(M' \otimes M)R)$$

Using $Tr(AB) = Tr(BA)$:

$$\begin{aligned} Tr(M \otimes M') &= Tr(R^{-1}(M' \otimes M)R) \\ &= Tr(RR^{-1}(M' \otimes M)) \\ &= Tr(M' \otimes M) \end{aligned}$$

Now using $Tr(A \otimes B) = Tr(A)Tr(B)$ we have:

$$\begin{aligned} Tr(M)Tr(M') &= Tr(M')Tr(M) \\ TT' &= T'T \\ [T, T'] &= 0 \end{aligned}$$

■

Condition (1.4.6) can be even further simplified. Namely, to show (1.4.1) it is sufficient to verify:

$$R(L_n \otimes L'_n) = (L'_n \otimes L_n)R \tag{1.4.7}$$

(1.4.7) is known as the Yang-Baxter equation and will be used frequently in the remainder of this thesis. To show (1.4.7), it is sufficient to prove (1.4.1). We will provided two proofs: a graphical proof (the "Railroad Proof") and a more formal proof.

1.4.1 Railroad Proof

We can represent $R(M \otimes M') = (M' \otimes M)R$ graphically as follows:

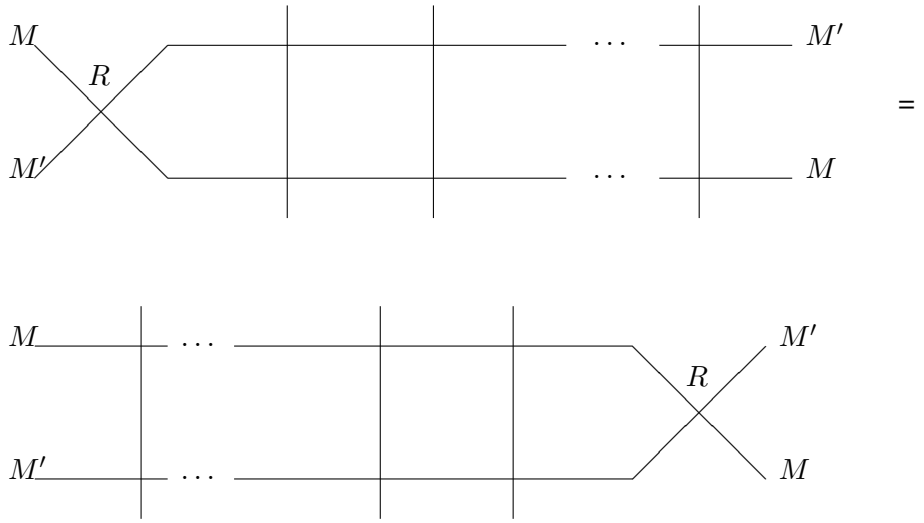


Figure 1

(Note that the only way to represent the product of a single vertex (R) with a $2 \times N$ lattice ($M \otimes M$) is to rotate the single vertex 45^o clockwise and attach it to the lattice (noting that they are non-commutative.) By the same reasoning we can represent $R(L_n \otimes L'_n) = (L'_n \otimes L_n)R$, graphically as:

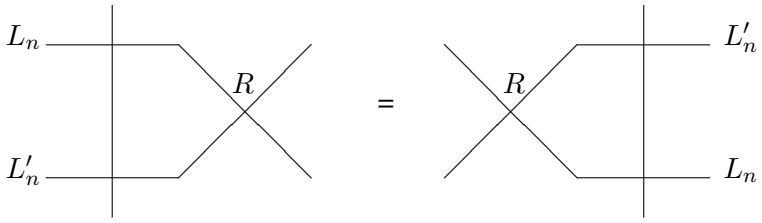
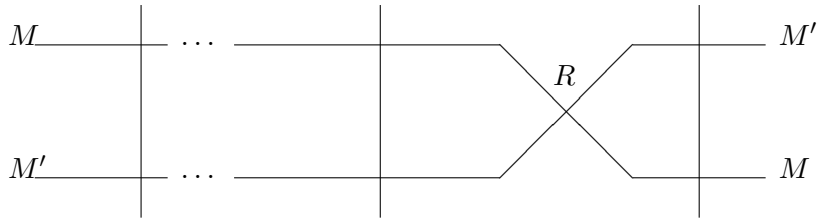
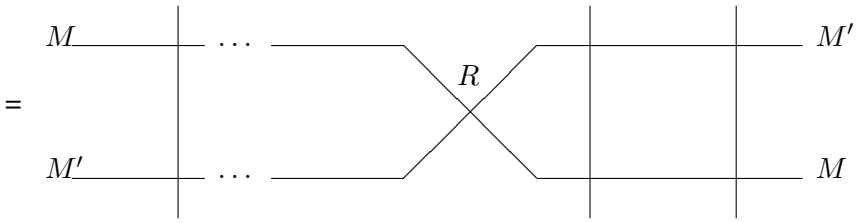


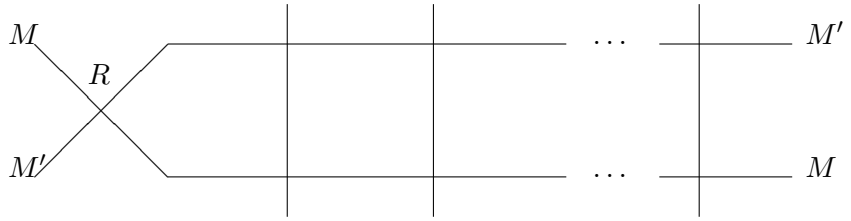
Figure 2

Using Figure 2, the RHS of Figure 1 becomes:





Applying this procedure N times gives:



Which is the RHS. This completes the railroad proof.

■

1.4.2 Formal Proof

Using 1.4.5 we can expand the LHS of 1.4.6 to:

$$\begin{aligned} R(M \otimes M') &= R(L_N, \ldots, L_1) \otimes (L'_N, \ldots, L'_1) \\ &= R(L_N)(L_{N-1} \ldots L_1) \otimes (L'_N)(L'_{N-1} \ldots L'_1) \end{aligned}$$

Using the tensor product of algebras, $AB \otimes CD = (A \otimes C)(B \otimes D)$, gives:

$$R(M \otimes M') = R(L_n \otimes L'_n)(L_{N-1} \ldots L_1) \otimes (L'_{N-1} \ldots L'_1)$$

Now using $R(L_N \otimes L'_N) = (L'_N \otimes L_N)R$ gives:

$$R(M \otimes M') = (L_n \otimes L'_n)R(L_{N-1} \ldots L_1) \otimes (L'_{N-1} \ldots L'_1) \tag{1.4.8}$$

Reiterating this process gives:

$$\begin{aligned} &= (L'_N \otimes L_N)R(L_{N-1})(L_{N-2} \ldots L_1) \otimes (L'_{N-1})(L_{N-2} \ldots L'_1) \\ &= (L'_N \otimes L_N)R(L_{N-1} \otimes L'_{N-1})(L_{N-2} \ldots L_1) \otimes (L_{N-2} \ldots L'_1) \\ &= (L'_N \otimes L_N)(L'_{N-1} \otimes L_{N-1})R(L_{N-2} \ldots L_1) \otimes (L_{N-2} \ldots L'_1) \end{aligned}$$

Using the tensor product of algebras again simplifies the above to:

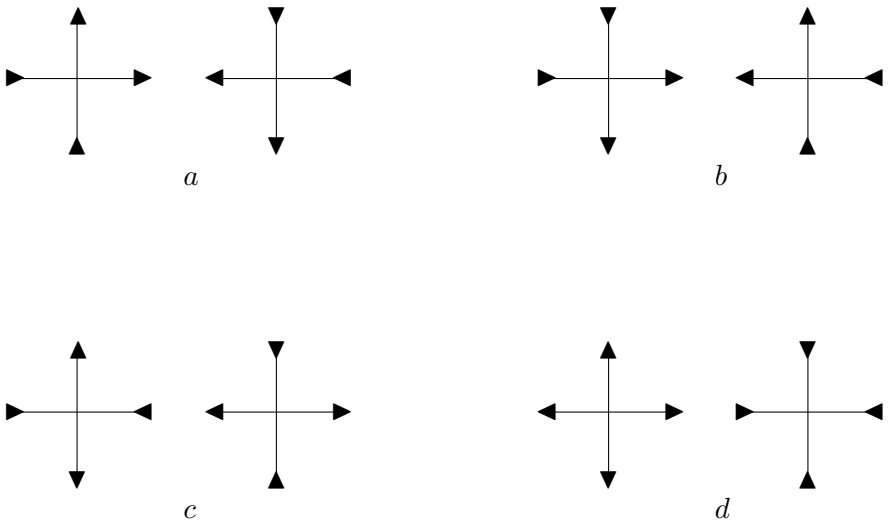
$$\begin{aligned} &= (L'_N L'_{N-1} \otimes L_N L_{N-1})R(L_{N-2} \ldots L_1) \otimes (L_{N-2} \ldots L'_1) \\ &\vdots \\ &= (L'_N, \ldots, L'_1) \otimes (L_N, \ldots, L_1)R \\ &= (M' \otimes M)R \end{aligned}$$

■

In summary, to verify that $[T, T'] = 0$, we need to verify $R(L_n \otimes L'_n) = (L'_n \otimes L_n)R$ rather than $R(M \otimes M') = (M' \otimes M)R$. Our task now becomes solving the Yang-Baxter equation which requires us to find an R matrix for our Boltzmann weights.

1.5 The 8-Vertex Model

So far we have been working with the 16-vertex model, as we have been considering the Boltzmann weights of all 16 vertex configurations. It turns out that no R matrix has been found that will satisfy the Yang-Baxter equation for the 16-vertex model [2], so we can no longer proceed with such generality. Instead, we shall consider the 8-vertex model, in which we only consider 8 vertices and set the weights of the rest to be zero. Assign the following configurations weight a, b, c or d:



There are three important observations to make here:

1. All weights are invariant under reversion of arrows;
2. a, b and c have two arrows going 'in' and two arrows going 'out', which we can think of as conservation of arrow flow; and
3. All configuration that do not have two arrows going 'in' and two arrows going 'out' (except for d) are deemed to have zero weight.

The Local Operator (or weight) matrices, L_n and L'_n, become:

[2] Takhatajan, L. J. Introduction to Agebraic Bethe Ansatz, *Lecture Notes in Physics, Exactly solvable problems in condensed matter and relativistic field theory*, 180

$$L_n = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}_n = \begin{bmatrix} a & 0 & 0 & d \\ 0 & b & c & 0 \\ 0 & c & b & 0 \\ d & 0 & 0 & a \end{bmatrix}_n$$

$$L'_n = \begin{bmatrix} \alpha' & \beta' \\ \gamma' & \delta' \end{bmatrix}_n = \begin{bmatrix} a' & 0 & 0 & d' \\ 0 & b' & c' & 0 \\ 0 & c' & b' & 0 \\ d' & 0 & 0 & a' \end{bmatrix}_n$$

To solve equation (1.4.7) we need to find an R matrix that will satisfy it for all possible L_n and L'_n. Let's try an R matrix of the form:

$$R = PL''_n \tag{1.5.1}$$

Where P is the permutation matrix in $\mathbb{C}^4$ that satisfies

$$P(e \otimes f) = f \otimes e$$

Specifically:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Which gives us:

$$R = \begin{bmatrix} a'' & 0 & 0 & d'' \\ 0 & c'' & b'' & 0 \\ 0 & b'' & c'' & 0 \\ d'' & 0 & 0 & a'' \end{bmatrix}$$

Let a'', b'', c'' and d'' be our new variables. If we substitute L_n and R into equation (1.4.7) the RHS becomes:

$$(L'_n \otimes L_n)R = \begin{bmatrix} \alpha' & \beta' \\ \gamma' & \delta' \end{bmatrix} \otimes \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \begin{bmatrix} a'' & 0 & 0 & d'' \\ 0 & c'' & b'' & 0 \\ 0 & b'' & c'' & 0 \\ d'' & 0 & 0 & a'' \end{bmatrix}$$

$$= \begin{bmatrix} \alpha'\alpha & \alpha'\beta & \beta'\,\alpha & \beta'\beta \\ \alpha'\gamma & \alpha'\delta & \beta'\gamma & \beta'\delta \\ \gamma'\alpha & \gamma'\beta & \delta'\alpha & \delta'\beta \\ \gamma'\gamma & \gamma'\delta & \delta'\gamma & \delta'\delta \end{bmatrix} \begin{bmatrix} a'' & 0 & 0 & d'' \\ 0 & c'' & b'' & 0 \\ 0 & b'' & c'' & 0 \\ d'' & 0 & 0 & a'' \end{bmatrix}$$

$$= \begin{bmatrix} \alpha'\alpha(a'') + \beta'\beta(d'') & \alpha'\beta(c'') + \beta'\alpha(b'') & \alpha'\beta(b'') + \beta'\alpha(c'') & \alpha'\alpha(d'') + \beta'\beta(a'') \\ \alpha'\gamma(a'') + \beta'\delta(d'') & \alpha'\delta(c'') + \beta'\gamma(b'') & \alpha'\delta(b'') + \beta'\gamma(c'') & \alpha'\gamma(d'') + \beta'\delta(a'') \\ \gamma'\alpha(a'') + \delta'\beta(d'') & \gamma'\beta(c'') + \delta'\alpha(b'') & \gamma'\beta(b'') + \delta'\alpha(c'') & \gamma'\alpha(d'') + \delta'\beta(a'') \\ \gamma'\gamma(a'') + \delta'\delta(d'') & \gamma'\delta(c'') + \delta'\gamma(b'') & \gamma'\delta(b'') + \delta'\gamma(c'') & \gamma'\gamma(d'') + \delta'\delta(a'') \end{bmatrix}$$

The LHS of (1.4.7) becomes:

$$R(L_n \otimes L'_n) = \begin{bmatrix} a'' & 0 & 0 & d'' \\ 0 & c'' & b'' & 0 \\ 0 & b'' & c'' & 0 \\ d'' & 0 & 0 & a'' \end{bmatrix} \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \otimes \begin{bmatrix} \alpha' & \beta' \\ \gamma' & \delta' \end{bmatrix}$$

$$= \begin{bmatrix} a'' & 0 & 0 & d'' \\ 0 & c'' & b'' & 0 \\ 0 & b'' & c'' & 0 \\ d'' & 0 & 0 & a'' \end{bmatrix} \begin{bmatrix} \alpha\alpha' & \alpha\beta' & \beta\,\alpha' & \beta\beta' \\ \alpha\gamma' & \alpha\delta' & \beta\gamma' & \beta\delta' \\ \gamma\alpha' & \gamma\beta' & \delta\alpha' & \delta\beta' \\ \gamma\gamma' & \gamma\delta' & \delta\gamma' & \delta\delta' \end{bmatrix}$$

$$= \begin{bmatrix} (a'')\alpha\alpha' + (d'')\gamma\gamma' & (a'')\alpha\beta' + (d'')\gamma\delta' & (a'')\beta\alpha' + (d'')\delta\gamma' & (a'')\beta\beta' + (d'')\delta\delta' \\ (c'')\alpha\gamma' + (b'')\gamma\alpha' & (c'')\alpha\delta' + (b'')\delta\alpha' & (c'')\beta\gamma' + (b'')\gamma\beta' & (c'')\beta\delta' + (b'')\delta\beta' \\ (b'')\alpha\gamma + (c'')\gamma\alpha' & (b'')\alpha\delta' + (c'')\delta\alpha' & (b'')\beta\gamma' + (c'')\gamma\beta' & (b'')\beta\delta' + (c'')\delta\beta' \\ (d'')\alpha\alpha' + (a'')\gamma\gamma' & (d'')\alpha\beta' + (a'')\gamma\delta' & (d'')\beta\alpha' + (a'')\delta\gamma' & (d'')\beta\beta' + (a'')\delta\delta' \end{bmatrix}$$

Equating the LHS and RHS gives 16 equations. We must then expand out α, β, γ and δ in each of the 16 equations and see what we get. The equations in the first row of the first column gives:

$$\alpha'\alpha(a'') + \beta'\beta(d'') = (a'')\alpha\alpha' + (d'')\gamma\gamma'$$

$$\begin{bmatrix} a' & 0 \\ 0 & b' \end{bmatrix}\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}a'' + \begin{bmatrix} 0 & d' \\ c' & 0 \end{bmatrix}\begin{bmatrix} 0 & d \\ c & 0 \end{bmatrix}d'' =$$
$$a''\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}\begin{bmatrix} a' & 0 \\ 0 & b' \end{bmatrix} + d''\begin{bmatrix} 0 & c \\ d & 0 \end{bmatrix}\begin{bmatrix} 0 & c' \\ d' & 0 \end{bmatrix}$$

$$a''\begin{bmatrix} aa' & 0 \\ 0 & bb' \end{bmatrix} + d''\begin{bmatrix} d'c & 0 \\ 0 & c'd \end{bmatrix} = a''\begin{bmatrix} aa' & 0 \\ 0 & bb' \end{bmatrix} + d''\begin{bmatrix} cd' & 0 \\ 0 & dc' \end{bmatrix}$$

Which gives us four equations, two of which state $0 = 0$. The two we are interested in are:

$$a''a'a + d''d'c = a''a'a + d''d'c$$
$$a''b'b + d''c'd = a''b'b + d''c'd$$

We can see immediately the equivalence of the LHS and RHS in both equations, which gives $0 = 0$. Thus, the first of the 16 equations tells us nothing. Let's try the second entry of the first column.

$$\alpha'\gamma(a'') + \beta'\delta(d'') = (c'')\alpha\gamma' + (b'')\gamma\alpha'$$

$$\begin{bmatrix} a' & 0 \\ 0 & b' \end{bmatrix}\begin{bmatrix} 0 & c \\ d & 0 \end{bmatrix}a'' + \begin{bmatrix} 0 & d' \\ c' & 0 \end{bmatrix}\begin{bmatrix} b & 0 \\ 0 & a \end{bmatrix}d'' =$$
$$c''\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}\begin{bmatrix} 0 & c' \\ d' & 0 \end{bmatrix} + b''\begin{bmatrix} 0 & c \\ d & 0 \end{bmatrix}\begin{bmatrix} a' & 0 \\ 0 & b \end{bmatrix}$$

$$a''\begin{bmatrix} 0 & a'c \\ b'd & 0 \end{bmatrix} + d''\begin{bmatrix} 0 & d'a \\ c'b & 0 \end{bmatrix} = c''\begin{bmatrix} 0 & ac' \\ bd' & 0 \end{bmatrix} + b''\begin{bmatrix} 0 & cb' \\ da' & 0 \end{bmatrix}$$

Which gives us the two linearly independent equations (say, set $\mathfrak{A}$):

$$a''a'c + d''d'a = c''c'a + b''b'c$$
$$a''b'd + d''c'b = c''d'b + b''da'$$

The third row of the first column gives:

$$\gamma'\alpha(a'') + \delta'\beta(d'') = (b'')\alpha\gamma' + (c'')\gamma\alpha'$$

$$\begin{bmatrix} 0 & c' \\ d' & 0 \end{bmatrix}\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}a'' + \begin{bmatrix} b' & 0 \\ 0 & a' \end{bmatrix}\begin{bmatrix} 0 & d \\ c & 0 \end{bmatrix}d'' =$$
$$b''\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}\begin{bmatrix} 0 & c' \\ d' & 0 \end{bmatrix} + c''\begin{bmatrix} 0 & c \\ d & 0 \end{bmatrix}\begin{bmatrix} a' & 0 \\ 0 & b \end{bmatrix}$$

$$a''\begin{bmatrix} 0 & c'b \\ d'a & 0 \end{bmatrix} + d''\begin{bmatrix} 0 & b'd \\ a'c & 0 \end{bmatrix} = b''\begin{bmatrix} 0 & ac' \\ bd' & 0 \end{bmatrix} + c''\begin{bmatrix} 0 & cb' \\ da' & 0 \end{bmatrix}$$

This too gives two linearly independent equations (say, set $\mathfrak{B}$):

$$a''c'b + d''b'd = b''c'a + c''b'c$$
$$a''d'a + d''a'c = b''d'b + c''a'd$$

The fourth row of the first column gives:

$$\gamma'\gamma(a'') + \delta'\delta(d'') = (d'')\alpha\alpha' + (a'')\gamma\gamma'$$

$$\begin{bmatrix} 0 & c' \\ d' & 0 \end{bmatrix}\begin{bmatrix} 0 & c \\ d & 0 \end{bmatrix}a'' + \begin{bmatrix} b' & 0 \\ 0 & a' \end{bmatrix}\begin{bmatrix} b & 0 \\ 0 & a \end{bmatrix}d'' =$$
$$d''\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}\begin{bmatrix} a' & 0 \\ 0 & b' \end{bmatrix} + a''\begin{bmatrix} 0 & c \\ d & 0 \end{bmatrix}\begin{bmatrix} 0 & c' \\ d' & 0 \end{bmatrix}$$

$$a''\begin{bmatrix} c'd & 0 \\ 0 & d'c \end{bmatrix} + d''\begin{bmatrix} b'b & 0 \\ 0 & a'a \end{bmatrix} = d''\begin{bmatrix} a'a & 0 \\ 0 & b'b \end{bmatrix} + a''\begin{bmatrix} cd' & 0 \\ c'd & 0 \end{bmatrix}$$

The two equations are identical, so we have one more equation (say, equation $\mathfrak{C}$):

$$a''c'd + d''b'b = d''a'a + a''d'c$$

The second row of the second column gives:

$$\alpha'\gamma(c'') + \beta'\gamma(b'') = (c'')\alpha\delta' + (b'')\delta\alpha'$$

$$\begin{bmatrix} a' & 0 \\ 0 & b' \end{bmatrix}\begin{bmatrix} b & 0 \\ 0 & a \end{bmatrix}c'' + \begin{bmatrix} 0 & d' \\ c' & 0 \end{bmatrix}\begin{bmatrix} 0 & c \\ d & 0 \end{bmatrix}b'' =$$
$$c''\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}\begin{bmatrix} b' & 0 \\ 0 & a' \end{bmatrix} + b''\begin{bmatrix} 0 & c \\ d & 0 \end{bmatrix}\begin{bmatrix} 0 & d' \\ c' & 0 \end{bmatrix}$$

$$c''\begin{bmatrix} a'b & 0 \\ 0 & b'a \end{bmatrix} + b''\begin{bmatrix} d'd & 0 \\ 0 & cc' \end{bmatrix} = c''\begin{bmatrix} ab' & 0 \\ 0 & ba' \end{bmatrix} + b''\begin{bmatrix} cc' & 0 \\ 0 & dd' \end{bmatrix}$$

Again the two equations are identical, giving us the equation (say, equation $\mathfrak{D}$):

$$c''a'b + b''d'd = c''a'b + b''c'c$$

Fortunately out of the 32 equations, only these 6 are linearly independent. They are distributed in the 4×4 matrix as follows:

$$\begin{bmatrix} 0 & \mathfrak{B} & \mathfrak{A} & \mathfrak{C} \\ \mathfrak{A} & \mathfrak{D} & 0 & \mathfrak{B} \\ \mathfrak{B} & 0 & \mathfrak{D} & \mathfrak{A} \\ \mathfrak{C} & \mathfrak{A} & \mathfrak{B} & 0 \end{bmatrix}$$

(Note the symmetry down the diagonal of this distribution.) Our six linearly independent equations are therefore:

$$ca'a'' + ad'd'' = cb'b'' + ac'c'' \tag{1.5.2}$$
$$ba'c'' + dd'b'' = ab'c'' + cc'b'' \tag{1.5.3}$$
$$bc'a'' + db'd'' = ac'b'' + cb'c'' \tag{1.5.4}$$
$$da'b'' + bd'c'' = db'a'' + bc'd'' \tag{1.5.5}$$
$$aa'd'' + cd'a'' = bb'd'' + dc'a'' \tag{1.5.6}$$
$$ad'a'' + ca'd'' = bd'b'' + da'c'' \tag{1.5.7}$$

If we want a non-trivial solution for a'', b'', c'' and d'' we should be able to choose any four of these equations, put them in a matrix and require that the determinant vanishes (otherwise an inverse will exist and all solutions will be zero). If we choose the first, third, fourth and sixth equations the determinant can be found to be (using Mathematica):

$$(abc'd' - cda'b')[(a^2 - b^2)(c'^2 - d'^2) + (c^2 - d^2)(a'^2 - b'^2)] \tag{1.5.8}$$

Which we require to be zero. If all transfer matrices, T and T', are to commute we know that when $a' = a, b' = b, c' = c$ and $d' = d$ that T will be a member of the commuting matrices and satisfy (1.4.6), and the determinant will also vanish. This may not be essential, there may be another a, b, c and d that satisfy the second factor of (1.5.8) and also satisfy the commuting family, however, the second factor will give a much more complicated relation between the primed and unprimed variables; we are only interest in finding a solution, not all solutions. In this case, the fist factor of the determinant will vanish, but the second will not. So our new condition is:

$$\begin{aligned} abc'd' - cda'b' &= 0 \\ abc'd' &= cda'b' \\ \frac{cd}{ab} &= \frac{c'd'}{a'b'} \end{aligned} \tag{1.5.9}$$

If we now solve for the first, third fourth and sixth equations by eliminating each variable (a'', b'', c'' and d'') and then substituting the expression into another (i.e. solving simultaneous equations), using (1.5.9) (and equivalently $abc'd' = a'b'cd$) we can simplify our expressions for the variables to:

$$\begin{aligned} a'' &= a'(cc' - dd')(b'^2c^2 - c'^2a^2)/c' \\ b'' &= b'(cd' - dc')(a'^2c^2 - a'^2a^2)/d' \\ c'' &= c'(bb' - aa')(a'^2c^2 - a'^2a^2)/a' \\ d'' &= d'(a'b - b'a)(b'^2c^2 - c'^2a^2)/b' \end{aligned}$$

Substituting these into either the second or the fifth equation gives:

$$\frac{a^2 + b^2 - c^2 - d^2}{ab} = \frac{a'^2 + b'^2 - c'^2 - d'^2}{a'b'} \tag{1.5.10}$$

If we define

$$\Delta = \frac{a^2 + b^2 - c^2 - d^2}{2(ab + cd)} \qquad \Delta' = \frac{a'^2 + b'^2 - c'^2 - d'^2}{2(a'b' + c'd')}$$

Then we can rewrite (1.5.10) in terms of Δ ,as:

$$\frac{\Delta 2(ab + cd)}{(ab)} = \frac{\Delta' 2(a'b' + c'd')}{(a'b')}$$

which we can rewrite as:

$$\Delta(1+\frac{cd}{ab}) = \Delta'(1+\frac{c'd'}{a'b'})$$

Using (1.5.9):

$$\Delta = \Delta'$$

Now define

$$\Gamma = (ab-cd)/(ab+cd) \qquad \Gamma' = (a'b'-c'd')/(a'b'+c'd')$$

We can easily show $\Gamma = \Gamma'$;

$$\begin{aligned} \frac{cd}{ab} &= \frac{c'd'}{a'b'} \\ 1-\frac{cd}{ab} &= 1-\frac{c'd'}{a'b'} \\ \frac{ab-cd}{ab} &= \frac{a'b'-c'd'}{a'b'} \qquad (1.5.11) \\ ab &= \frac{(ab-cd)a'b'}{a'b'-c'd'} \qquad (1.5.12) \end{aligned}$$

Subbing (1.5.12) into (1.5.11) gives:

$$\begin{aligned} \frac{(cd+ab)(a'b'-c'd')}{(ab-cd)a'b'} &= \frac{a'b'+c'd'}{a'b'} \\ \frac{ab+cd}{ab-cd} &= \frac{a'b'+c'd'}{a'b'-c'd'} \\ \Gamma &= \Gamma' \end{aligned}$$

What this means is that our transfer matrices are ensured to commute if they have the same values of Δ and Γ.

1.6 Parameterization Using Jacobi's Elliptic Functions

Our next step is to parameterize a, b, c and d in terms of four new variables; γ, u, k and λ. If we let:

$$\gamma = (1-\Gamma)/(1+\Gamma) = cd/ab$$

We can eliminate d as follows:

$$2\Delta(ab+cd) = a^2+b^2-c^2-d^2$$
$$2\Delta ab(1+\frac{cd}{ab}) = a^2+b^2-c^2-d^2(\frac{a^2b^2c^2}{a^2b^2c^2})$$
$$2\Delta(1+\gamma)ab = a^2+b^2-c^2-a^2b^2\gamma^2c^{-2}$$

If we then divide through by c^2 we have:

$$2\Delta(1+\gamma)(a/c)(b/c) = (a/c)^2+(b/c)^2+(a/c)^2(b/c)^2\gamma^2-1$$

This can be rewritten as a quadratic with a/c as our variable.

$$(a/c)^2(1-\gamma^2(b/c)^2)-2\Delta(1+\gamma)(b/c)(a/c)+(b/c)^2-1=0$$

The discriminant of this quadratic (without the factor of 4) is:

$$\Delta^2(1+\gamma)^2(b/c)^2-[(b/c)^2-1][1-\gamma^2(b/c)^2] \tag{1.6.1}$$

which is a quadratic in $(b/c)^2$ and can be rewritten:

$$(1-y^2(b/c)^2)(1-k^2y^2(b/c)^2)$$

where

$$k^2y^4 = \gamma^2 \tag{1.6.2}$$

$$(1+k^2)y^2 = 1+\gamma^2-\Delta^2(1+\gamma)^2 \tag{1.6.3}$$

We can now parameterize this equation using elliptic functions by making the substitution:

$$(b/c) = y^{-1}\text{sn}(i\lambda,k) \tag{1.6.4}$$

Where sn(λ,k) denotes elliptic sine of modulus k. The reason for using i in the argument will soon become clear. We can now solve for (a/c) using the quadratic formula:

$$(a/c) = \frac{2\Delta(1+\gamma)(b/c) \pm \sqrt{4(1-y^2(b/c)^2)(1-k^2y^2(b/c)^2)}}{2(1-\gamma^2(b/c)^2}$$

$$= \frac{\Delta(1+\gamma)\frac{1}{y}\text{sn}(i\lambda) \pm \sqrt{(1-\text{sn}^2(i\lambda))(1-k^2\text{sn}^2(i\lambda)}}{1-\frac{\gamma^2}{y^2}\text{sn}^2(i\lambda)}$$

using the elliptic identities:

$$\text{cn}^2(u) + \text{sn}^2(u) = 1$$
$$\text{dn}^2(u) + k^2\text{sn}^2(u) = 1$$

(a/c) becomes:

$$(a/c) = \frac{\Delta(1+\gamma)\frac{1}{y}\text{sn}(i\lambda) \pm \sqrt{(\text{cn}^2(i\lambda)\text{dn}^2(i\lambda)}}{(1-\frac{\gamma^2}{y^2}\text{sn}^2(i\lambda)}$$

$$= \frac{y[\Delta(1+\gamma)\text{sn}(i\lambda) + y\text{cn}(i\lambda)\text{dn}(i\lambda)]}{y^2-\gamma^2\text{sn}^2(i\lambda)}$$

This equation can be simplified further by defining:

$$k\text{sn}(iu) = -\frac{\gamma}{y} \tag{1.6.5}$$

Then we get from Equation (1.6.2):

$$k^2y^2 = (\frac{\gamma}{y})^2 \tag{1.6.6}$$
$$= (-k\text{sn}(iu))^2$$
$$= k^2\text{sn}^2(iu)$$
$$\Rightarrow y = \text{sn}(iu) \tag{1.6.7}$$

Equation (1.6.5) gives:

$$\gamma = -yk\text{sn}(iu) \tag{1.6.8}$$
$$\Rightarrow \gamma = -k\text{sn}^2(iu) \tag{1.6.9}$$

Equations (1.6.3) and (1.6.7) give:

$$\begin{aligned}
\Delta^2 &= \frac{-1}{(1+\gamma)^2}((1+k^2)y^2 - 1 - \gamma^2) \\
&= \frac{-1}{(1-k\text{sn}^2(iu))}((1+k^2)\text{sn}^2(iu) - 1 - k^2\text{sn}^4(iu)) \\
&= \frac{1}{(1-k\text{sn}^2(iu))}(1 + k^2\text{sn}^4(iu) - \text{sn}^2(iu) - k^2\text{sn}^2(iu)) \\
&= \frac{1}{(1-k\text{sn}^2(iu))}(1 - \text{sn}^2(iu))(1 - k^2\text{sn}^2(iu)) \\
\Rightarrow \Delta &= \frac{-1}{(1-k\text{sn}^2(iu))}\text{cn}(iu)\text{dn}(iu)
\end{aligned}$$

Where we have taken the negative square root of Δ for physical reasons. Now that we have Δ, y and γ we can simplify our expression for (a/c)

$$\begin{aligned}
(a/c) &= \frac{y[\Delta(1+\gamma)\text{sn}(i\lambda) + y\text{cn}(i\lambda)\text{dn}(i\lambda)}{y^2 - \gamma^2\text{sn}(i\lambda)} \\
&= \frac{\text{sn}(iu)[\frac{(-\text{cn}(i\lambda)\text{dn}(i\lambda))}{1-k\text{sn}^2(iu)}(1 - k^2\text{sn}(iu)) + \text{sn}(iu)\text{cn}(i\lambda)\text{dn}(i\lambda)]}{\text{sn}^2(iu)(1 - k^2\text{sn}^2(iu)\text{sn}^2(i\lambda))} \\
&= \frac{1}{\text{sn}(iu)}\frac{\text{sn}(iu)\text{cn}(i\lambda)\text{dn}(i\lambda) - \text{cn}(iu)\text{dn}(iu)\text{sn}(i\lambda)}{1 - k^2\text{sn}^2(iu)\text{sn}^2(i\lambda)}
\end{aligned}$$

Using the identity [3]:

$$\text{sn}(u-v) = \frac{\text{sn}(u)\text{cn}(v)\text{dn}(v) - \text{cn}(u)\text{dn}(u)\text{sn}(v)}{1 - k^2\text{sn}^2(u)\text{sn}^2(v)}$$

(a/c) becomes:

$$(a/c) = \frac{1}{\text{sn}(iu)} \times \text{sn}(i(u-\lambda))$$

We can also use our new expressions for y to simplify (b/c):

$$\begin{aligned}
(b/c) &= y^{-1}\text{sn}(i\lambda) \\
&= \frac{\text{sn}(i\lambda)}{\text{sn}(iu)}
\end{aligned}$$

[3] Baxter, Rodney. J, *Exactly Solved Models in Statistical Mechanics,* (Academic Press, 1982), 463

Now we can find an expression for (d/c) using :

$$\begin{aligned}
\gamma &= \frac{cd}{ab} \\
\frac{ab}{c^2}\gamma &= c/d \\
(a/c)(b/c)\gamma &= d/c \\
d/c &= \frac{\text{sn}(i(u-\lambda))}{\text{sn}(iu)} \times \frac{\text{sn}(i\lambda)}{\text{sn}(iu)} \times -k^2\text{sn}^2(iu) \\
\Rightarrow (d/c) &= -k\text{sn}(i(u-\lambda))\text{sn}(i\lambda)
\end{aligned}$$

We now have three ratios for a, b, c and d.

$$a/c = \frac{\text{sn}(i(u-\lambda))}{\text{sn}(iu)} \tag{1.6.10}$$

$$b/c = \frac{\text{sn}(i\lambda)}{\text{sn}(iu)} \tag{1.6.11}$$

$$d/c = -k\text{sn}(i(u-\lambda))\text{sn}(i\lambda) \tag{1.6.12}$$

Using the hyperbolic elliptic sine, defined by:

$$\text{snh}(u) = -i\text{sn}(iu) = i\text{sn}(-iu) \quad u \in \mathbb{R}, 0 < k < 1$$

we can simplify our ratios to:

$$\begin{aligned}
a/c &= \frac{\text{snh}(u-\lambda)}{\text{snh}(u)} \\
b/c &= \frac{\text{snh}(\lambda)}{\text{snh}(u)} \\
d/c &= k\text{snh}(u-\lambda)\text{snh}(\lambda)
\end{aligned}$$

We can now solve a, b, c and d up to a normalization factor ρ

$$\begin{aligned}
a &= \rho\text{snh}(u-\lambda) \\
b &= \rho\text{snh}(\lambda) \\
c &= \rho\text{snh}(u) \\
d &= \rho k\text{snh}(u)\text{snh}(\lambda)\text{snh}(u-\lambda)
\end{aligned}$$

Now we have now parameterized a, b, c and d in terms of ρ, λ, u and k. (a, b, c and d can be further defined using theta functions. We will not use these, as they are quite complicated, do not simplify the problem and are therefore unnecessary for our purposes.)

Keeping k, ρ and u fixed, we can regard the transfer matrix, T, as a function of λ, $T(\lambda)$, and our commuting Transfer matrix, T' as a function of μ, that is: $T' \rightarrow T(\mu)$. We can now express 1.4.1 as:

$$T(\lambda)T(\mu) = T(\mu)T(\lambda)$$

This implies:

$$L \rightarrow L(\lambda) \quad L' \rightarrow L(\mu)$$

1.7 Relation Between λ, μ and η

Equations (1.5.2) - (1.5.7) are unaltered when we swap the primed and double primed variables ($1 \rightarrow 1, 2 \rightarrow 3, 3 \rightarrow 2, 4 \rightarrow 4, 5 \rightarrow 6,\ 6 \rightarrow 5$). Therefore, the relation we found between the primed and unprimed variables should be identical to the relation between the unprimed and double-primed variables. Thus:

$$\Gamma = \Gamma' = \Gamma'' \qquad \Delta = \Delta' = \Delta''$$

Given that Δ and Γ only depend on k and u, we should be able to parameterize our new variables, a'', b'', c'' and d'', in terms of k, u and some new variables, say η and ρ'. Rewriting (1.5.2) as:

$$\begin{aligned} c(a'a'' - b'b'') &= a(c'c'' - d'd'') \\ \frac{1}{cc'c''}c(a'a'' - b'b'') &= \frac{1}{cc'c''}a(c'c'' - d'd'') \\ (\frac{a'}{c'}\frac{a''}{c''} - \frac{b'}{c'}\frac{b''}{c''}) &= \frac{a}{c}(1 - \frac{d'}{c'}\frac{d''}{c''}) \end{aligned}$$

Using (1.6.10) - (1.6.12)

$$\begin{aligned} &\frac{\text{sn}(u-\mu)}{\text{sn}(u)}\frac{\text{sn}(u-\eta)}{\text{sn}(u)} - \frac{\text{sn}(\mu)}{\text{sn}(u)}\frac{\text{sn}(\eta)}{\text{sn}(u)} \\ = &\frac{\text{sn}(u-\lambda)}{\text{sn}(u)}(1 - (-k)\text{sn}(u-\mu)\text{sn}(u)(-k)\text{sn}(u-\eta)\text{sn}(u)) \end{aligned}$$

Canceling the $\text{sn}(u)$ gives:

$$\frac{\text{sn}(u-\mu)\text{sn}(u-\eta)-\text{sn}(\mu)\text{sn}(\eta)}{\text{sn}(u)}$$
$$=\text{sn}(u-\lambda)(1-k^2\text{sn}(u-\mu)\text{sn}(u)\text{sn}(u-\eta)\text{sn}(u))$$

Which becomes:

$$\frac{\text{sn}(u-\mu)\text{sn}(u-\eta)-\text{sn}(\mu)\text{sn}(\eta)}{1-k^2\text{sn}(u-\mu)\text{sn}(u)\text{sn}(u-\eta)\text{sn}(u)} =\text{sn}(u-\lambda)\text{sn}(u)$$

Using the identity [4]:

$$\frac{\text{sn}(a-u)\text{sn}(a-v)-\text{sn}(u)\text{sn}(v)}{1-k^2\text{sn}(u)\text{sn}(v)\text{sn}(a-u)\text{sn}(a-v)}=\text{sn}(a)\text{sn}(a-u-v)$$

our equation becomes:

$$\begin{aligned}\text{sn}(u)\text{sn}(u-\mu-\eta) &=\text{sn}(u)\text{sn}(u-\lambda)\\ \text{sn}(u-\mu-\eta) &=\text{sn}(u-\lambda)\\ u-\mu-\eta &=u-\lambda\\ \eta &=\lambda-\mu\end{aligned}$$

It turns out that the parameter for the double primed variables depends on the parameters of the unprimed and primed variables! We can now write (1.4.6) as:

$$R(\lambda-\mu)(L_n(\lambda)\otimes L_n(\mu))=(L_n(\mu)\otimes L_n(\lambda))R(\lambda-\mu)$$

This is the solution to the Yang-Baxter equation.

[4]Baxter, Rodney. J, *Exactly Solved Models in Statistical Mechanics,* (Academic Press, 1982), 463

Chapter 2

Algebraic Bethe Ansatz in "abc-form"

2.1 Triangularizing M

We have solved the Yang-Baxter equation and have the conditions required for the transfer matrix T to lie in a commuting family $T(\lambda)$. Now we will start the procedure for diagonalization of T. For simplicity we will consider the 6-vertex case, where $d = 0$. This is known as the 'ice-rule', [1] and is equivalent to letting $k \to 0$, so that $\mathrm{snh}(u) \to \sinh(u)$ and our elliptic functions become trigonometric ones. If we take $\rho = 1$ and $\rho' = 1$ for simplicity:

$$a = \sinh(u - \lambda) \tag{2.1.1}$$

$$b = \sinh(\lambda) \tag{2.1.2}$$

$$c = \sinh(u) \tag{2.1.3}$$

$$d = 0 \tag{2.1.4}$$

It is important to see that c is independent of the repidities, but still depends on the crossing parameter, u. Nonetheless, we will keep the argument of c in our calculations.

[1] We can think of the lattice as a block of H_20 atoms. Each vertex represents an oxygen atom anf the directed arrows represent the sharing of a hydrogen atom. An in arrow represents taking half a hydrogen atom from a neighbour, an out arrow represents the giving of half an hydrogen atom to a neighbour.

Considering L_n as a 2×2 matrix, there exists a vector $w_n = \begin{pmatrix} 1 \\ 0 \end{pmatrix}_n$ such that operator L_n acting on w_n becomes upper triangular. That is:

$$L_n \hat{\otimes} w_n = \begin{bmatrix} \alpha_n & \beta_n \\ \gamma_n & \delta_n \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}_n$$

$$= \begin{bmatrix} \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ c & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 & c \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} b & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{bmatrix}_n$$

$$= \begin{bmatrix} a & 0 \\ 0 & c \\ 0 & b \\ 0 & 0 \end{bmatrix}_n$$

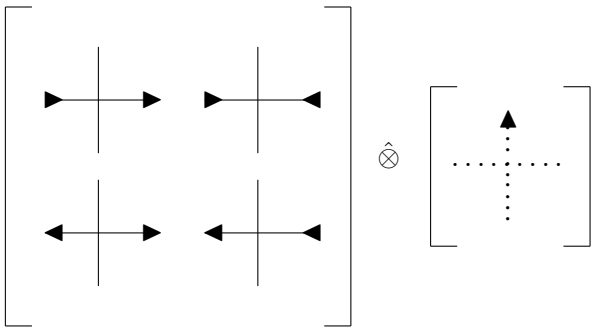
$$= \begin{bmatrix} a \begin{pmatrix} 1 \\ 0 \end{pmatrix} & * \\ 0 & b \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{bmatrix}_n$$

$$= \begin{bmatrix} a(\lambda) w_n & * \\ 0 & b(\lambda) w_n \end{bmatrix}$$

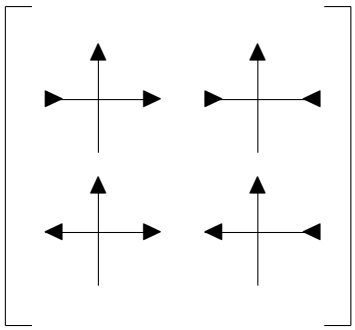
We represent the top right matrix entry with an asterix as its value is not required.

For the graphical proof, we must interpret w_n as placing an up-arrow on the top edge of a vertex. This is best illustrated graphically:

$L_n \hat{\otimes} w_n \quad =$

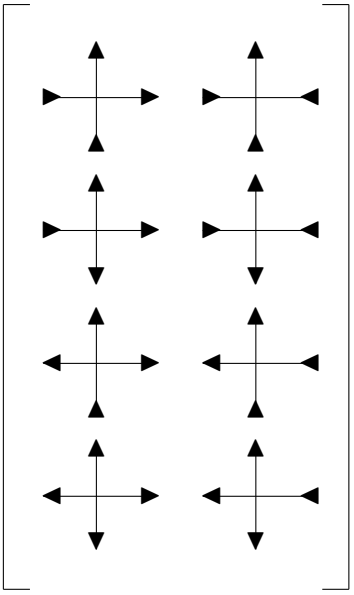


$=$



The notation $\hat{\otimes}$ treats the 2×2 matrix L_n is operating on line a constant, which is why we may bring it inside the matrix.

In the 6-vertex model, the bottom left configuration does not satisfy the conservation of arrow flow and is zero. However, we will continue to consider it for completeness. If we now sum each entry over all configurations of the bottom edge our matrix becomes:

$=$ 

Recalling that a configuration that does not have two arrows in and two arrows out is assigned a weight of zero, the matrix becomes:

$$= \begin{bmatrix} a & 0 \\ 0 & c \\ 0 & b \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} a(\lambda)w_n & * \\ 0 & b(\lambda)w_n \end{bmatrix}$$

Aside

At this point it should be clear what is meant by 'operating on'. L_n acts on vectors v_n (which must lie in the same space; hence v_n and not v_m) and produces a particular arrow configuration, restricting the values the vertices may take. If L_n acts on four well defined vectors it will produce a particular vertex configuration from L_n.

The vectors, specifying right, left, up and down arrows, etc, are defined:

$$\begin{aligned}
\text{Top } \blacktriangle &= \hat{\otimes} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
\text{Top } \blacktriangledown &= \hat{\otimes} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
\text{Bottom } \blacktriangle &= \begin{bmatrix} 1 & 0 \end{bmatrix} \hat{\otimes} \\
\text{Bottom } \blacktriangledown &= \begin{bmatrix} 0 & 1 \end{bmatrix} \hat{\otimes} \\
\text{Right } \blacktriangleright &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
\text{Right } \blacktriangleleft &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
\text{Left } \blacktriangleright &= \begin{bmatrix} 1 & 0 \end{bmatrix} \\
\text{Left } \blacktriangleleft &= \begin{bmatrix} 0 & 1 \end{bmatrix}
\end{aligned}$$

Again, $\hat{\otimes}$ works lets L_n in the form of a 2×2 matrix treat the vector as a constant and bring it to the matrix, only then to operate under normal matrix multiplication. For example, if we wanted to finish with a c vertex, we would need L_n to operate the following vectors:

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \left(\begin{bmatrix} 0 & 1 \end{bmatrix} \hat{\otimes} \right) L_n \left(\hat{\otimes} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

As the matrices lie in different spaces, they are commutative on each side and L_n may act on them in any order. Starting by specifying the top edge as an up arrow:

$$
\begin{aligned}
&\left[\begin{array}{cccc} a & 0 & 0 & 0 \\ 0 & b & c & 0 \\ 0 & c & b & 0 \\ 0 & 0 & 0 & a \end{array}\right]_n \hat{\otimes} \left[\begin{array}{c} 1 \\ 0 \end{array}\right]_n \\
&= \left[\begin{array}{cc} \left(\begin{array}{cc} a & 0 \\ 0 & b \end{array}\right)\left(\begin{array}{c} 1 \\ 0 \end{array}\right) & \left(\begin{array}{cc} 0 & 0 \\ c & 0 \end{array}\right)\left(\begin{array}{c} 1 \\ 0 \end{array}\right) \\ \left(\begin{array}{cc} 0 & c \\ 0 & 0 \end{array}\right)\left(\begin{array}{c} 1 \\ 0 \end{array}\right) & \left(\begin{array}{cc} b & 0 \\ 0 & 0 \end{array}\right)\left(\begin{array}{c} 1 \\ 0 \end{array}\right) \end{array}\right]_n \\
&= \left[\begin{array}{cc} a & 0 \\ 0 & c \\ 0 & b \\ 0 & 0 \end{array}\right]_n \\
&= \left[\begin{array}{c} \alpha \\ \delta \end{array}\right]_n
\end{aligned}
$$

Specify the bottom edge as a down arrow

$$
\begin{aligned}
&\left[\begin{array}{cc} 0 & 1 \end{array}\right]_n \hat{\otimes} \left[\begin{array}{cc} a & 0 \\ 0 & c \\ 0 & b \\ 0 & 0 \end{array}\right]_n \\
&= \left[\begin{array}{c} \left(\begin{array}{cc} 0 & 1 \end{array}\right)\left(\begin{array}{cc} a & 0 \\ 0 & c \end{array}\right) \\ \left(\begin{array}{cc} 0 & 1 \end{array}\right)\left(\begin{array}{cc} 0 & b \\ 0 & 0 \end{array}\right) \end{array}\right]_n \\
&= \left[\begin{array}{cc} 0 & c \\ 0 & 0 \end{array}\right]_n
\end{aligned}
$$

Specify the left edge as a right arrow

$$
\left[\begin{array}{cc} 1 & 0 \end{array}\right]_n \left[\begin{array}{cc} 0 & c \\ 0 & 0 \end{array}\right]_n = \left[\begin{array}{cc} 0 & c \end{array}\right]_n
$$

Finally, specify the right edge as top edge as left arrow

$$\left[\begin{array}{cc} 0 & c \end{array}\right]_n \left[\begin{array}{c} 0 \\ 1 \end{array}\right]_n = c$$

It is worth noting that trying to specify two different arrows at the same position will result in an undefined matrix multiplication, such as $M_{mn} \times N_{pq}$ where $n \neq p$.

Graphically, the operator L_n can be thought of as starting with representation (1.3.1), and after each operation (specifying an edge and arrow direction) we remove from (1.3.1) all the entries that do not have the property that v_n specifies. For example, in the first operation in the above example we removed all vertices that did not have an up arrow on the top.

■

The point is that when operator L_n acts on w_n, the result is upper triangular! We call vector w_n the *local vacuum*. Given w_n makes L_n upper trianglar, we would expect there to exist a vector that has the same effect on the monodromy matrix $M(\lambda)$. Such a vector does exist and it is:

$$\Omega = \prod_{n=1}^{N} \otimes w_n \quad \in \mathcal{V}_N$$

Ω can be thought of as N 'up' arrows for top edges. Similarly with w_n, the arrows do not lie on a lattice, but are operated on by blank lattices. Ω is called the 'reference state', and $M(\lambda)$ acts on on Ω to give:

$$M(\lambda)\Omega = \left[\begin{array}{cc} a^N\Omega & * \\ 0 & b^N\Omega \end{array}\right]$$

Proof: Because L_n may only act on its own space (for example, L_3 cannot operate

on w_2), we can rearrange the order of multiplication as follows:

$$
\begin{aligned}
M\Omega &= \prod_{n=1}^{N} L_n(\lambda) \prod_{n=1}^{N} \otimes w_n \\
&= \prod_{n=1}^{N} \otimes L_n(\lambda) w_n \\
&= L_1 w_1 \otimes L_2 w_2 \otimes \ldots \otimes L_N w_n \\
&= \begin{bmatrix} a(\lambda)w_1 & * \\ 0 & b(\lambda)w_1 \end{bmatrix} \otimes \begin{bmatrix} a(\lambda)w_2 & * \\ 0 & b(\lambda)w_2 \end{bmatrix} \otimes \ldots \otimes \begin{bmatrix} a(\lambda)w_N & * \\ 0 & b(\lambda)w_N \end{bmatrix}
\end{aligned}
$$

using the rule for multiplication of triangular matrices

$$
\begin{aligned}
&= \begin{bmatrix} a(\lambda)w_1 \otimes a(\lambda)w_2 \otimes \ldots \otimes a(\lambda)w_N & * \\ 0 & b(\lambda)w_1 \otimes b(\lambda)w_2 \otimes \ldots \otimes b(\lambda)w_N \end{bmatrix} \\
&= \begin{bmatrix} a^N(\lambda)w_1 \otimes w_2 \otimes \ldots \otimes w_N & * \\ 0 & b^N(\lambda)w_1 \otimes w_2 \otimes \ldots \otimes w_N \end{bmatrix} \\
&= \begin{bmatrix} a^N(\lambda)\Omega & * \\ 0 & b^N(\lambda)\Omega \end{bmatrix}
\end{aligned}
$$

Graphically, the proof is very similar to triangularizing $L_n w_n$, only the Ω 'freezes' the top row of the $\mathcal{A}$ lattice to give the $a^N(\lambda)$ and the top row of the $\mathcal{D}$ lattice $b^N(\lambda)$.

■

Recalling that $T = trace(M)$ we can see:

$$
\begin{aligned}
T(\lambda)\Omega &= (\mathcal{A}(\lambda) + \mathcal{D}(\lambda))\Omega \\
&= (a^N(\lambda) + b^N(\lambda))\Omega
\end{aligned}
$$

Thus, Ω is the eigenvector of the T matrix and $a(\lambda)^N + b(\lambda)^N$ is the eigenvalue. This is the ansatz in 'Bethe ansatz': The assumption that this procedure generates all eigenvectors. This is not very clear in the limit $N \to \infty$.

Now we wish to construct other eigenvectors of $T(\lambda)$. To do this we make use of (1.4.6):

$$
R(\lambda - \mu)(M(\mu) \otimes M(\lambda) = (M(\lambda) \otimes M(\mu) R(\lambda - \mu)
$$

$$\begin{bmatrix} a(\lambda-\mu) & 0 & 0 & 0 \\ 0 & c(\lambda-\mu) & b(\lambda-\mu) & 0 \\ 0 & b(\lambda-\mu) & c(\lambda-\mu) & 0 \\ 0 & 0 & 0 & a(\lambda-\mu) \end{bmatrix} \begin{bmatrix} \mathcal{A}(\lambda) & \mathcal{B}(\lambda) \\ \mathcal{C}(\lambda) & \mathcal{D}(\lambda) \end{bmatrix} \otimes \begin{bmatrix} \mathcal{A}(\mu) & \mathcal{B}(\mu) \\ \mathcal{C}(\mu) & \mathcal{D}(\mu) \end{bmatrix}$$

$$= \begin{bmatrix} \mathcal{A}(\mu) & \mathcal{B}(\mu) \\ \mathcal{C}(\mu) & \mathcal{D}(\mu) \end{bmatrix} \begin{bmatrix} \mathcal{A}(\lambda) & \mathcal{B}(\lambda) \\ \mathcal{C}(\lambda) & \mathcal{D}(\lambda) \end{bmatrix} \otimes \begin{bmatrix} a(\lambda-\mu) & 0 & 0 & 0 \\ 0 & c(\lambda-\mu) & b(\lambda-\mu) & 0 \\ 0 & b(\lambda-\mu) & c(\lambda-\mu) & 0 \\ 0 & 0 & 0 & a(\lambda-\mu) \end{bmatrix}$$

(**Aside**: Equation 1.4.6 tells us R is only defined up to a constant, α:

$$R(M \otimes M') = (M' \otimes M)R$$
$$\alpha R'(M \otimes M') = (M' \otimes M)\alpha R'$$
$$R'(M \otimes M') = (M' \otimes M)R'$$

Thus, if R solves the Yang-Baxter equation, so does R'. Accordingly, at this point, some authors choose to divide through by a, b or c, so that there are only 2 variables (say, $b/a = \delta$ and $c/a = \gamma$). As this does not significantly simplify the equations, for now we will continue using the same variables.)

Equating the LHS and RHS we get 16 equations. For convenience we will define:

$$a(\lambda-\mu) = a_{\lambda,\mu} \quad b(\lambda-\mu) = b_{\lambda,\mu} \quad c(\lambda-\mu) = c_{\lambda,\mu}$$

Which gives us:

$$\begin{bmatrix} a_{\lambda,\mu} & 0 & 0 & 0 \\ 0 & c_{\lambda,\mu} & b_{\lambda,\mu} & 0 \\ 0 & b_{\lambda,\mu} & c_{\lambda,\mu} & 0 \\ 0 & 0 & 0 & a_{\lambda,\mu} \end{bmatrix} \begin{bmatrix} \mathcal{A}(\lambda)\mathcal{A}(\mu) & \mathcal{A}(\lambda)\mathcal{B}(\mu) & \mathcal{B}(\lambda)\mathcal{A}(\mu) & \mathcal{B}(\lambda)\mathcal{B}(\mu) \\ \mathcal{A}(\lambda)\mathcal{C}(\mu) & \mathcal{A}(\lambda)\mathcal{D}(\mu) & \mathcal{B}(\lambda)\mathcal{C}(\mu) & \mathcal{B}(\lambda)\mathcal{D}(\mu) \\ \mathcal{C}(\lambda)\mathcal{A}(\mu) & \mathcal{C}(\lambda)\mathcal{B}(\mu) & \mathcal{D}(\lambda)\mathcal{A}(\mu) & \mathcal{D}(\lambda)\mathcal{B}(\mu) \\ \mathcal{C}(\lambda)\mathcal{C}(\mu) & \mathcal{C}(\lambda)\mathcal{D}(\mu) & \mathcal{D}(\lambda)\mathcal{C}(\mu) & \mathcal{D}(\lambda)\mathcal{D}(\mu) \end{bmatrix}$$

$$= \begin{bmatrix} \mathcal{A}(\mu)\mathcal{A}(\lambda) & \mathcal{A}(\mu)\mathcal{B}(\lambda) & \mathcal{B}(\mu)\mathcal{A}(\lambda) & \mathcal{B}(\mu)\mathcal{B}(\lambda) \\ \mathcal{A}(\mu)\mathcal{C}(\lambda) & \mathcal{A}(\mu)\mathcal{D}(\lambda) & \mathcal{B}(\mu)\mathcal{C}(\lambda) & \mathcal{B}(\mu)\mathcal{D}(\lambda) \\ \mathcal{C}(\mu)\mathcal{A}(\lambda) & \mathcal{C}(\mu)\mathcal{B}(\lambda) & \mathcal{D}(\mu)\mathcal{A}(\lambda) & \mathcal{D}(\mu)\mathcal{B}(\lambda) \\ \mathcal{C}(\mu)\mathcal{C}(\lambda) & \mathcal{C}(\mu)\mathcal{D}(\lambda) & \mathcal{D}(\mu)\mathcal{C}(\lambda) & \mathcal{D}(\mu)\mathcal{D}(\lambda) \end{bmatrix} \begin{bmatrix} a_{\lambda,\mu} & 0 & 0 & 0 \\ 0 & c_{\lambda,\mu} & b_{\lambda,\mu} & 0 \\ 0 & b_{\lambda,\mu} & c_{\lambda,\mu} & 0 \\ 0 & 0 & 0 & a_{\lambda,\mu} \end{bmatrix}$$

When these are expanded out, they give 16 equations. These are

$$a_{\lambda,\mu}\mathcal{A}(\lambda)\mathcal{A}(\mu) = \mathcal{A}(\mu)\mathcal{A}(\lambda)a_{\lambda,\mu} \tag{2.1.5}$$

$$a_{\lambda,\mu}\mathcal{A}(\lambda)\mathcal{B}(\mu) = \mathcal{A}(\mu)\mathcal{B}(\lambda)c_{\lambda,\mu} + \mathcal{B}(\mu)\mathcal{A}(\lambda)b_{\lambda,\mu} \tag{2.1.6}$$

$$a_{\lambda,\mu}\mathcal{B}(\lambda)\mathcal{A}(\mu) = \mathcal{A}(\mu)\mathcal{B}(\lambda)b_{\lambda,\mu} + \mathcal{B}(\mu)\mathcal{A}(\lambda)c_{\lambda,\mu} \tag{2.1.7}$$

$$a_{\lambda,\mu}\mathcal{B}(\lambda)\mathcal{B}(\mu) = \mathcal{B}(\mu)\mathcal{B}(\lambda)a_{\lambda,\mu} \tag{2.1.8}$$

$$c_{\lambda,\mu}\mathcal{A}(\lambda)\mathcal{C}(\mu) + b_{\lambda,\mu}\mathcal{C}(\lambda)\mathcal{A}(\mu) = \mathcal{A}(\mu)\mathcal{C}(\lambda)a_{\lambda,\mu} \tag{2.1.9}$$

$$c_{\lambda,\mu}\mathcal{A}(\lambda)\mathcal{D}(\mu) + b_{\lambda,\mu}\mathcal{C}(\lambda)\mathcal{B}(\mu) = \mathcal{A}(\mu)\mathcal{D}(\lambda)c_{\lambda,\mu} + \mathcal{B}(\mu)\mathcal{C}(\lambda)b_{\lambda,\mu} \tag{2.1.10}$$

$$c_{\lambda,\mu}\mathcal{B}(\lambda)\mathcal{C}(\mu) + b_{\lambda,\mu}\mathcal{D}(\lambda)\mathcal{A}(\mu) = \mathcal{A}(\mu)\mathcal{D}(\lambda)b_{\lambda,\mu} + \mathcal{B}(\mu)\mathcal{C}(\lambda)c_{\lambda,\mu} \tag{2.1.11}$$

$$c_{\lambda,\mu}\mathcal{B}(\lambda)\mathcal{D}(\mu) + b_{\lambda,\mu}\mathcal{D}(\lambda)\mathcal{B}(\mu) = \mathcal{B}(\mu)\mathcal{D}(\lambda)a_{\lambda,\mu} \tag{2.1.12}$$

$$b_{\lambda,\mu}\mathcal{A}(\lambda)\mathcal{C}(\mu) + c_{\lambda,\mu}\mathcal{C}(\lambda)\mathcal{A}(\mu) = \mathcal{C}(\mu)\mathcal{A}(\lambda)a_{\lambda,\mu} \tag{2.1.13}$$

$$b_{\lambda,\mu}\mathcal{A}(\lambda)\mathcal{D}(\mu) + c_{\lambda,\mu}\mathcal{C}(\lambda)\mathcal{B}(\mu) = \mathcal{C}(\mu)\mathcal{B}(\lambda)c_{\lambda,\mu} + \mathcal{D}(\mu)\mathcal{A}(\lambda)b_{\lambda,\mu} \tag{2.1.14}$$

$$b_{\lambda,\mu}\mathcal{B}(\lambda)\mathcal{C}(\mu) + c_{\lambda,\mu}\mathcal{D}(\lambda)\mathcal{A}(\mu) = \mathcal{C}(\mu)\mathcal{B}(\lambda)b_{\lambda,\mu} + \mathcal{D}(\mu)\mathcal{A}(\lambda)c_{\lambda,\mu} \tag{2.1.15}$$

$$b_{\lambda,\mu}\mathcal{B}(\lambda)\mathcal{D}(\mu) + c_{\lambda,\mu}\mathcal{D}(\lambda)\mathcal{B}(\mu) = \mathcal{D}(\mu)\mathcal{B}(\lambda)a_{\lambda,\mu} \tag{2.1.16}$$

$$a_{\lambda,\mu}\mathcal{C}(\lambda)\mathcal{C}(\mu) = \mathcal{C}(\mu)\mathcal{C}(\lambda)a_{\lambda,\mu} \tag{2.1.17}$$

$$a_{\lambda,\mu}\mathcal{C}(\lambda)\mathcal{D}(\mu) = \mathcal{C}(\mu)\mathcal{D}(\lambda)c_{\lambda,\mu} + \mathcal{D}(\mu)\mathcal{C}(\lambda)b_{\lambda,\mu} \tag{2.1.18}$$

$$a_{\lambda,\mu}\mathcal{D}(\lambda)\mathcal{C}(\mu) = \mathcal{A}(\mu)\mathcal{D}(\lambda)c_{\lambda,\mu} + \mathcal{B}(\mu)\mathcal{C}(\lambda)b_{\lambda,\mu} \tag{2.1.19}$$

$$a_{\lambda,\mu}\mathcal{D}(\lambda)\mathcal{D}(\mu) = \mathcal{D}(\mu)\mathcal{D}(\lambda)a_{\lambda,\mu} \tag{2.1.20}$$

Where the first four equations correspond to the first row, the second four to the second row, etc. Three of these equations are of particular imprtance to us.

Equation number (2.1.8) gives:

$$\begin{aligned} a(\lambda-\mu)\mathcal{B}(\lambda)\mathcal{B}(\mu) &= \mathcal{B}(\mu)\mathcal{B}(\lambda)a(\lambda-\mu) \\ \mathcal{B}(\lambda)\mathcal{B}(\mu) - \mathcal{B}(\mu)\mathcal{B}(\lambda) &= 0 \\ [\mathcal{B}(\lambda),\mathcal{B}(\mu)] &= 0 \end{aligned} \tag{2.1.21}$$

Equation number (2.1.7) gives:

$$\mathcal{A}(\mu)\mathcal{B}(\lambda)b(\lambda-\mu) + \mathcal{B}(\mu)\mathcal{A}(\lambda)c(\lambda-\mu) = a(\lambda-\mu)\mathcal{B}(\lambda)\mathcal{A}(\mu)$$

Rearranging the equation and making the substitution $\mu \to \lambda$ and $\lambda \to \mu$ gives:

$$\mathcal{A}(\lambda)\mathcal{B}(\mu) = \frac{a(\mu-\lambda)}{b(\mu-\lambda)}\mathcal{B}(\mu)\mathcal{A}(\lambda) - \frac{c(\mu-\lambda)}{b(\mu-\lambda)}\mathcal{B}(\lambda)\mathcal{A}(\mu) \tag{2.1.22}$$

Equation (2.1.12) gives:

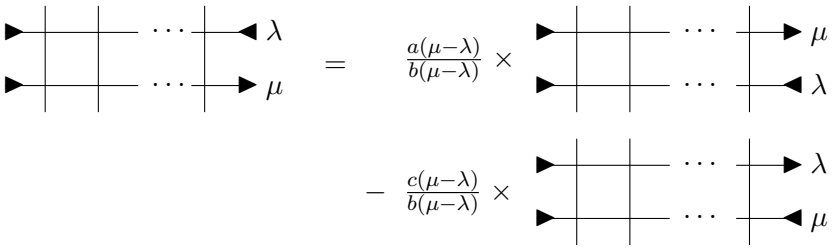
$$b(\lambda-\mu)\mathcal{D}(\lambda)\mathcal{B}(\mu)+c(\lambda-\mu)\mathcal{B}(\lambda)\mathcal{D}(\mu)=\mathcal{B}(\mu)\mathcal{D}(\lambda)a(\lambda-\mu)$$

Which after some rearranging gives:

$$\mathcal{D}(\lambda)\mathcal{B}(\mu)=\frac{a(\lambda-\mu)}{b(\lambda-\mu)}\mathcal{B}(\mu)\mathcal{D}(\lambda)-\frac{c(\lambda-\mu)}{b(\lambda-\mu)}\mathcal{B}(\lambda)\mathcal{D}(\mu) \tag{2.1.23}$$

The real importance of these equations is what they represent graphically: (2.1.21) tells us that that a product of $\mathcal{B}$ operators is invariant under an argument (i.e. λ and μ) change:

This will become important shortly. (2.1.22) tells us that we can swap an $\mathcal{A}$ and a $\mathcal{B}$ operator, but by doing so, we must multiply the result by a constant and add another lattice with the arguments changed. So $\mathcal{A}$ moves up the lattice in in both terms but the arguments are swapped in the second term. Graphically:

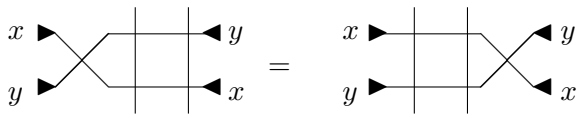


Equation (2.1.23) tells us exactly the same about the $\mathcal{D}$ operator as (2.1.22) told us about the $\mathcal{A}$ operator, only the arguments of the constants have been swapped.

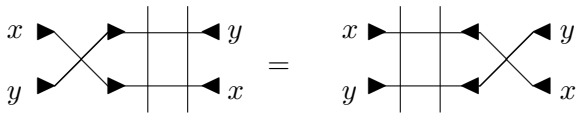
Equations (2.1.5)-(2.1.20) can also be derived graphically. This is done by considering (1.4.6):

$$R(x-y)(M(y)\otimes M(x))=(M(x)\otimes M(y))R(x-y) \tag{2.1.24}$$

If we expanded this out graphically, each matrix entry corresponds to an arrow configuration at each end. For example:



Where the arrows are the same on each end, only the R matrix has passed through. Now we sum over all combinations of bonds. This may seem like a big task, but by recognizing that the only non-zero configurations occur when the R vertex has two arrows 'in' and two arrows 'out', we only need to sum over, at most, two configurations of R. In this example, the R vertex on the LHS has two 'in' arrows, so we can 'freeze' two of them to be 'out' arrows. The RHS R vertex also has two in arrows. So we can freeze the R vertex to have to out arrows:



Rotating the R vertices 45^o *anti*-clockwise reveals them to be a-vertices, so that this particular configuration becomes:

$$\begin{aligned} a(x-y)\mathcal{B}(x)\mathcal{B}(y) =& \mathcal{B}(y)\mathcal{B}(x)a(x-y) \\ \mathcal{B}(x)\mathcal{B}(y) =& \mathcal{B}(y)\mathcal{B}(x) \end{aligned}$$

Which is equation (2.1.21). Repeating this process for all 16 configuration of outer arrows will give us (2.1.5)-(2.1.12).

2.2 Eigenvectors of T

By successive application of operators $B(\lambda)$ to the reference state, Ω, we can show that:

$$\begin{aligned} T(\lambda)\Phi &= [\mathcal{A}(\lambda) + \mathcal{D}(\lambda)]\Phi \\ &= \Lambda(\lambda)\Phi \end{aligned}$$

Where the eigenvector, Φ, of $T(\lambda)$ is:

$$\Phi(\{\lambda_j\}) = \prod_{j=1}^{l} B(\lambda_j)\Omega \tag{2.2.1}$$

and the eigenvalue, $\Lambda(\lambda)$, of $T(\lambda)$ is:

$$\Lambda(\lambda;\lambda_1,\ldots,\lambda_l) = a^N(\lambda)\prod_{j=1}^{l}\frac{a(\lambda_j-\lambda)}{b(\lambda_j-\lambda)} + b^N(\lambda)\prod_{j=1}^{l}\frac{a(\lambda-\lambda_j)}{b(\lambda-\lambda_j)}$$

The proof of this statement involves three steps:

1. Construct $\mathcal{A}(\lambda)\Phi$ and convincing ourselves that:

$$\begin{aligned} \mathcal{A}(\lambda)\Phi &= a^N(\lambda)\prod_{j=1}^{l}\frac{a(\lambda_j-\lambda)}{b(\lambda_j-\lambda)}\Phi - a^N(\lambda_k)\sum_{k=1}^{N}\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\Phi_k \\ &= M(\lambda)\Phi + \sum_{k=1}^{l} M_k(\lambda) \end{aligned}$$

   and that a similar expression exists for $\mathcal{D}(\lambda)\Phi$: $(= N(\lambda)\Phi + \sum_{k=1}^{N} N_k(\lambda))$;

2. Proving by induction that the answer we have constructed is indeed the correct one;

3. Using the 'Bethe' equation to prove that $M_k + N_k = 0$

2.2.1 Construction of $\mathcal{A}(\lambda)\Phi$

Consider the action of $\mathcal{A}(\lambda)$ on Φ.

$$A(\lambda)\Phi = A(\lambda)B(\lambda_1)B(\lambda_2)\ldots B(\lambda_l)\Omega$$

Recalling that when multiplying vertices the product goes from the bottom to the top, we can represent this product graphically as:

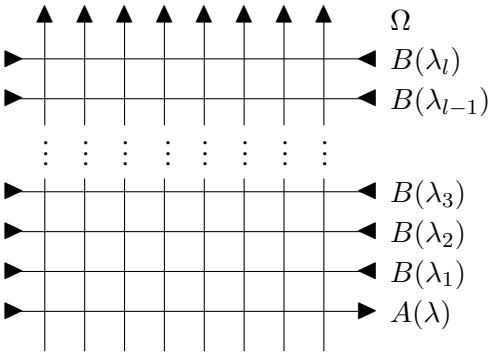
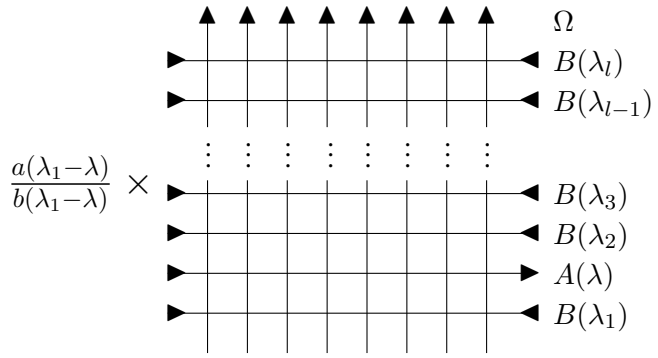
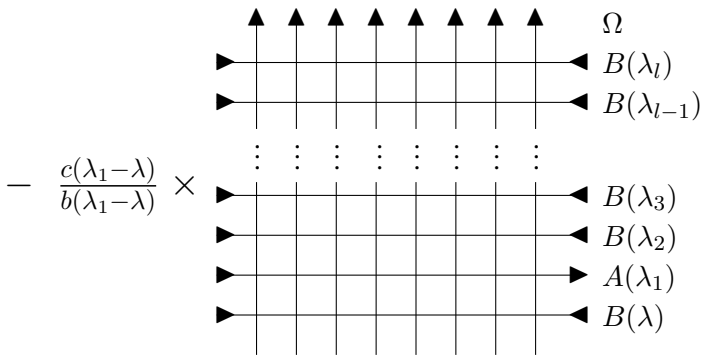


Figure 3

Using the equation (2.1.22), we can move $\mathcal{A}(\lambda)$ up one position on the lattice. Doing this then gives us two lattices, each multipled by a constant. The arguments will reman in the first term and swap in the second. Figure 3 becomes:





The aim is to apply this procedure over and over and get all $\mathcal{A}(\lambda;\lambda_j)$ to the top. At first it looks as though we will need 2^l terms. However, this is not the case; we can combine it into $l+1$ terms as follows: the first term has $A(\lambda)$ at the top, and all the $\mathcal{B}$'s preserve their arguments (λ_j); the second term λ_1 is missing from the argument of $\mathcal{B}$ and $\mathcal{A}(\lambda_1)$ is at the top; a third with the λ_2 argument missing and $\mathcal{A}(\lambda_2)$ on top, etc. Thus our $l+1$ terms are:

$$A(\lambda)\Phi = M(\lambda;\{\lambda_j\})\Phi + \sum_{k=1}^{l} M_k(\lambda;\{\lambda_j\})\Phi_k$$

Where $\{\lambda_j\}$ is some combination of $\lambda_j, j = 1, 2 \ldots l$ and

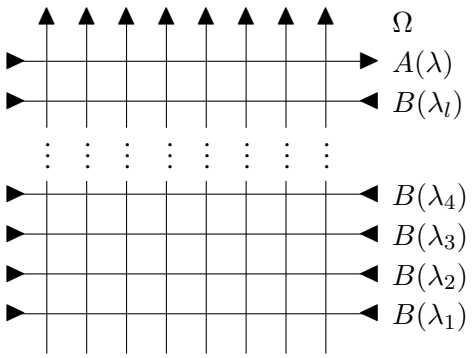
$$\Phi_k(\lambda;\{\lambda_j\}) = \prod_{j=1,\ j\neq k}^{l} B(\lambda_j)B(\lambda)\Omega \tag{2.2.2}$$

That is, Φ with a $\mathcal{B}(\lambda_j)$ replaced by $\mathcal{B}(\lambda)$, as the λ_j is in the argument of $\mathcal{A}$. Our task now is to discover the coefficients M and M_k. We can easily see that the fist coefficient, M, is obtained by continuously taking the first term of (2.1.22), essentially passing $\mathcal{A}(\lambda)$ throught the whole lattice without changing its argument. This gives:

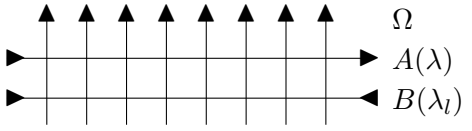
$$M = a^N(\lambda)\prod_{j=1}^{l}\frac{a(\lambda_j-\lambda)}{b(\lambda_j-\lambda)}$$

The a^N term comes from the top of the lattice as follows: When $\mathcal{A}$ finally reaches the top, regardless of its argument, it is a product with Ω. There is only one possible non-zero arrow configuration at the top, so the top row's arrows are essentially frozen into place. That is:

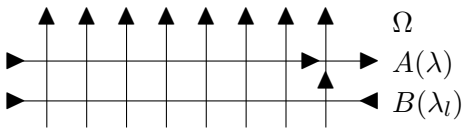
$$\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\ldots\mathcal{B}(\lambda_l)\mathcal{A}(\lambda)\Omega =$$



The top two rows look like:



Notice that the top right vertex has two out arrows; so the only non-zero configuration for it is two in arrows which will make it an a-vertex:



Now the vertex to the left of the a-vertex has two outwards arrows, so it must also be fixed as an a-vertex. It turns out that the entire top row must be a-vertices. As a lattice's weight is the product of its vertex weights, and there are N columns in the lattice, the top row must be a^N! Our lattice becomes:

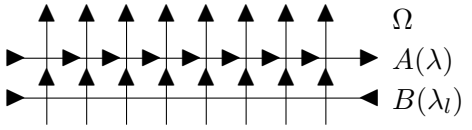
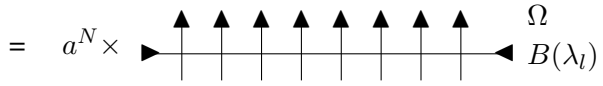


Figure 4



Which can be expressed in an equation as:

$$\mathcal{B}(\lambda)\mathcal{A}(\mu)\Omega = a^N\mathcal{B}(\lambda)\Omega$$

Hence,

$$\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\dots\mathcal{B}(\lambda_N)\mathcal{A}(\lambda)\Omega = a^N\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\dots\mathcal{B}(\lambda_N)\Omega$$

Now let's find M_1. The only way to obtain Φ_1 this is to swap the first two terms and then push $A(\lambda_1)$ all the way through, i.e. take the second term in 2.1.22 and then keep taking the first terms:

$$M_1 = -a^N(\lambda_1)\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\prod_{j=2}^{l}\frac{a(\lambda_j-\lambda_1)}{b(\lambda_j-\lambda_1)}$$

To obtain the co-efficents M_k for $k \geq 2$ it appears that we will need to combine many terms. However, instead we can use the following trick: use the commutativity of $\mathcal{B}$ to put $\mathcal{B}(\lambda_k)$ in first place (all combinations/arrangements of $\mathcal{B}(\lambda_k)$ are equivalent to this) and THEN take the second term of (2.1.22) and push $\mathcal{A}(\lambda_j)$ all the way through. If λ_k is in first place, this is the only way to obtain it. That is:

$$\mathcal{A}(\lambda)\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\dots\mathcal{B}(\lambda_N)\Omega = \mathcal{A}(\lambda)\mathcal{B}(\lambda_k)\mathcal{B}(\lambda_1)\dots\mathcal{B}(\lambda_N)\Omega$$

Because the LHS should have the same co-efficient for Φ_k as the RHS and there is one way to obtain Φ_k on the RHS:

$$M_k = -a^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=1,j\neq k}^{N}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}$$

So that:

$$A(\lambda)\Phi = a^N(\lambda)\prod_{j=1}^{l}\frac{a(\lambda_j-\lambda)}{b(\lambda_j-\lambda)}\Phi - \sum_{k=1}^{l}a^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\Phi_k \tag{2.2.3}$$

The exact same process can be applied to $\mathcal{D}(\lambda)\Phi$ to give:

$$\mathcal{D}(\lambda)\Phi = N(\lambda;\{\lambda_j\})\Phi + \sum_{k=1}^{l} N_k(\lambda;\{\lambda_j\})\Phi_k$$

Where $\{\lambda_j\}$ is some combination of $\lambda_j, j = 1, 2 \dots l$, and

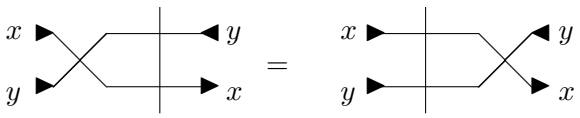
$$N = b^N(\lambda)\prod_{j=1}^{l}\frac{a(\lambda-\lambda_j)}{b(\lambda-\lambda_j)}$$

$$N_k = -b^N(\lambda_k)\frac{c(\lambda-\lambda_k)}{b(\lambda-\lambda_k)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_k-\lambda_j)}{b(\lambda_k-\lambda_j)}$$

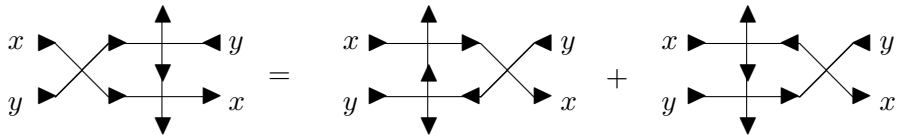
2.2.2 Proof by Induction

Although the first proof is sufficient, it can be hard to follow. To remove any doubt about it's correctness, we will prove (2.2.3) by induction.

To do this we need to make use of another of the Yang-Baxter equations. Consider second of our Yang Baxter equations (2.1.20)and the case $N = 1$ (i.e. one horizontal line so the entries of M become single vertices):



There is only one possible non-zero arrow configuration for the LHS and two for the RHS. Which freezes them to:



Giving us the equation:

$$a(y-x)b(y)c(x) = c(x)a(y)b(y-x) + b(x)c(y)c(y-x) \tag{2.2.4}$$

As these are single vertices, they are all constants ($a = \sinh(u-\lambda),\ b = \sinh(\lambda), etc$) and are therefore commutative.

Aside Some authors choose to label the vertical line as a z-axis and represent this equation using slightly different notation:

$$a(y,x)c(x,z)b(y,z) = a(y,z)c(x,z)b(y,x)b(y,x) + c(y,z)b(x,z)c(y,x)$$

where $a(x,y) = a(x-y) = a(\text{vertical axis, horizontal axis})$. However, this is not appropriate for our notayion.

Base Case: $l = 2$

We wish to show that the expansion given by (2.2.3) is correct:

$$\begin{aligned}\mathcal{A}(\lambda)\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\Omega =& a^N(\lambda)\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{a(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\Omega \\ &- a^N(\lambda_1)\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{a(\lambda_2-\lambda_1)}{b(\lambda_2-\lambda_1)}\mathcal{B}(\lambda)\mathcal{B}(\lambda_2)\Omega \\ &- a^N(\lambda_2)\frac{c(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}\frac{a(\lambda_1-\lambda_2)}{b(\lambda_1-\lambda_2)}\mathcal{B}(\lambda)\mathcal{B}(\lambda_1)\Omega\end{aligned}$$

This can be proved by expanding $\mathcal{A}(\lambda)\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)$ using (2.1.22):

$$\mathcal{A}(\lambda)\mathcal{B}(\mu) = \frac{a(\mu-\lambda)}{b(\mu-\lambda)}\mathcal{B}(\mu)\mathcal{A}(\lambda) - \frac{c(\mu-\lambda)}{b(\mu-\lambda)}\mathcal{B}(\lambda)\mathcal{A}(\mu) \tag{2.2.5}$$

So that we get:

$$\begin{aligned}\mathcal{A}(\lambda)\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2) &= [\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda_1)\mathcal{A}(\lambda) - \frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda)\mathcal{A}(\lambda_1)]\mathcal{B}(\lambda_2) \\ &= \frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda_1)\mathcal{A}(\lambda)\mathcal{B}(\lambda_2) - \frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda)\mathcal{A}(\lambda_1)\mathcal{B}(\lambda_2) \\ &= \frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda_1)[\frac{a(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}\mathcal{B}(\lambda_2)\mathcal{A}(\lambda) - \frac{c(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}\mathcal{B}(\lambda)\mathcal{A}(\lambda_2)] \\ &- \frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda)[\frac{a(\lambda_2-\lambda_1)}{b(\lambda_2-\lambda_1)}\mathcal{B}(\lambda_2)\mathcal{A}(\lambda_1) - \frac{c(\lambda_2-\lambda_1)}{b(\lambda_2-\lambda_1)}\mathcal{B}(\lambda_1)\mathcal{A}(\lambda_2)]\end{aligned}$$

Using the commutativity of $\mathcal{B}$:

$$\begin{aligned}&\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{a(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\mathcal{A}(\lambda) - \frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{c(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}\mathcal{B}(\lambda)\mathcal{B}(\lambda_1)\mathcal{A}(\lambda_2) - \\ &\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{a(\lambda_2-\lambda_1)}{b(\lambda_2-\lambda_1)}\mathcal{B}(\lambda)\mathcal{B}(\lambda_2)\mathcal{A}(\lambda_1) + \frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{c(\lambda_2-\lambda_1)}{b(\lambda_2-\lambda_1)}\mathcal{B}(\lambda)\mathcal{B}(\lambda_1)\mathcal{A}(\lambda_2)\end{aligned}$$

Collecting the second and fourth terms gives:

$$\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{a(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\mathcal{A}(\lambda)-\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{a(\lambda_2-\lambda_1)}{b(\lambda_2-\lambda_1)}\mathcal{B}(\lambda)\mathcal{B}(\lambda_2)\mathcal{A}(\lambda_1)- \\ [\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{c(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}-\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{c(\lambda_2-\lambda_1)}{b(\lambda_2-\lambda_1)}]\mathcal{B}(\lambda)\mathcal{B}(\lambda_1)\mathcal{A}(\lambda_2) \tag{2.2.6}$$

The coefficient of the third term:

$$\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{c(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}-\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{c(\lambda_2-\lambda_1)}{b(\lambda_2-\lambda_1)} \tag{2.2.7}$$

can be expanded to:

$$\frac{a(\lambda_1-\lambda)c(\lambda_2-\lambda)b(\lambda_2-\lambda_1)-c(\lambda_1-\lambda)c(\lambda_2-\lambda_1)b(\lambda_2-\lambda)}{b(\lambda_2-\lambda_1)b(\lambda_2-\lambda)b(\lambda_1-\lambda)} \tag{2.2.8}$$

At this point we must make use of some properties of our vertices: the anti-symmetry of b ($b(-x) = -b(x)$) and the fact that c is not dependent on its argument ($c(x) = c(y)$). This allows us to rearrange (2.2.8) as:

$$\begin{aligned}&\frac{-[a(\lambda_1-\lambda)c(\lambda-\lambda_2)b(\lambda_1-\lambda_2)-c(\lambda_1-\lambda)c(\lambda_1-\lambda_2)b(\lambda-\lambda_2)]}{-b(\lambda_1-\lambda_2)b(\lambda_2-\lambda)b(\lambda_1-\lambda)} \\ =&\frac{a(\lambda_1-\lambda)c(\lambda-\lambda_2)b(\lambda_1-\lambda_2)-c(\lambda_1-\lambda)c(\lambda_1-\lambda_2)b(\lambda-\lambda_2)}{b(\lambda_1-\lambda_2)b(\lambda_2-\lambda)b(\lambda_1-\lambda)}\end{aligned} \tag{2.2.9}$$

Where we have swapped both the arguments of b in the numerator, two c's in the numerator and one b in the denominator. Using our Yang-Baxter equation (2.2.4) we can rewrite the numerator of (2.2.9) as:

$$\begin{aligned}a(\lambda_1-\lambda)c(\lambda-\lambda_2)b(\lambda_1-\lambda_2)-c(\lambda_1-\lambda)c(\lambda_1-\lambda_2)b(\lambda-\lambda_2)& \\ =a(\lambda_1-\lambda_2)c(\lambda_2-\lambda)b(\lambda_1-\lambda)&\end{aligned}$$

by making the substitutions, $y - x \to \lambda_1 - \lambda$, $\lambda - \lambda_2 \to x$ and $\lambda_1 - \lambda \to y$. So

that (2.2.9) becomes:

$$\begin{aligned}&\frac{a(\lambda_1-\lambda_2)c(\lambda_2-\lambda)b(\lambda_1-\lambda)}{b(\lambda_1-\lambda_2)b(\lambda_2-\lambda)b(\lambda_1-\lambda)}\\=&\frac{a(\lambda_1-\lambda_2)c(\lambda_2-\lambda)}{b(\lambda_1-\lambda_2)b(\lambda_2-\lambda)}\end{aligned}$$

Subbing this back into (2.2.6) gives:

$$\begin{aligned}\mathcal{A}(\lambda)\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2) =&\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{a(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\mathcal{A}(\lambda)\\&-\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{a(\lambda_2-\lambda_1)}{b(\lambda_2-\lambda_1)}\mathcal{B}(\lambda)\mathcal{B}(\lambda_2)\mathcal{A}(\lambda_1)\\&-\frac{a(\lambda_1-\lambda_2)c(\lambda_2-\lambda)}{b(\lambda_1-\lambda_2)b(\lambda_2-\lambda)}\mathcal{B}(\lambda)\mathcal{B}(\lambda_1)\mathcal{A}(\lambda_2)\end{aligned}$$

And:

$$\begin{aligned}\mathcal{A}(\lambda)\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\Omega =&a^N(\lambda)\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{a(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\Omega\\&-a^N(\lambda_1)\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{a(\lambda_2-\lambda_1)}{b(\lambda_2-\lambda_1)}\mathcal{B}(\lambda)\mathcal{B}(\lambda_2)\Omega\\&-a^N(\lambda_2)\frac{c(\lambda_2-\lambda)}{b(\lambda_2-\lambda)}\frac{a(\lambda_1-\lambda_2)}{b(\lambda_1-\lambda_2)}\mathcal{B}(\lambda)\mathcal{B}(\lambda_1)\Omega\end{aligned}$$

As required. This proves the base case.

Inductive Step

Assume the following is true for l:

$$\mathcal{A}(\lambda)\Phi = a^N(\lambda)\prod_{j=1}^{l}\frac{a(\lambda_j-\lambda)}{b(\lambda_j-\lambda)}\Phi+\sum_{k=1}^{l}a^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\Phi_k$$

Now we'll try and show it's true for $l+1$. Using (2.2.5):

$$\begin{aligned}&\mathcal{A}(\lambda)\Phi = \mathcal{A}(\lambda)\mathcal{B}(\lambda_1)\mathcal{B}(\lambda_2)\dots\mathcal{B}(\lambda_{l+1}) = \mathcal{A}(\lambda)\mathcal{B}(\lambda_1)\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\\&=[\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda_1)\mathcal{A}(\lambda)-\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda)\mathcal{A}(\lambda_1)]\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\\&=\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda_1)\times\mathcal{A}(\lambda)\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)-\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda)\times\mathcal{A}(\lambda_1)\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\end{aligned}\tag{2.2.10}$$

Using the inductive step to evaluate $\mathcal{A}(\lambda)\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)$:

$$\begin{aligned}\mathcal{A}(\lambda)\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega =& a^N(\lambda)\prod_{j=2}^{l+1}\frac{a(\lambda_j-\lambda)}{b(\lambda_j-\lambda)}\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega \\ &+\sum_{k=2}^{l+1}a^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=2,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega\end{aligned}$$

Note that $\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)$ is equivalent to $\prod_{i=1}^{l}\mathcal{B}(\lambda_i)$ by the substitution $j-1=i$. Using exactly the same process we can evaluate $\mathcal{A}(\lambda_1)\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)$:

$$\begin{aligned}\mathcal{A}(\lambda_1)\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega =& a^N(\lambda_1)\prod_{j=2}^{l+1}\frac{a(\lambda_j-\lambda_1)}{b(\lambda_j-\lambda_1)}\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega \\ &+\sum_{k=2}^{l+1}a^N(\lambda_k)\frac{c(\lambda_k-\lambda_1)}{b(\lambda_k-\lambda_1)}\prod_{j=2,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega\end{aligned}$$

Subbing these into (2.2.10) and expanding gives:

$$\begin{aligned}\mathcal{A}(\lambda)\mathcal{B}(\lambda_1)\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega =& \\ &\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda_1)\times a^N(\lambda)\prod_{j=2}^{l+1}\frac{a(\lambda_j-\lambda)}{b(\lambda_j-\lambda)}\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega \\ &-\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda_1)\times\sum_{k=2}^{l+1}a^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=2,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega \\ &-\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda)\times a^N(\lambda_1)\prod_{j=2}^{l+1}\frac{a(\lambda_j-\lambda_1)}{b(\lambda_j-\lambda_1)}\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega \\ &+\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\mathcal{B}(\lambda)\times\sum_{k=2}^{l+1}a^N(\lambda_k)\frac{c(\lambda_k-\lambda_1)}{b(\lambda_k-\lambda_1)}\prod_{j=2,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega\end{aligned}$$

Using the commutativity of $\mathcal{B}$, moving constants into the sum and collecting the second and fourth term gives:

$$\mathcal{A}(\lambda)\mathcal{B}(\lambda_1)\prod_{j=2}^{l+1}\mathcal{B}(\lambda_j)\Omega = a^N(\lambda)\prod_{j=1}^{l+1}\frac{a(\lambda_j-\lambda)}{b(\lambda_j-\lambda)}\prod_{j=1}^{l+1}\mathcal{B}(\lambda_j)\Omega$$
$$- a^N(\lambda_1)\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\prod_{j=2}^{l+1}\frac{a(\lambda_j-\lambda_1)}{b(\lambda_j-\lambda_1)}\Phi_1$$
$$- \sum_{k=2}^{l+1}a^N(\lambda_k)[\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)} - \frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{c(\lambda_k-\lambda_1)}{b(\lambda_k-\lambda_1)}]$$
$$\times \prod_{j=2,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\prod_{j=2}^{l+1}\mathcal{B}(\lambda)\mathcal{B}(\lambda_j)\Omega \qquad (2.2.11)$$

Using the Yang-Baxter equation (2.2.4) the coefficient of the third term can be simplified exactly the same way as we simplified (2.2.7) by letting $\lambda_2 \to \lambda_k$, giving:

$$\frac{a(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)} - \frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\frac{c(\lambda_k-\lambda_1)}{b(\lambda_k-\lambda_1)} = \frac{a(\lambda_1-\lambda_k)}{b(\lambda_1-\lambda_k)}\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}$$

The third term of (2.2.11) becomes:

$$\sum_{k=2}^{l+1}a^N(\lambda_k)\frac{a(\lambda_1-\lambda_k)}{b(\lambda_1-\lambda_k)}\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=2,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\Phi_k$$
$$= \sum_{k=2}^{N+1}a^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\Phi_k$$

So that (2.2.11) becomes:

$$\mathcal{A}(\lambda)\prod_{j=1}^{l+1}\mathcal{B}(\lambda_j)\Omega = a^N(\lambda)\prod_{j=1}^{l+1}\frac{a(\lambda_j-\lambda)}{b(\lambda_j-\lambda)}\prod_{j=1}^{l+1}\mathcal{B}(\lambda_j)\Omega$$
$$- a^N(\lambda_1)\frac{c(\lambda_1-\lambda)}{b(\lambda_1-\lambda)}\prod_{j=2}^{l+1}\frac{a(\lambda_j-\lambda_1)}{b(\lambda_j-\lambda_1)}\Phi_1$$
$$- \sum_{k=2}^{l+1}a^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\Phi_k$$

Noting that the second term is equivalent to a $k = 1$ term in the sum, we can simplify this to:

$$\mathcal{A}(\lambda)\prod_{j=1}^{l+1}\mathcal{B}(\lambda_j) = a^N(\lambda)\prod_{j=1}^{l+1}\frac{a(\lambda_j-\lambda)}{b(\lambda_j-\lambda)}\Phi$$
$$-\sum_{k=1}^{l+1}a^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)}\Phi_k$$

As required. This completes the proof by induction. A similar procedure can prove $\mathcal{D}(\lambda)\Phi = (N+N_k)\Phi$.

2.2.3 Bethe Equations

At this point we have

$$T(\lambda)\Phi = (\mathcal{A}+\mathcal{D})\Phi \tag{2.2.12}$$
$$= (M+N)\Phi + \sum_{k=1}^{l}(M_k+N_k)\Phi_k \tag{2.2.13}$$

To make the above an eigenvector equation, we wish for the second term to become zero. Consider

$$M_k+N_k = -a^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)} - b^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda-\lambda_k)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_k-\lambda_j)}{b(\lambda_k-\lambda_j)}$$

Because $b(\lambda)$ is antisymmetric ($\sinh(-x) = -\sinh(x)$ or $b(x-y) = -b(y-x)$), and c is independent of λ, then $\frac{c(\lambda-\lambda_k)}{b(\lambda-\lambda_k)} = \frac{-c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}$ allows us to rewrite the coefficient of the second product:

$$M_k+N_k = -a^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)} + b^N(\lambda_k)\frac{c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_k-\lambda_j)}{b(\lambda_k-\lambda_j)}$$
$$= \frac{-c(\lambda_k-\lambda)}{b(\lambda_k-\lambda)}(a^N(\lambda_k)\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_k)} - b^N(\lambda_k)\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_k-\lambda_j)}{b(\lambda_k-\lambda_j)}) \tag{2.2.14}$$

The 'Bethe Equation' can be expressed [2]:

$$\frac{a^N(\lambda_k)}{b^N(\lambda_k)} = \prod_{j=1,j\neq k}^{l}\frac{b(\lambda_j-\lambda_k)}{a(\lambda_j-\lambda_k)}\times\frac{a(\lambda_k-\lambda_j)}{b(\lambda_k-\lambda_j)} \qquad k=1,2,3,\ldots,l.$$

[2] Takhatajan, L. J. Introduction to Agebraic Bethe Ansatz, *Lecture Notes in Physics, Exactly solvable problems in condensed matter and relativistic field theory*, 187

Or,

$$a^N(\lambda_k)\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_j-\lambda_k)}{b(\lambda_j-\lambda_j)}=b^N(\lambda_k)\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_k-\lambda_j)}{b(\lambda_k-\lambda_j)}$$

Which we can substitute into (2.2.14) to get

$$\begin{aligned}M_k+N_k=&\frac{-c\lambda_k-\lambda)}{b(\lambda_k-\lambda)}(b^N(\lambda_k)\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_k-\lambda_j)}{b(\lambda_k-\lambda_j)}-b^N(\lambda_k)\prod_{j=1,j\neq k}^{l}\frac{a(\lambda_k-\lambda_j)}{b(\lambda_k-\lambda_j)})\\=&0\end{aligned}$$

We have just proved:

$$M_k(\lambda,\{\lambda_j\})+N_k(\lambda,\{\lambda_j\})=0$$

for $k=1,2,\ldots,l$

Using this we can simplify equation (2.2.13)

$$T(\lambda)\Phi=(M(\lambda)+N(\lambda))\Phi \tag{2.2.15}$$

Provided that the horizontal rapidity variables $\{\mu_j\}$ satisfy the Bethe equations for all $j=1,\ldots,l$ the eigenvalues of the transfer matrix are given by $M+N=\Lambda(\lambda,\{\lambda_j\})$, and the vector $\Phi(\{\lambda_j\})$, is really the eigenvector of $T(\lambda)$.

2.3 Diagonalization of T

At this point we are able to diagonalize T using elementary linear algebra techniques. We can create a diagonal matrix D, consisting of the eigenvalues of T ($\Lambda(\lambda,\{\lambda_j\})$) along the diagonal, and a matrix P, with the eigenvectors of T ($\Phi(\{\lambda_j\})$) as columns.

$$D=P^{-1}TP$$

Then we can easily compute T^M as follows:

$$
\begin{aligned}
T^M &= (PDP^{-1})^M \\
&= PDP^{-1}PDP^{-1}PDP^{-1}\dots PDP^{-1} \\
&= PD(I)D(I)D(I)\dots DP^{-1} \\
&= PD^MP^{-1}
\end{aligned}
$$

given that powers of diagonal matrixces are simple to calculate, we have just compted T^M which was our aim! "This procedure for the diagonalization of commuting familiy of transfer matrices is called Algebraic Bethe Ansatz (in the simplest abc-form)."[3]

Now that we have the partition function Z, we can go ahead and compute various physical quantities, such as the bulk correlation functions. Bulk correlation functions are beyond the scope of this thesis. Instead, we will consider a simpler application: we will use the elements of the monodromy matrix M to give an explicit expression, in terms of determinants, for the boundary 1-point function, in the presence of *domain wall boundary conditions*.

[3]Takhatajan, L. J. Introduction to Agebraic Bethe Ansatz, *Lecture Notes in Physics, Exactly solvable problems in condensed matter and relativistic field theory*, 190

Chapter 3

Boundary Correlation Functions of the Six Vertex-Model

3.1 New Notation

The field of vertex models is widely researched and many papers are published in this area. Accordingly, a variety of notations are used. For the rest of this thesis we will use different notation to what we have used in the past two chapters. The changes are easily incorporated into all previous working, and also give a broader insight into this field.

3.1.1 Spaces

So far we have been working in the spaces $\mathcal{V}$ and h_n. We will now represent these as:

$$\omega_n = \begin{pmatrix} 1 \\ 0 \end{pmatrix}_n = |\uparrow\rangle_n$$
$$\Omega = \prod_{k=1}^{N} \otimes w_n = \prod_{n=1}^{N} \otimes |\uparrow\rangle = |\Uparrow\rangle$$

This new notation is graphically much clearer. A single up arrow in column n is simply represented by an arrow! N up arrows are represented by a solid arrow.

The rest of the spaces discussed in section 2.1 become:

$\langle\uparrow|_\alpha \in \mathcal{V}$ Defines an right arrow on the α-th left edge

$\langle\downarrow|_\alpha \in \mathcal{V}$ Defines a left arrow on the α-th left edge

$\langle\uparrow|_\kappa \in \mathcal{H}$ Defines an up arrow on the β-th bottom edge

$\langle\downarrow|_\kappa \in \mathcal{H}$ Defines a down arrow on the β-th bottom edge

$|\uparrow\rangle_\alpha \in \mathcal{V}$ Defines an right arrow on the α-th right edge

$|\downarrow\rangle_\alpha \in \mathcal{V}$ Defines a left arrow on the α-th right edge

$|\uparrow\rangle_\kappa \in \mathcal{H}$ Defines an up arrow on the β-th top edge

$|\downarrow\rangle_\kappa \in \mathcal{H}$ Defines a down arrow on the β-th top edge

Where $\mathcal{H}$ is the space of horizontal pointing arrows.

3.1.2 Vertices

We will now give a different parameterisation of the vertex weights, a, b and c from those in (2.1.1) - (2.1.3). Let $\lambda \to \eta - \lambda$ and $u \to 2\eta$. This gives:

$$\begin{aligned} a(\lambda) &= \sinh(\lambda + \eta) \\ -b(\lambda) &= \sinh(\lambda - \eta) \\ c(\lambda) &= \sinh(2\eta) \end{aligned}$$

We can then let $b(\lambda) \to -b(\lambda)$ so that our final parameterisation is:

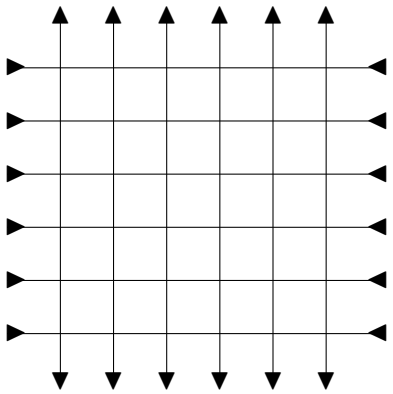
$$\begin{aligned} a(\lambda) &= \sinh(\lambda + \eta) \\ b(\lambda) &= \sinh(\lambda - \eta) \\ c(\lambda) &= \sinh(2\eta) \end{aligned}$$

Note that $c(\lambda)$ remains independent of λ. It should also be noted that all partition functions must have an even number of b vertices. This is why the partition function is invariant under a change of sign to the b-vertex.

3.1.3 Boundary Conditions

Although not strictly a change in notation, we will work with new boundary conditions. We will now consider an $N \times N$ lattice rather than $M \times N$, and we will also

use Domain Wall Boundary Conditions (DWBC) rather than require periodicity in both directions. Domain wall boundary conditions have horizontal arrows facing in, and vertical arrow pointing out:



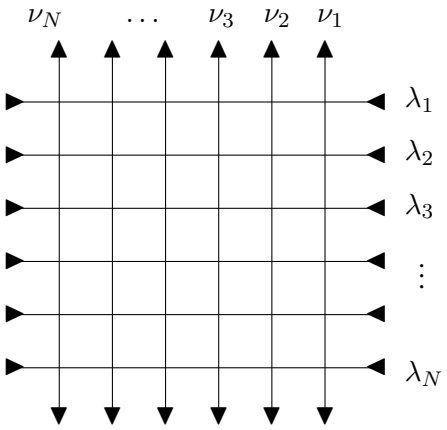
A rotation of $\frac{\pi}{2}$, or a reversion of all arrows, still satisfies the DWBC. Though, we will only work in the above case.

3.2 The Inhomogeneous Case

So far we have we have been working in the homogenous case. We are now in a position to appreciate the significance of the differences between homogeneous and inhomogeneous cases. Let's go over the main differences.

3.2.1 The Lattice

Consider our new lattice, and label it as follows:



Each site in the lattice can be weight a, b or c (in the 6-vertex case). In the inhomogenous case we assumed that all of these sites were equivalent. However, in the inhomogenous case, each vertex weight is dependent on its site. So an a vertex on the 4^{th} row and 5^{th} column can have a different weight to an a vertex on the 6^{th} row and 9^{th} column. Thus, a vertex at site $(\lambda_\alpha, \nu_\kappa)$ must depend these two parameters. We can then represent the vertex weights as:

$$a(\lambda_\alpha, \nu_\kappa) \quad b(\lambda_\alpha, \nu_\kappa) \quad c(\lambda_\alpha, \nu_\kappa)$$

Specifying different values of λ_α and ν_κ for each row and column allows every vertex to be assigned a different weight.

3.2.2 The M and L Operators and the spaces $\mathcal{V}$ and $\mathcal{H}$

From the beginning of this thesis, we worked with L_n, where the n represented the horizontal space that L_n operated in. We ignored the effect of λ_α and treated all vertical spaces as equivalent. In the inhomogeneous case, we want an L-matrix that operates in the entire space $\mathcal{V} \otimes \mathcal{H}$, meaning it can specify the value of any vertex. This would require a dependance on both α and κ, where α defines the single vertical space that $L_{\alpha\beta}$ operates and κ defines the horizontal space. This would mean that the entries of $\lambda_{\alpha\beta}$, the vertex weights, would have to be dependent on both λ_α and ν_κ.

If we were to go back to the beginning of Chapter 1 and consider the inhomogeneous case (and work with our original boundary conditions), our working would be almost identical. This is because our two most important operators; M and L_n

only operated in the vertical space, $\mathcal{V}$ (as it was their vertical arrows that needed to be specified). This independence of λ_α essentially allows us to treat it as a constant. Thus, in the case where Boltzmann weights are dependent on repidities, which are not necessarily independent variables, we let $\lambda_\alpha - \nu_\kappa \to \lambda$, and we have the homogeneous case, which we have already solved! Thus our vertex weights must be:

$$a(\lambda_\alpha, \nu_\kappa) = \sinh(\lambda_\alpha - \nu_\kappa + \eta) \tag{3.2.1}$$

$$b(\lambda_\alpha, \nu_\kappa) = \sinh(\lambda_\alpha - \nu_\kappa - \eta) \tag{3.2.2}$$

$$c(\lambda_\alpha, \nu_\kappa) = \sinh(2\eta) \tag{3.2.3}$$

And the L matrix becomes:

$$L_{\alpha,\kappa}(\lambda_\alpha, \nu_\kappa) = \begin{bmatrix} a(\lambda_\alpha, \nu_\kappa) & 0 & 0 & 0 \\ 0 & b(\lambda_\alpha, \nu_\kappa) & c(\lambda_\alpha, \nu_\kappa) & 0 \\ 0 & c(\lambda_\alpha, \nu_\kappa) & b(\lambda_\alpha, \nu_\kappa) & 0 \\ 0 & 0 & 0 & a(\lambda_\alpha, \nu_\kappa) \end{bmatrix}_{\alpha,\kappa}$$

An alternative representation of $L_{\alpha,\kappa}$ uses 'Pauli matrices', σ^j:

$$L_{\alpha,\kappa}(\lambda_\alpha, \nu_\kappa) = \begin{bmatrix} \sinh(\lambda_\alpha - \nu_\kappa + \eta\sigma^3_\kappa) & \sigma^-_\kappa \sinh(2\eta) \\ \sigma^-_\kappa \sinh(2\eta) & \sinh(\lambda_\alpha - \nu_\kappa - \eta\sigma^3_\kappa) \end{bmatrix}_\alpha$$

Where the Pauli matrices are:

$$\sigma^1_\kappa = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}_\kappa \quad \sigma^2_\kappa = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}_\kappa \quad \sigma^3_\kappa = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}_\kappa$$

And

$$\sigma^+_\kappa = \frac{1}{2}(\sigma^1_\kappa + i\sigma^2_\kappa) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}_\kappa$$

$$\sigma^-_\kappa = \frac{1}{2}(\sigma^1_\kappa - i\sigma^2_\kappa) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}_\kappa$$

When σ occurs inside the argument of $\sinh$ we treat it as follows:

$$\sinh(\lambda_\alpha - \nu_\kappa + \eta\sigma^3_\kappa) = \begin{pmatrix} \sinh(\lambda_\alpha - \nu_\kappa + \eta) & 0 \\ 0 & \sinh(\lambda_\alpha - \nu_\kappa - \eta) \end{pmatrix}$$

When squared the Pauli martices all become the 2×2 identity matrix:

$$(\sigma^1)^2 = (\sigma^2)^2 = (\sigma^3)^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbb{I}$$

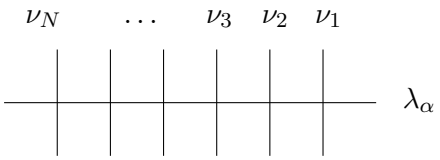
Note: it is standard notation to represent Pauli matrices with the numbers $1, 2, 3$ as a subscript. However, we have assigned the subscript, κ, to denote the space in which the L matrix acts. The reason for this will soon become clear. Some authors will also occasionally replace the Pauli matrix number $1, 2, 3$ with x, y, z.

■

In the inhomogenous case the monodromy matrix, $M_\alpha(\lambda_\alpha)$, becomes:

$$M_\alpha(\lambda_\alpha) = L_{\alpha N}(\lambda_\alpha, \nu_N) \dots L_{\alpha 1}(\lambda_\alpha, \nu_1) = \begin{bmatrix} \mathcal{A}(\lambda_\alpha) & \mathcal{B}(\lambda_\alpha) \\ \mathcal{C}(\lambda_\alpha) & \mathcal{D}(\lambda_\alpha) \end{bmatrix}_\alpha \tag{3.2.4}$$

The reason that M is only dependent on λ_α and not ν_κ is that M lies in $\mathcal{V}$. We can represent this graphically as:



3.2.3 The Eigenvalues

The eigenvalues given in (2.2.15) are slightly different in the inhomogeneous case. Recall in Figure 4 we treated $a(\lambda)$ as a constant, and obtained the eigenvalues of $\mathcal{A} \prod_{j=1}^{l} \mathcal{B}(\lambda_j) | \Uparrow \rangle$. Part of this involved showing:

$$\mathcal{A}(\lambda) | \Uparrow \rangle = a^N(\lambda) | \Uparrow \rangle$$

In the inhomogeneous case we are unable to do this, as each term in the product of a vertices is dependent on (ν_k). Thus:

$$\mathcal{A}(\lambda) | \Uparrow \rangle = \prod_{k=1}^{N} \sinh(\lambda - \nu_k + \eta) | \Uparrow \rangle \tag{3.2.5}$$

Similarly:

$$\mathcal{D}(\lambda)|\Uparrow\rangle = \prod_{k=1}^{N} \sinh(\lambda - \nu_k - \eta)|\Uparrow\rangle \tag{3.2.6}$$

We can think of (3.2.5) and (3.2.6) as giving the eigenvalues of $\mathcal{A}$ and $\mathcal{D}$ with eigenfunction $|\Uparrow\rangle$. The eigenvalues of $T(\lambda)$ become:

$$\begin{aligned}\Lambda(\lambda;\{\lambda_j\}_{j=1}^{j=l}) = &\prod_{k=1}^{N} \sinh(\lambda - \nu_k + \eta) \prod_{j=1}^{l} \frac{a(\lambda_j - \lambda)}{b(\lambda_j - \lambda)} \\ &+ \prod_{k=1}^{N} \sinh(\lambda - \nu_k - \eta) \prod_{j=1}^{l} \frac{a(\lambda - \lambda_j)}{b(\lambda - \lambda_j)}\end{aligned}$$

where

$$T(\lambda)\Phi = [\mathcal{A}(\lambda) + \mathcal{D}(\lambda)]\Phi = \Lambda(\lambda)\Phi$$

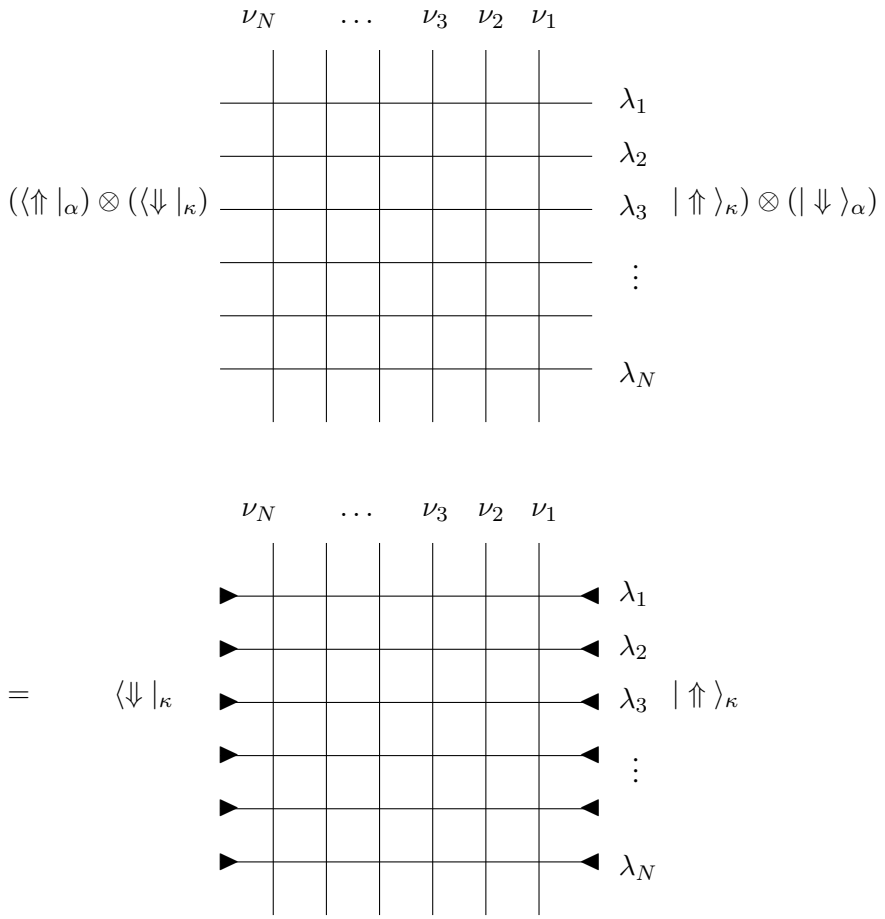
Unfortunately, T^M does not represent the partition function with DWBC, and the transfer matrix is of no help in calculating the partition function with such boundary conditions. We need to represent the partition function in an alternative way.

3.2.4 The Partition Function

Using our new notation we can represent the partition function with DWBC as:

$$Z_N = (\prod_{\alpha=1}^{N} \langle\uparrow|_\alpha) \otimes (\prod_{\kappa=1}^{N} \langle\downarrow|_\kappa) M_N(\lambda_N) \dots M_N(\lambda_1) (\prod_{\kappa=1}^{N} |\uparrow\rangle_\kappa) \otimes (\prod_{\alpha=1}^{N} |\downarrow\rangle_\alpha)$$

The products over α places 'in' arrows on the left and right boundaries of the lattice and the products over β place 'out' arrows on the top and bottom of the lattice. Z_N is the sum over all configurations with these boundary conditions. We can represent Z_N:



or as:

$$Z_N = \langle\Downarrow|\mathcal{B}(\lambda_N)\dots\mathcal{B}(\lambda_1)|\Uparrow\rangle$$

We can swap the order that $M_N(\lambda_N)\dots M_N(\lambda_1)$ acts on the spaces $\mathcal{H}$ and $\mathcal{V}$. This is why we could act on $\langle\Downarrow|_\alpha$ and $|\Uparrow\rangle_\alpha$ before $\langle\Downarrow|_\kappa$. In addition to the above representation of the partition function, we will also use the following representation:[1]

$$Z_N = \frac{\prod_{\alpha=1}^{N}\prod_{\kappa=1}^{N}\sinh(\lambda_\alpha-\nu_\kappa+\eta)\sinh(\lambda_\alpha-\nu_\kappa-\eta)}{\prod_{1\le\alpha<\beta\le N}\sinh(\lambda_\beta-\lambda_\alpha)\prod_{1\le\kappa<j\le N}\sinh(\nu_\kappa-\nu_j)}\det_N\mathcal{Z} \tag{3.2.7}$$

[1]We will not prove this, but a proof is provided in: Izergin A G 1987 Partition function of the 6-vertex model in the finite volume, *Sov. Phys. Dokl.* 32

Where the entries of matrix $\mathcal{Z}$ are:

$$\mathcal{Z}_{\alpha\kappa} = \phi(\lambda_\alpha, \nu_\kappa), \qquad \alpha, \kappa = 1 \dots N \tag{3.2.8}$$

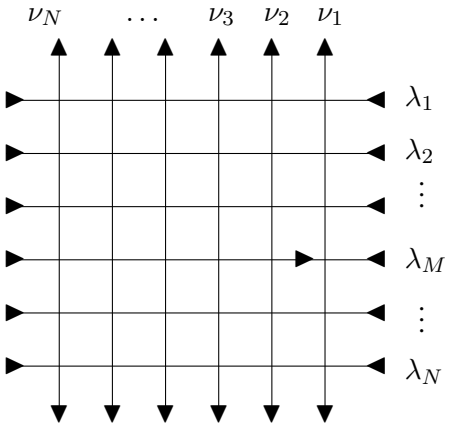
and the function $\phi(\lambda_\alpha, \nu_\kappa)$ is defined:

$$\phi(\lambda, \nu) = \frac{c(\lambda, \nu)}{a(\lambda, \nu) b(\lambda, \nu)} = \frac{\sinh(2\eta)}{\sinh(\lambda - \nu + \eta)\sinh(\lambda - \nu - \eta)} \tag{3.2.9}$$

3.3 Boundary Correlation Functions

3.3.1 $H_N^{(M)}$ and $G_N^{(M)}$

We will consider two types of boundary correlation functions in this paper. The first, $H_N^{(M)}$, is a function describing the weighted probability that an arrow on the M-th row and first column will be inverted or inwards pointing. The inverted arrow is also known as a "boundary spontaneous polarization"[2]. We can represent this graphically as:



The second type of boundary correlation function, $G_N^{(M)}$ describes the 'weighted probability' of an inverted arrow on, or before, the M-th row. Any arrow from row $1 \dots M$ may be inverted.

The 'weighted probability' does not refer to the number of lattices with an inverted arrow divided by the total number of non-zero lattices, as the term 'probability'

[2]Bogoliubov, N. M.; Pronko, A. G.; Zvonarev, M. B, Boundary correlation functions of the six-vertex model, 6

would suggest. Instead. it represents the $weight$ of all lattices with DWBC and an inverted arrow on the M-th row (or first to M-th for $G_N^{(M)}$), divided by the $weights$ of all lattices with or without the inverted arrow (i.e. the partition function Z_N). We can represent these functions:

$$H_N^{(M)} = \frac{1}{Z_N}\langle \Downarrow |\mathcal{B}(\lambda_N)\dots\mathcal{B}(\lambda_{M+1})q_1\mathcal{B}(\lambda_M)p_1\mathcal{B}(\lambda_{M-1})\dots\mathcal{B}(\lambda_1)| \Uparrow \rangle \tag{3.3.1}$$

$$G_N^{(M)} = \frac{1}{Z_N}\langle \Downarrow |\mathcal{B}(\lambda_N)\dots\mathcal{B}(\lambda_{M+1})q_1\mathcal{B}(\lambda_M)p_1\dots\mathcal{B}(\lambda_1)| \Uparrow \rangle \tag{3.3.2}$$

For now, we need to see that in $H_N^{(M)}$, the q_1 and p_1 have the effect of ensuring that that the inverted arrow occurs on the M-th row of the lattice, that is $\mathcal{B}(\lambda_M)$. In $G_N^{(M)}$, the q_1 ensures that the inverted arrow will occur on the right of q_1, but does not 'trap' the $\mathcal{B}(\lambda_M)$ term. Instead the commutativity of $\mathcal{B}(\lambda_j)$ means that the inverted arrow could occur on any of the rows from $1\dots M$, as required.

Before we calculate the the boundary correlation functions, we will make two observations.

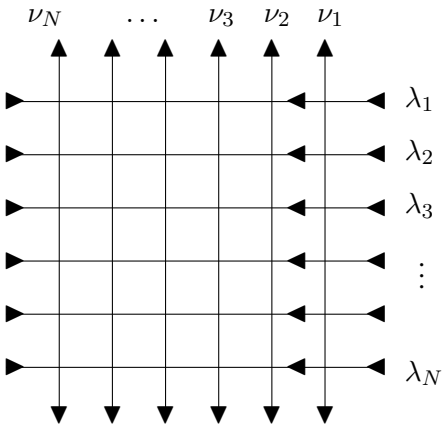
Firstly, we can quite easily see from the definition of $G_N^{(M)}$ and $H_N^{(M)}$ that:

$$H_N^{(M)} = G_N^{(M)} - G_N^{(M-1)}$$

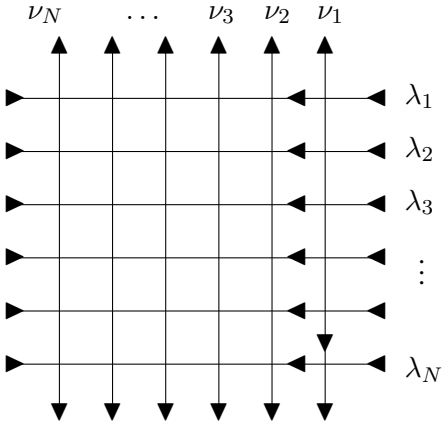
Secondly, we will show later that:

$$G_N^{(M)} = H_N^{(M)} + H_N^{(M-1)} + \dots H_N^{(1)} \tag{3.3.3}$$

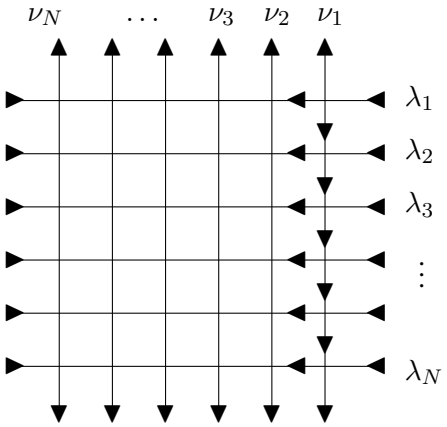
Thirdly, the probability that at least one arrow in the first column will be inverted, is one. That is, $G_N^{(N)} = 1$. To prove this we will consider what happens when *no* arrows are inverted:



The bottom right vertex has two in arrows and one out arrow. We freeze the surrounding arrows as follows:



We continue freezing arrows until we get:



The top right vertex has been frozen to have three out arrows and therefore has a weight of zero. As the partition function is equal to the product of the vertex weights, one zero weight sends the entire weight to zero. This means that no lattice exists with DWBC and no inverted arrows in the first column. Hence, $G_N^{(N)} = 1$.

3.3.2 Partitioning of Monodromy Matrix

Let's write the monodromy matrix in the following form:

$$M_\alpha(\lambda_\alpha) = M_{\alpha 2}(\lambda_\alpha) M_{\alpha 1}(\lambda_\alpha)$$

Where

$$M_{\alpha 2}(\lambda_\alpha) = L_{\alpha N}(\lambda_\alpha, \nu_N) \dots L_{\alpha N}(\lambda_\alpha, \nu_2) = \begin{pmatrix} \mathcal{A}_2(\lambda_\alpha) & \mathcal{B}_2(\lambda_\alpha) \\ \mathcal{D}_2(\lambda_\alpha) & \mathcal{D}_2(\lambda_\alpha) \end{pmatrix} \tag{3.3.4}$$

$$M_{\alpha 1}(\lambda_\alpha) = L_{\alpha 1}(\lambda_\alpha, \nu_1) = \begin{pmatrix} \mathcal{A}_1(\lambda_\alpha) & \mathcal{B}_1(\lambda_\alpha) \\ \mathcal{D}_1(\lambda_\alpha) & \mathcal{D}_1(\lambda_\alpha) \end{pmatrix} \tag{3.3.5}$$

The $M_{\alpha 1}(\lambda_\alpha)$ acts in the first vertical space $|\Uparrow_1\rangle \equiv |\uparrow\rangle_1$, and $M_{\alpha 2}(\lambda_\alpha)$ acts in the rest of the $N-1$ vertical spaces,$|\Uparrow_2\rangle \equiv \prod_{j=2}^{N-1} \otimes |\uparrow\rangle_j$. Where

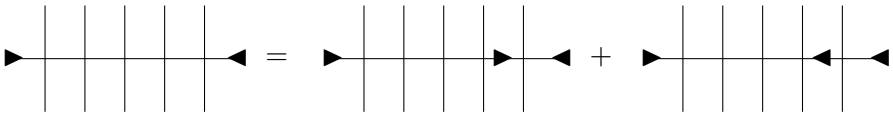
$$|\Uparrow\rangle = |\Uparrow_2\rangle \otimes |\Uparrow_1\rangle \tag{3.3.6}$$

The entries of $M_{\alpha 1}(\lambda_\alpha)$ correspond to the entries of the $L_{\alpha 1}(\lambda_\alpha, \nu_1)$ given in (3.2.4).

Now that we have decomposed M, let's decompose $\mathcal{B}$:

$$\mathcal{B}(\lambda) = \mathcal{A}_2(\lambda)\mathcal{B}_1(\lambda) + \mathcal{B}_2(\lambda)\mathcal{D}_1(\lambda) \tag{3.3.7}$$

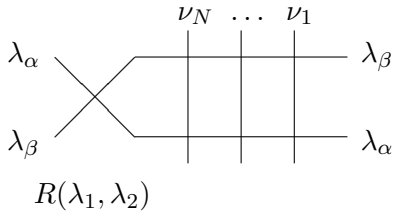
This can be obtained from matrix multiplication of equations (3.3.5) and (3.3.4). It is also quite clear when considered graphically:



3.3.3 Inhomogenous Yang-Baxter Equation and the Quantum R-Matrix

It is convenient to use new notation for the the Yang-Baxter equation. If we do not, the formulas we are about to derive will not have the same simplicity. Consider the Yang-Baxter equation, and express it:

$$R_{\alpha\beta}(\lambda_\alpha,\lambda_\beta)L_{\alpha\kappa}(\lambda_\alpha,\lambda_\kappa)L_{\beta\kappa}(\lambda_\beta,\lambda_\kappa) = L_{\beta\kappa}(\lambda_\beta,\lambda_\kappa)L_{\alpha\kappa}(\lambda_\alpha,\lambda_\kappa)R_{\alpha\beta}(\lambda_\alpha,\lambda_\beta)$$



$R(\lambda_1,\lambda_2)$

This is easily obtained from the Yang-Baxter equation by making the substitution: $\lambda \to \lambda_\alpha - \lambda_\kappa$ and $\lambda' \to \lambda_\beta - \lambda_\kappa$. Note that the R matrix acts in the space $h_\alpha \otimes h_\beta$. We can represent the inhomogenous R-matrix:

$$\begin{bmatrix} a(\lambda_\alpha,\nu_\beta) & 0 & 0 & 0 \\ 0 & c(\lambda_\alpha,\nu_\beta) & b(\lambda_\alpha,\nu_\beta) & 0 \\ 0 & b(\lambda_\alpha,\nu_\beta) & c(\lambda_\alpha,\nu_\beta) & 0 \\ 0 & 0 & 0 & a(\lambda_\alpha,\nu_\beta) \end{bmatrix}_{\alpha\beta}$$

Recalling that the R-matrix is only defined up to a constant, we can divide through by $c(\lambda_\alpha,\nu_\beta)$ to obtain:

$$\begin{bmatrix} \frac{a(\lambda_\alpha,\nu_\beta)}{b(\lambda_\alpha,\nu_\beta)} & 0 & 0 & 0 \\ 0 & \frac{c(\lambda_\alpha,\nu_\beta)}{b(\lambda_\alpha,\nu_\beta)} & 1 & 0 \\ 0 & 1 & \frac{c(\lambda_\alpha,\nu_\beta)}{b(\lambda_\alpha,\nu_\beta)} & 0 \\ 0 & 0 & 0 & \frac{a(\lambda_\alpha,\nu_\beta)}{b(\lambda_\alpha,\nu_\beta)} \end{bmatrix}_{\alpha\beta}$$

Making the substitution $\lambda_\alpha - \nu_\beta - \eta \to \lambda - \lambda'$ leaves us with:

$$\frac{a(\lambda_\alpha,\nu_\beta)}{b(\lambda_\alpha,\nu_\beta)} = \frac{\sinh(\lambda-\lambda'+2\eta)}{\sinh(\lambda-\lambda')}$$
$$\frac{c(\lambda_\alpha,\nu_\beta)}{b(\lambda_\alpha,\nu_\beta)} = \frac{\sinh(2\eta)}{\sinh(\lambda'-\lambda)}$$

We can now write the R-matrix as:

$$R_{\alpha\beta}(\lambda_\alpha, \lambda_\beta) = \begin{bmatrix} f(\lambda', \lambda) & 0 & 0 & 0 \\ 0 & g(\lambda', \lambda) & 1 & 0 \\ 0 & 1 & g(\lambda', \lambda) & 0 \\ 0 & 0 & 0 & f(\lambda', \lambda) \end{bmatrix}_{\alpha\beta}$$

Where

$$f(\lambda', \lambda) = \frac{\sinh(\lambda - \lambda' + 2\eta)}{\sinh(\lambda - \lambda')}$$
$$g(\lambda', \lambda) = \frac{\sinh(2\eta)}{\sinh(\lambda' - \lambda)}$$

Whilst λ and λ' are arbitrary, it is clear that the R matrix has no ν dependance when viewed graphically, as above:

The new definition of the R matrix will affect our commutation relation given by (2.1.22), which becomes:

$$\mathcal{A}(\lambda) \prod_{\alpha=1}^{M} \mathcal{B}(\lambda_\alpha)| \Uparrow \rangle = \Lambda(\lambda) \prod_{\alpha=1}^{M} \mathcal{B}(\lambda_\alpha)| \Uparrow \rangle + \sum_{\beta=1}^{M} \Lambda_\beta(\lambda) \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} \mathcal{B}(\lambda_\alpha)| \Uparrow \rangle \tag{3.3.8}$$

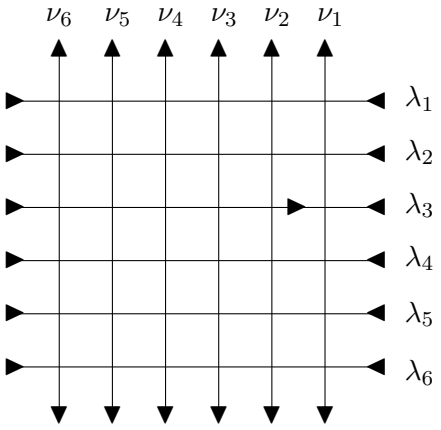
where:

$$\Lambda(\lambda) = \prod_{\alpha=1}^{N} \sinh(\lambda - \nu_\alpha + \eta) \prod_{\gamma=1}^{M} f(\lambda, \lambda_\gamma) \tag{3.3.9}$$

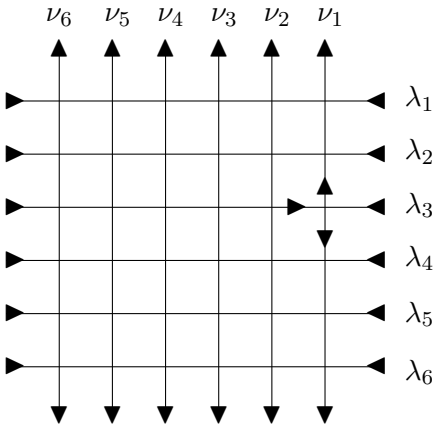
$$\Lambda_\beta(\lambda) = \prod_{\alpha=1}^{N} \sinh(\lambda - \nu_\alpha + \eta) g(\lambda_\beta, \lambda) \prod_{\substack{\gamma=1 \\ \gamma\neq\beta}}^{M} f(\lambda_\beta, \lambda_\gamma) \tag{3.3.10}$$

3.3.4 Derivation of $H_N^{(M)}$

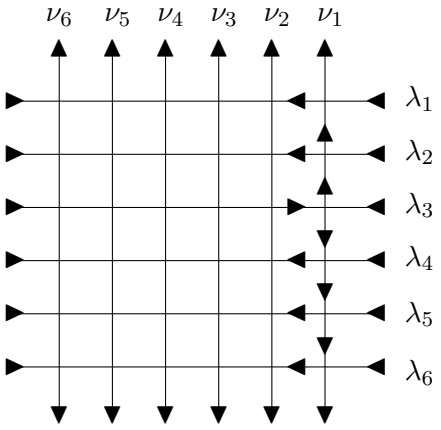
Consider the function $H_N^{(M)}$, and represent it graphically as a lattice with DWBC and an inverted arrow at λ_M in $| \Uparrow \rangle$. For example, consider the case $N = 6$ and $M = 3$. $H_6^{(3)}$ is represented:



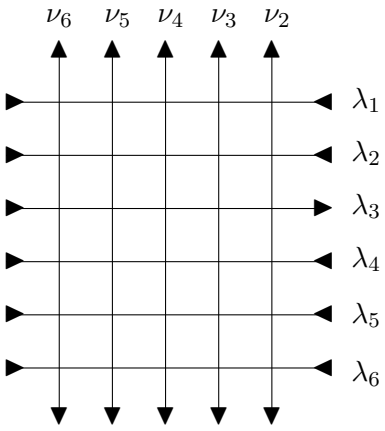
We can now see that the vertex at (λ_3, ν_1) has two 'in' arrows. To enforce the conservation of arrow flow, we must freeze the vertical arrows of to both point 'out,' giving:



This configuration now freezes the vertices at (λ_2, ν_1) and (λ_4, ν_1), which in turn which freeze more vertices, and so on until all vertices (λ_α, ν_1) (i.e. the entire space $\mathcal{H}_1$) are frozen to:



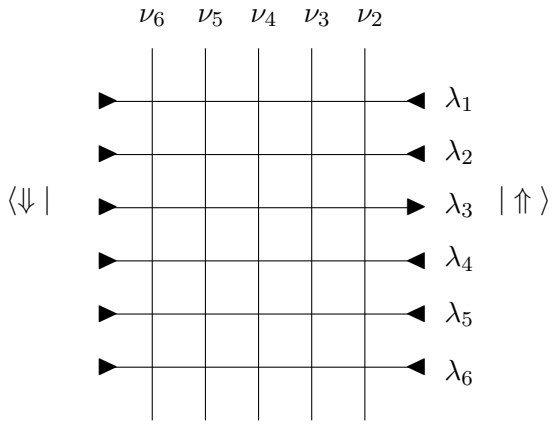
The frozen vertices all represent constants, and we can 'peel' them off the lattice, leaving a new $N \times N-1$ lattice, multiplied by constants. Considering that the second column must have at least one inverted arrow (proved earlier) and we have just shown that the one inverted arrow on any row freezes all the other arrows in the column, we can say that the all non-zero weight lattices with DWBC must have one, and only one inverted arrow in the first column, which proves (3.3.3). Also notice that the vertex at (λ_1, ν_1) is a b-vertex, as is (λ_1, ν_2). Vertex (λ_1, ν_3) is a c-vertex, and (λ_1, ν_4), (λ_1, ν_5) and (λ_1, ν_6) are a-vertices. It is not difficult to see what the pattern will be for higher M: all vertices above λ_M will be b's, all below λ_M will be a's, and the λ_M term itself will be a c-vertex. Thus, we have:



$$\times \sinh(2\eta) \prod_{\alpha=4}^{6} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=1}^{2} \sinh(\lambda_\alpha - \nu_1 + \eta)$$

The $\sinh$ function comes form the definition of a, b and c. If we remove the 'up' arrows from from the top of the new lattice, as well as remove the 'down' arrows

from the bottom, they will correspond to $|\Uparrow\rangle_2$ and $\langle\Downarrow|_2$ (as defined in (3.3.6) leaving us with the representation:



$$\times \sinh(2\eta) \prod_{\alpha=4}^{6} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=1}^{3} \sinh(\lambda_\alpha - \nu_1 + \eta)$$

we can see that the vertex is nothing more than a product of $\mathcal{B}_j$'s ($j \neq 3$) and $\mathcal{A}_3$ in the middle. We now use the the algebraic Bethe ansatz to solve this model by pushing the $\mathcal{A}_3$ through the $\mathcal{B}$'s. The value H_3^6 is

$$H_3^6 = Z_6^{-1} \sinh(2\eta) \prod_{\alpha=1}^{2} \sinh \lambda_j - \nu_1 - \eta$$
$$\times \langle\Downarrow|_2 \mathcal{B}(\lambda_6)_2 \mathcal{B}(\lambda_5)_2 \mathcal{B}(\lambda_4)_2 \mathcal{A}_(\lambda_3) 2 \mathcal{B}_(\lambda_2) 2 \mathcal{B}(\lambda_1)_2 |\Uparrow\rangle_2$$

This can be easily generalised for higher values of M, giving us:

$$\begin{aligned} H_N^{(M)} =& Z_N^{-1} \sinh(2\eta) \prod_{\alpha=1}^{M-1} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\ & \times \langle\Downarrow| \mathcal{B}_2(\lambda_N) \dots \mathcal{B}_2(\lambda_{M+1}) \mathcal{A}_2(\lambda_M) \mathcal{B}_2(\lambda_{M-1}) \dots \mathcal{B}_2(\lambda_1) |\Uparrow\rangle \end{aligned} \tag{3.3.11}$$

To simplify this we must make use of the relation (3.3.8) to 'push' the $\mathcal{A}_2(\lambda_M)$ through the $\mathcal{B}_2$'s:

$$\mathcal{A}_1(\lambda_M) \mathcal{B}_2(\lambda_{M-1}) \dots \mathcal{B}_2(\lambda_1) |\Uparrow\rangle = [\Lambda(\lambda_M) \prod_{\alpha=1}^{M} \mathcal{B}_2(\lambda_\alpha) + \sum_{\beta=1}^{M} \Lambda_\beta(\lambda) \prod_{\substack{\alpha=1 \\ \alpha \neq \beta}}^{M} \mathcal{B}_2(\lambda_\alpha)] |\Uparrow\rangle$$

Substituting in (3.3.10) and (3.3.10):

$$\mathcal{A}_2(\lambda_M)\prod_{j=1}^{M-1}\mathcal{B}_2(\lambda_j)|\Uparrow\rangle = \prod_{\alpha=2}^{N}\sinh(\lambda_M-\nu_\alpha+\eta)\prod_{\gamma=1}^{M-1}f(\lambda_M,\lambda_\gamma)\prod_{j=1}^{M-1}\mathcal{B}_2(\lambda_j)|\Uparrow\rangle$$
$$+\sum_{\beta=1}^{M-1}\prod_{\alpha=1}^{N}\sinh(\lambda_\beta-\nu_\alpha+\eta)g(\lambda_\beta,\lambda_M)$$
$$\times\prod_{\substack{\gamma=1\\\gamma\neq\beta}}^{M-1}f(\lambda_\beta,\lambda_\gamma)\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{M-1}\mathcal{B}_2(\lambda_\alpha)|\Uparrow\rangle$$

Note that the first product over α starts from $\alpha = 2$ because, it represents the eigenvector of $\mathcal{A}_2$, not $\mathcal{A}_2$. If we multiply the product over γ in the third line by $f(\lambda_\beta,\lambda_M)/f(\lambda_\beta,\lambda_M)$, the product changes from $(1\ldots M-1)$ to $(1\ldots M)$, not including β:

$$=\prod_{\alpha=2}^{N}\sinh(\lambda_M-\nu_\alpha+\eta)\prod_{\gamma=1}^{M-1}f(\lambda_M,\lambda_\gamma)\prod_{j=1}^{M-1}\mathcal{B}_2(\lambda_j)|\Uparrow\rangle$$
$$+\sum_{\beta=1}^{M-1}\prod_{\alpha=2}^{N}\sinh(\lambda_\beta-\nu_\alpha+\eta)\frac{g(\lambda_\beta,\lambda_M)}{f(\lambda_\beta,\lambda_M)}\prod_{\substack{\gamma=1\\\gamma\neq\beta}}^{M}f(\lambda_\beta,\lambda_\gamma)\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{M-1}\mathcal{B}_2(\lambda_\alpha)|\Uparrow\rangle$$

The trick now is to recognise that the first term is equivalent to an M-th term of the sum over β, because at $\gamma = M$, the fraction $g(\lambda,\lambda)/f(\lambda,\lambda) = \sinh 2\eta/\sinh 2\eta = 1$. Incorporating the first term into the sum gives:

$$=\sum_{\beta=1}^{M}\prod_{\alpha=2}^{N}\sinh(\lambda_\beta-\nu_\alpha+\eta)\frac{g(\lambda_\beta,\lambda_M)}{f(\lambda_\beta,\lambda_M)}\prod_{\substack{\gamma=1\\\gamma\neq\beta}}^{M}f(\lambda_\beta,\lambda_\gamma)\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{M-1}\mathcal{B}(\lambda_\alpha)|\Uparrow\rangle$$

Substituting this into (3.3.11) and noting that entire sum over β is a constant, we

may write:

$$\begin{aligned}
H_N^{(M)} =& Z_N^{-1}\sinh(2\eta)\prod_{\alpha=1}^{M-1}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=M+1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)\\
&\langle\Downarrow|\mathcal{B}_2(\lambda_N)\dots\mathcal{B}_2(\lambda_{M+1})\times\sum_{\beta=1}^{M}\prod_{\alpha=2}^{N}\sinh(\lambda_\beta-\nu_\alpha+\eta)\frac{g(\lambda_\beta,\lambda_M)}{f(\lambda_\beta,\lambda_M)}\\
&\times\prod_{\substack{\gamma=1\\\gamma\neq\beta}}^{M}f(\lambda_\beta,\lambda_\gamma)\times\mathcal{B}_2(\lambda_{M-1})\dots\mathcal{B}_2(\lambda_1)|\Uparrow\rangle\\
=& Z_N^{-1}\sinh(2\eta)\prod_{\alpha=1}^{M-1}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=M+1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)\\
&\times\sum_{\beta=1}^{M}\prod_{\alpha=2}^{N}\sinh(\lambda_\beta-\nu_\alpha+\eta)\frac{g(\lambda_\beta,\lambda_M)}{f(\lambda_\beta,\lambda_M)}\prod_{\substack{\gamma=1\\\gamma\neq\beta}}^{M}f(\lambda_\beta,\lambda_\gamma)\\
&\times\langle\Downarrow|\mathcal{B}_2(\lambda_N)\dots\mathcal{B}_2(\lambda_{M+1})\mathcal{B}_2(\lambda_{M-1})\dots\mathcal{B}_2(\lambda_1)|\Uparrow\rangle
\end{aligned}$$

The third line is simply an $N-1\times N-1$ partition function, missing λ_β and ν_1, which we express as $Z_{N-1}(\{\lambda\}^N_{\alpha=1\ \alpha\neq\beta};\{\nu\}^N_{\kappa=2}$. Hence, we finally have the result:

$$\begin{aligned}
H_N^{(M)} =& Z_N^{-1}\sinh(2\eta)\prod_{\alpha=1}^{M-1}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=M+1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)\\
&\sum_{\beta=1}^{M}\prod_{\alpha=2}^{N}\sinh(\lambda_\beta-\nu_\alpha+\eta)\frac{g(\lambda_\beta,\lambda_M)}{f(\lambda_\beta,\lambda_M)}\\
&\times\prod_{\substack{\gamma=1\\\gamma\neq\beta}}^{M}f(\lambda_\beta,\lambda_\gamma)Z_{N-1}(\{\lambda\}^N_{\alpha=1\ \alpha\neq\beta};\{\nu\}^N_{\kappa=2}
\end{aligned}\tag{3.3.12}$$

3.3.5 Derivation of $G_N^{(M)}$

Using (3.3.3) and (3.3.11), we see that:

$$\begin{aligned}
G_N^{(M)} =& Z_N^{-1}\sinh(2\eta)\sum_{\beta=1}^{M}\prod_{\alpha=1}^{\beta-1}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=\beta+1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)\\
&\times\langle\Downarrow|\mathcal{B}_2(\lambda_N)\dots\mathcal{B}_2(\lambda_{\beta+1})\mathcal{A}_2(\lambda_\beta)\mathcal{B}_2(\lambda_{\beta-1})\dots\mathcal{B}_2(\lambda_1)|\Uparrow\rangle
\end{aligned}\tag{3.3.13}$$

We can use (3.3.8) to push $\mathcal{A}_\beta$ through the $\mathcal{B}$'s, as we did with $H_N^{(M)}$. However, if we did this we would be left with a double summation. To avoid this, we will

approach the problem a different way. If we were to choose the term of the sum corresponding to $\beta = M$, and then expanded that term only using (3.3.8) then we would have:

$$\begin{aligned}
G_N^{(M)} =& Z_N^{-1} \sinh(2\eta) \prod_{\alpha=1}^{M-1} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \times \langle \Downarrow | \mathcal{B}_2(\lambda_N) \dots \mathcal{B}_2(\lambda_{M+1}) \mathcal{A}_2(\lambda_M) \mathcal{B}_2(\lambda_{M-1}) \dots \mathcal{B}_2(\lambda_1) | \Uparrow \rangle \\
=& Z_N^{-1} \sinh(2\eta) \prod_{\alpha=1}^{M-1} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \times \langle \Downarrow | \mathcal{B}_2(\lambda_N) \dots \mathcal{B}_2(\lambda_{M+1}) \times \mathcal{A}_2(\lambda_M) \prod_{\alpha=1}^{M-1} \mathcal{B}_2(\lambda_\alpha) | \Uparrow \rangle \\
=& Z_N^{-1} \sinh(2\eta) \prod_{\alpha=1}^{M-1} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \times \langle \Downarrow | \mathcal{B}_2(\lambda_N) \dots \mathcal{B}_2(\lambda_{M+1}) (\prod_{k=2}^{N} \sinh(\lambda_M - \nu_k + \eta) \prod_{\gamma=1}^{M-1} f(\lambda_M, \lambda_\gamma) \\
& + \text{other terms}) \prod_{\alpha=1}^{M-1} \mathcal{B}_2(\lambda_\alpha) | \Uparrow \rangle \\
=& Z_N^{-1} \sinh(2\eta) \prod_{\alpha=1}^{M-1} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \Big(\times \prod_{k=2}^{N} \sinh(\lambda_M - \nu_k + \eta) \prod_{\gamma=1}^{M-1} f(\lambda_M, \lambda_\gamma) \\
& \times \langle \Downarrow | \mathcal{B}_2(\lambda_N) \dots \mathcal{B}_2(\lambda_{M+1}) \mathcal{B}_2(\lambda_{M-1}) \dots \mathcal{B}_2(\lambda_1) | \Uparrow \rangle + \text{other terms} \Big)
\end{aligned}$$

Multiplying by $\sinh(\lambda_M - \nu_1 - \eta) / \sinh(\lambda_M - \nu_1 - \eta)$ allows us to make the first product over α from $1 \dots M$ rather than $1 \dots M-1$, but leaves an extra $1/\sinh(\lambda_M - \nu_1 - \eta)$; and bringing $\sinh(2\eta)$ inside the sum gives:

$$\begin{aligned}
=& Z_N^{-1} \prod_{\alpha=1}^{M} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \Big(\times \frac{\sinh(2\eta) \prod_{k=2}^{N} \sinh(\lambda_M - \nu_k + \eta)}{\sinh(\lambda_M - \nu_1 - \eta)} \prod_{\gamma=1}^{M-1} f(\lambda_M, \lambda_\gamma) \qquad (3.3.14) \\
& \times \langle \Downarrow | \mathcal{B}_2(\lambda_N) \dots \mathcal{B}_2(\lambda_{M+1}) \mathcal{B}_2(\lambda_{M-1}) \dots \mathcal{B}_2(\lambda_1) | \Uparrow \rangle + \text{other terms} \Big) \qquad (3.3.15)
\end{aligned}$$

Note that the "other terms" are different in each line, but all correspond to other products involving $\mathcal{B}_2(\lambda_M)$. In fact, the first term in the brackets is the only pos-

sible term in $G_N^{(M)}$ that does not contain $\mathcal{B}_2(\lambda_M)$. This is because, although all the other terms in (3.3.13) will 'push out' a (λ_j) argument, they will all be for $j < M$, meaning there is no other way to remove the (λ_j) argument from the $\mathcal{B}_2(\lambda_j)$'s.

Also note that (3.3.13) is symmetric under permutations of $\lambda_1 \dots \lambda_M$. This is because it makes no difference which order we arrange the terms in the sum, the products in the first line are commutative, and the product of $\mathcal{B}_2(\lambda_j)$'s is commutative.

The first term of our most recent expression for $G_N^{(M)}$ (3.3.15) will be symmetric under permutations of $\lambda_1 \dots \lambda_{M-1}$ (we don't include the λ_M term because it is absent). This is because it only involves a product of constants, whose order is unimportant, and a product of $\mathcal{B}_2(\lambda_j)$'s which is commutative.

It follows from these symmetry considerations that the rest of the "other terms" in (3.3.15) must be the cyclic permutations of $\lambda_1 \dots \lambda_M$. If the first term of (3.3.15) corresponds to the cycle with λ_M missing, the rest of the terms must share the same structure, only the β-th term will have λ_β missing. Using the fact that

$$\langle \Downarrow | \mathcal{B}_2(\lambda_N) \dots \mathcal{B}_2(\lambda_{M+1}) \mathcal{B}_2(\lambda_{M-1}) \dots \mathcal{B}_2(\lambda_1) | \Uparrow \rangle$$

represents an $N-1 \times N-1$ partition function, missing (λ_β, ν_j), we have the following expression for correlation function $G_N^{(M)}$:

$$\begin{aligned} G_N^{(M)} =& Z_N^{-1} \prod_{\alpha=1}^{M} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\ & \times \sum_{\beta=1}^{M} \prod_{\alpha=2}^{N} \sinh(\lambda_\beta - \nu_\alpha + \eta) \frac{\sinh(2\eta)}{\sinh(\lambda_\beta - \nu_1 - \eta)} \\ & \times \prod_{\substack{\gamma=1 \\ \gamma \neq \beta}}^{M} f(\lambda_\beta, \lambda_\gamma) Z_{N-1}(\{\lambda\}_{\alpha=1\ \alpha\neq\beta}^{N}; \{\nu\}_{\kappa=2}^{N}) \end{aligned} \tag{3.3.16}$$

3.3.6 Implications

The significance of (3.3.16) is that when $M = N$, the equation simplifies, and gives a recursion relation between the partition function of an $N \times N$ lattice and

an $N-1\times N-1$ lattice. Recalling that $G_N^{(M)}=1$:

$$\begin{aligned}1 =& Z_N^{-1}\sinh(2\eta)\prod_{\alpha=1}^{N}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=N+1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)\\ &\times\sum_{\beta=1}^{N}\prod_{\alpha=2}^{N}\sinh(\lambda_\beta-\nu_\alpha+\eta)\frac{1}{\sinh(\lambda_\beta-\nu_1-\eta)} \qquad (3.3.17)\\ &\times\prod_{\substack{\gamma=1\\ \gamma\neq\beta}}^{N}f(\lambda_\beta,\lambda_\gamma)Z_{N-1} \qquad (3.3.18)\end{aligned}$$

Bringing the first product (which is a constant) inside the sum gives:

$$\begin{aligned}Z_N =& \sinh(2\eta)\sum_{\beta=1}^{N}\prod_{\alpha=2}^{N}\sinh(\lambda_\beta-\nu_\alpha+\eta)\frac{\prod_{\alpha=1}^{N}\sinh(\lambda_\alpha-\nu_1-\eta)}{\sinh(\lambda_\beta-\nu_1-\eta)}\\ &\times\prod_{\substack{\gamma=1\\ \gamma\neq\beta}}^{N}f(\lambda_\beta,\lambda_\gamma)Z_{N-1}\end{aligned}$$

Canceling outthe β-th term of the second product leaves us the very imporrtant relation:

$$\begin{aligned}Z_N(\{\lambda\}_{\alpha=1}^{N};\{\nu\}_{\kappa=1}^{N}) =& \sinh(2\eta)\sum_{\beta=1}^{N}\prod_{\alpha=2}^{N}\sinh(\lambda_\beta-\nu_\alpha+\eta)\prod_{\substack{\alpha=1\\ \alpha\neq\beta}}^{N}\sinh(\lambda_\alpha-\nu_1-\eta)\\ &\times\prod_{\substack{\gamma=1\\ \gamma\neq\beta}}^{N}f(\lambda_\beta,\lambda_\gamma)Z_{N-1}(\{\lambda\}_{\alpha=1\ \alpha\neq\beta}^{N};\{\nu\}_{\kappa=2}^{N}) \qquad (3.3.19)\end{aligned}$$

This is the recurrence relation between the value of an $N\times N$ partition function and an $N-1\times N-1$ partition function with DWBC. It can be used to calculate Z_N when an initial state, Z_1, is given. Due to the symmetry of Z_N with respect to permutations of the variables $\nu_1\ldots\nu_N$, we can express this in the general form:

$$\begin{aligned}Z_N(\{\lambda\}_{\alpha=1}^{N};\{\nu\}_{\kappa=1}^{N}) =& \sinh(2\eta)\sum_{\beta=1}^{N}\prod_{\substack{\alpha=1\\ \alpha\neq j}}^{N}\sinh(\lambda_\beta-\nu_\alpha+\eta)\prod_{\substack{\alpha=1\\ \alpha\neq\beta}}^{N}\sinh(\lambda_\alpha-\nu_j-\eta)\\ &\times\prod_{\substack{\gamma=1\\ \gamma\neq\beta}}^{N}f(\lambda_\beta,\lambda_\gamma)Z_{N-1}(\{\lambda\}_{\alpha=1\ \alpha\neq\beta}^{N};\{\nu\}_{\kappa=j}^{N})\end{aligned}$$

3.4 Determinant Representations for Boundary Correlation Functions

3.4.1 Proof that det$\mathcal{Z}_N$ Satisfies Recurrence Relation

We will now prove that the determinant representation of the partition function satisfies the recursion relation (3.3.19) Consider the function

$$g_N(\lambda) = \frac{\prod_{\alpha=1}^{N} \sinh(\lambda_\alpha - \lambda + 2\eta)}{\prod_{\kappa=1}^{N} \sinh(\lambda - \lambda_\kappa + \eta)} \tag{3.4.1}$$

For this function the following identity exists[3]:

$$\frac{\prod_{\substack{\gamma=1 \\ \gamma\neq\alpha}}^{N} \sinh(\lambda_\gamma - \lambda_\alpha + 2\eta)}{\prod_{j=1}^{N} \sinh(\lambda_\alpha - \nu_j - \eta)} = \sum_{\kappa=1}^{N} \frac{\prod_{\substack{\gamma=1 \\ \gamma\neq\alpha}}^{N} \sinh(\lambda_\gamma - \nu_\kappa + \eta)}{\prod_{\substack{j=1 \\ j\neq\kappa}}^{N} \sinh(\nu_\kappa - \nu_j)} \frac{1}{\sinh(\lambda_\alpha - \nu_k - \eta)} \tag{3.4.2}$$

If we multiply both sides of (3.4.2) by $\sinh(\lambda_\alpha - \lambda_\alpha + 2\eta) = \sinh(2\eta)$ we have:

$$\sinh(\lambda_\alpha - \lambda_\alpha + 2\eta) \frac{\prod_{\substack{\gamma=1 \\ \gamma\neq\alpha}}^{N} \sinh(\lambda_\gamma - \lambda_\alpha + 2\eta)}{\prod_{j=1}^{N} \sinh(\lambda_\alpha - \nu_j - \eta)} =$$

$$\sinh(2\eta) \times \sum_{\kappa=1}^{N} \frac{\prod_{\substack{\gamma=1 \\ \gamma\neq\alpha}}^{N} \sinh(\lambda_\gamma - \nu_\kappa + \eta)}{\prod_{\substack{j=1 \\ j\neq\kappa}}^{N} \sinh(\nu_\kappa - \nu_j)} \frac{1}{\sinh(\lambda_\alpha - \nu_k - \eta)}$$

[3] Bogoliubov, N. M.; Pronko, A. G.; Zvonarev, M. B, Boundary correlation functions of the six-vertex model, 11

Because $\sinh(2\eta)$ does not depend on the summation index, γ we can bring it inside the sum. We can do the same for $\sinh(\lambda_\alpha - \lambda_\alpha + 2\eta)$ so that the product on the numerator becomes a product over all values of γ.

$$\frac{\prod_{\gamma=1}^{N} \sinh(\lambda_\gamma - \lambda_\alpha + 2\eta)}{\prod_{j=1}^{N} \sinh(\lambda_\alpha - \nu_j - \eta)} = \sum_{\kappa=1}^{N} \frac{\prod_{\substack{\gamma=1 \\ \gamma\neq\alpha}}^{N} \sinh(\lambda_\gamma - \nu_\kappa + \eta)}{\prod_{\substack{j=1 \\ j\neq\kappa}}^{N} \sinh(\nu_\kappa - \nu_j)} \frac{\sinh(2\eta)}{\sinh(\lambda_\alpha - \nu_k - \eta)}$$

If we multiply the RHS by the constant $\frac{\sinh(\lambda_\alpha - \nu_\alpha + \eta)}{\sinh(\lambda_\alpha - \nu_\alpha + \eta)}$ and bring it inside the sum over κ, we can then bring the numerator inside the product over γ so that it is now over all values:

$$\begin{aligned}\frac{\prod_{\gamma=1}^{N} \sinh(\lambda_\gamma - \lambda_\alpha + 2\eta)}{\prod_{j=1}^{N} \sinh(\lambda_\alpha - \nu_j - \eta)} = &\sum_{\kappa=1}^{N} \frac{\sinh(\lambda_\alpha - \nu_\alpha + \eta) \times \prod_{\substack{\gamma=1 \\ \gamma\neq\alpha}}^{N} \sinh(\lambda_\gamma - \nu_\kappa + \eta)}{\prod_{\substack{j=1 \\ j\neq\kappa}}^{N} \sinh(\nu_\kappa - \nu_j)} \\ &\times \frac{\sinh(2\eta)}{\sinh(\lambda_\alpha - \nu_k - \eta)\sinh(\lambda_\alpha - \nu_\alpha + \eta)}\end{aligned}$$

$$\begin{aligned}&= \sum_{\kappa=1}^{N} \frac{\prod_{\gamma=1}^{N} \sinh(\lambda_\gamma - \nu_\kappa + \eta)}{\prod_{\substack{j=1 \\ j\neq\kappa}}^{N} \sinh(\nu_\kappa - \nu_j)} \\ &\times \frac{\sinh(2\eta)}{\sinh(\lambda_\alpha - \nu_k - \eta)\sinh(\lambda_\alpha - \nu_\alpha + \eta)}\end{aligned}$$

Which allows us to rewrite (3.4.2) as:

$$g_N(\lambda_\alpha) = \sum_{\kappa=1}^{N} \Phi_\kappa \phi(\lambda_\alpha, \nu_\kappa), \qquad \alpha = 1 \ldots N \tag{3.4.3}$$

Where ϕ is as defined in (3.2.9) and Φ_k is defined as:

$$\Phi_\kappa = \frac{\prod_{\alpha=1}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta)}{\prod_{\substack{j=1 \\ j\neq\kappa}}^{N} \sinh(\nu_\kappa - \nu_j)} \qquad \kappa = 1 \ldots N \tag{3.4.4}$$

Expanding (3.4.3) gives N linearly independent equations:

$$\begin{aligned} g_N(\lambda_1) =& \Phi_1\phi(\lambda_\alpha,\nu_1) + \Phi_2\phi(\lambda_\alpha,\nu_2) + \ldots + \Phi_N\phi(\lambda_\alpha,\nu_N) \\ g_N(\lambda_2) =& \Phi_1\phi(\lambda_\alpha,\nu_1) + \Phi_2\phi(\lambda_\alpha,\nu_2) + \ldots + \Phi_N\phi(\lambda_\alpha,\nu_N) \\ & \vdots \\ g_N(\lambda_N) =& \Phi_1\phi(\lambda_\alpha,\nu_1) + \Phi_2\phi(\lambda_\alpha,\nu_2) + \ldots + \Phi_N\phi(\lambda_\alpha,\nu_N) \end{aligned}$$

If we treat Φ_κ as our variable, we can solve this system of equations using Cramer's rule. Using the fact that the determinant of the matrix with entries $\phi(\lambda_\alpha,\nu_\kappa)$ is $\det_N \mathcal{Z}$, we have:

$$\Phi_1 = \frac{\det_N \mathcal{Z}_1}{\det_N \mathcal{Z}}$$

Where $\det_N \mathcal{Z}_1$ is matrix $\det_N \mathcal{Z}$ with the first column replaced by $g_N(\lambda_1)\ldots g_N(\lambda_N)$ (this is Cramer's rule). Dividing through by Φ_1 and multiplying out $\det_N \mathcal{Z}$ gives:

$$\det_N \mathcal{Z} = \frac{1}{\Phi_1}\begin{vmatrix} g_N(\lambda_1) & \phi(\lambda_1,\nu_2) & \ldots & \phi(\lambda_1,\nu_N) \\ g_N(\lambda_2) & \phi(\lambda_2,\nu_2) & \ldots & \phi(\lambda_2,\nu_N) \\ \vdots & \vdots & \ddots & \vdots \\ g_N(\lambda_N) & \phi(\lambda_N,\nu_2) & \ldots & \phi(\lambda_N,\nu_N) \end{vmatrix}_N$$

Using elementary linear algebra techniques, we can calculate the determinant by expanding along the first column, so that we are left with:

$$\det_N \mathcal{Z} = \frac{1}{\Phi_1}\sum_{\beta=1}^{N}(-1)^{\beta-1} g_N(\lambda_\beta)\Delta_{N-1}^{(\beta)} \tag{3.4.5}$$

where $\Delta_{N-1}^{(\beta)}$ denotes the determinant of an $N-1\times N-1$ matrix:

$$\Delta_{N-1}^{(\beta)} = \begin{vmatrix} \phi(\lambda_1,\nu_2) & \ldots & \phi(\lambda_1,\nu_N) \\ \vdots & \ddots & \vdots \\ \phi(\lambda_{\beta-1},\nu_2) & \ldots & \phi(\lambda_{\beta-1},\nu_N) \\ \phi(\lambda_{\beta+1},\nu_2) & \ldots & \phi(\lambda_{\beta+1},\nu_N) \\ \vdots & \ddots & \vdots \\ \phi(\lambda_N,\nu_2) & \ldots & \phi(\lambda_N,\nu_N) \end{vmatrix}_{N-1} \tag{3.4.6}$$

Equation (3.4.6) is helpful in expressing $Z_{N-1}(\{\lambda\}_{\alpha=1\ \alpha\neq\beta}^{N};\{\nu\}_{\kappa=2}^{N})$ using (3.2.7). Looking closely at (3.2.7) you can see that $Z_{N-1})$ is similar to the formula for Z_N, only we cannot include ν_1 or λ_β in any of the terms. Before we represent Z_{N-1}, we will simplify some of the products in (3.2.7), when they do not include ν_1 or λ_β:

Products 1 and 2

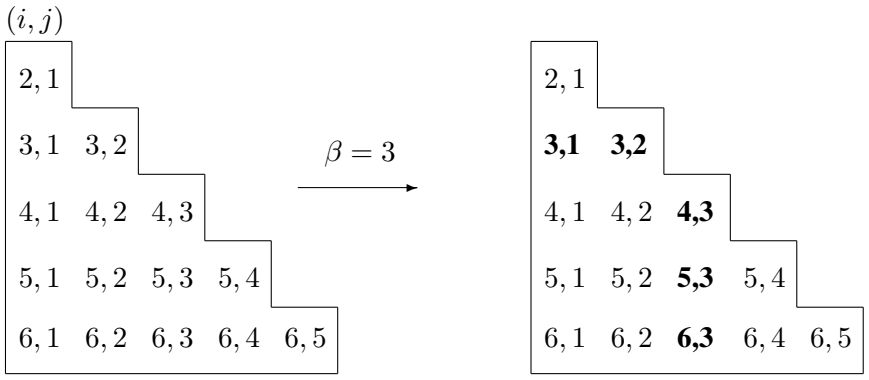
It is quite simple to see that if we take out all ν_1 and λ_β from the product in the numerator of (3.2.7) we have:

$$\begin{aligned}
&\prod_{\alpha=1}^{N}\prod_{\kappa=1}^{N}\sinh(\lambda_\alpha-\nu_\kappa+\eta)\sinh(\lambda_\alpha-\nu_\kappa-\eta)=\\
&[\prod_{\kappa=1}^{N}\sinh(\lambda_\beta-\nu_\kappa+\eta)\sinh(\lambda_\beta-\nu_\kappa-\eta)]\\
&\times[\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)\sinh(\lambda_\alpha-\nu_1-\eta)]\\
&\times\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{N}\prod_{\kappa=2}^{N}\sinh(\lambda_\alpha-\nu_\kappa+\eta)\sinh(\lambda_\alpha-\nu_\kappa-\eta)
\end{aligned}\tag{3.4.7}$$

The reason for the $\alpha\neq\beta$ in the terminal of the third product, is that product corresponding to $\alpha=\beta,\kappa=1$ is already included in the second product, and we do not want to count it twice.

Product 3

We now want to take out all ν_1 and λ_β from the first product in the denominator. This is a bit trickier. Firstly, given that the terminals α and β are dummy variables, to avoid confusion with λ_β we will change the product indices to i and j, so that our product becomes $\prod_{\substack{1\leq j<i\leq N\\i,j\neq\beta}}\sinh(\lambda_i-\lambda_j)$. To simplify this, we will represent it pictorially in the example $N=6$, $\beta=3$:



The LHS represents all possible values of (i, j) where $1 \leq j < i \leq 6$, while on the RHS, the numbers in bold are the pairs of (i, j) we cannot include in the product when $\beta = 3$. Our product becomes:

$$\prod_{1\leq j<i\leq 6} \sinh(\lambda_i - \lambda_j) = \prod_{\substack{1\leq j<i\leq 6 \\ i,j\neq 3}} \sinh(\lambda_i - \lambda_j) \times \sinh(\lambda_4 - \lambda_3)\sinh(\lambda_5 - \lambda_3) \times \sinh(\lambda_6 - \lambda_3)\sinh(\lambda_3 - \lambda_1)\sinh(\lambda_3 - \lambda_2)$$

Swapping the argument of the last two terms gives:

$$\begin{aligned}\prod_{1\leq j<i\leq 6} \sinh(\lambda_i - \lambda_j) &= \prod_{\substack{1\leq j<i\leq 6 \\ i,j\neq 3}} \sinh(\lambda_i - \lambda_j) \times \sinh(\lambda_4 - \lambda_3)\sinh(\lambda_5 - \lambda_3) \\ &\quad \times \sinh(\lambda_6 - \lambda_3) \times (-1)^{3-1}\sinh(\lambda_1 - \lambda_3)\sinh(\lambda_2 - \lambda_3) \\ &= \prod_{\substack{1\leq j<i\leq 6 \\ i,j\neq 3}} \sinh(\lambda_i - \lambda_j) \times (-1)^{3-1} \prod_{\substack{\alpha=1 \\ \alpha\neq 3}}^{N=6} \sinh(\lambda_\alpha - \lambda_3)\end{aligned}$$

We can easily generalize this result to:

$$\prod_{1\leq j<i\leq N} \sinh(\lambda_i - \lambda_j) = \prod_{\substack{1\leq j<i\leq N \\ i,j\neq \beta}} \sinh(\lambda_i - \lambda_j) \times (-1)^{\beta-1} \prod_{\substack{\alpha=1 \\ \alpha\neq \beta}}^{N} \sinh(\lambda_\alpha - \lambda_\beta) \tag{3.4.8}$$

Product 4

Simplifying the second product in the denominator is much simpler than above and we may be able to see the answer immediately. However, if we go through the same

process as we did for Product 3, except for $\kappa = j = 1$, only the first column is eliminated from the product. This is the product over $\kappa = 1$ and $i = 2 \dots N$:

$$\prod_{1\leq\kappa<i\leq N} \sinh(\nu_\kappa - \nu_i) = \prod_{j=2}^{N} \sinh(\nu_1 - \nu_j) \times \prod_{2\leq\kappa<i\leq N} \sinh(\nu_\kappa - \nu_i) \quad (3.4.9)$$

■

We are now in a position to prove that (3.2.7) satisfies (3.3.19), that is, the determinant representation of the partition function satisfies the recurrence relation. Firstly we will write Z_{N-1} using (3.2.7).

$$Z_{N-1}(\{\lambda\}_{\substack{i=1\\ i\neq\beta}}^{N};\{\nu\}_{\kappa=2}^{N}) = \frac{\prod_{\substack{\alpha=1\\ \alpha\neq\beta}}^{N}\prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta)\sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{\substack{1\leq j<i\leq N\\ i,j\neq\beta}} \sinh(\lambda_i - \lambda_j) \prod_{2\leq\kappa<j\leq N} \sinh(\nu_\kappa - \nu_j)} \det_{N-1}\mathcal{Z}$$

We can also see that $\det_{N-1}\mathcal{Z}$ is the determinant of the $N-1\times N-1$ matrix with entries $\phi(\lambda_\alpha, \nu_\kappa)$ where $\alpha \neq \beta$ and $\kappa \neq 1$. This is precisely (3.4.6)! Z_{N-1}becomes:

$$Z_{N-1}(\{\lambda\}_{\substack{i=1\\ i\neq\beta}}^{N};\{\nu\}_{\kappa=2}^{N}) = \frac{\prod_{\substack{\alpha=1\\ \alpha\neq\beta}}^{N}\prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta)\sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{\substack{1\leq j<i\leq N\\ i,j\neq\beta}} \sinh(\lambda_i - \lambda_j) \prod_{2\leq\kappa<j\leq N} \sinh(\nu_\kappa - \nu_j)} \Delta_{N-1}^{(\beta)} \quad (3.4.10)$$

We can see that this is very similar to the (3.2.7) expression for Z_N. Making use of (3.4.1), (3.4.8) and (3.4.9) we can see that:

$$\begin{aligned} Z_{N-1} =& Z_N \times \frac{\Delta_{N-1}^{(\beta)}}{\det_{N-1}\mathcal{Z}} \times \frac{1}{\prod_{\kappa=1}^{N}\sinh(\lambda_\beta - \nu_\kappa + \eta)\sinh(\lambda_\beta - \nu_\kappa - \eta)} \\ &\times \frac{1}{\prod_{\substack{\alpha=1\\ \alpha\neq\beta}}^{N}\sinh(\lambda_\alpha - \nu_1 + \eta)\sinh(\lambda_\alpha - \nu_1 - \eta)} \\ &\times (-1)^{\beta-1}\prod_{\substack{\alpha=1\\ \alpha\neq\beta}}^{N}\sinh(\lambda_\alpha - \lambda_\beta) \times \prod_{j=2}^{N}\sinh(\nu_1 - \nu_j) \end{aligned}$$

Substituting this expression for Z_N into the LHS of equation (3.3.19) gives:

$$\begin{aligned}
Z_N = \sinh(2\eta) \sum_{\beta=1}^{N} \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\kappa=2}^{N} \sinh(\lambda_\beta - \nu_\kappa + \eta) \\
\times \prod_{\substack{\gamma=1 \\ \gamma\neq\beta}}^{N} \frac{\sinh(\lambda_\gamma - \lambda_\beta + 2\eta)}{\sinh(\lambda_\gamma - \lambda_\beta)} \times Z_N \times \frac{\Delta_{N-1}^{(\beta)}}{\det_{N-1}\mathcal{Z}} \\
\times \frac{1}{\prod_{\kappa=1}^{N} \sinh(\lambda_\beta - \nu_\kappa + \eta) \sinh(\lambda_\beta - \nu_\kappa - \eta)} \\
\times \frac{1}{\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \sinh(\lambda_\alpha - \nu_1 - \eta)} \\
\times (-1)^{\beta-1} \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \sinh(\lambda_\alpha - \lambda_\beta) \times \prod_{j=2}^{N} \sinh(\nu_1 - \nu_j)
\end{aligned}$$

We now wish to simplify this. We can cancel the Z_N terms, and move the $\det_{N-1}\mathcal{Z}$ term to the RHS, as both of these are constants. We can also see that the denominator of the product over β in the second line, will cancel with the product over α in the fifth line:

$$\begin{aligned}
\det_{N-1}\mathcal{Z} = \sinh(2\eta) \sum_{\beta=1}^{N} \Delta_{N-1}^{(\beta)} \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\kappa=2}^{N} \sinh(\lambda_\beta - \nu_\kappa + \eta) \\
\times \prod_{\substack{\gamma=1 \\ \gamma\neq\beta}}^{N} \sinh(\lambda_\gamma - \lambda_\beta + 2\eta) \\
\times \frac{1}{\prod_{\kappa=1}^{N} \sinh(\lambda_\beta - \nu_\kappa + \eta) \sinh(\lambda_\beta - \nu_\kappa - \eta)} \\
\times \frac{1}{\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \sinh(\lambda_\alpha - \nu_1 - \eta)} \\
\times (-1)^{\beta-1} \prod_{j=2}^{N} \sinh(\nu_1 - \nu_j)
\end{aligned}$$

Now rewrite the final product in terms of Φ_1 using (3.4.4); break up the products in the third and fourth terms; and bring the $\sinh(2\eta)$ inside the product over γ on the second line, so that it becomes a product over all γ.

$$\begin{aligned}\det_{N-1}\mathcal{Z} = &\sum_{\beta=1}^{N}\Delta_{N-1}^{(\beta)}\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{N}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\kappa=2}^{N}\sinh(\lambda_\beta-\nu_\kappa+\eta)\\ &\times\prod_{\gamma=1}^{N}\sinh(\lambda_\gamma-\lambda_\beta+2\eta)\\ &\times\frac{1}{\prod_{\kappa=1}^{N}\sinh(\lambda_\beta-\nu_\kappa+\eta)}\times\frac{1}{\prod_{\kappa=1}^{N}\sinh(\lambda_\beta-\nu_\kappa-\eta)}\\ &\times\frac{1}{\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)}\times\frac{1}{\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{N}\sinh(\lambda_\alpha-\nu_1-\eta)}\\ &\times(-1)^{\beta-1}\prod_{\gamma=1}^{N}\sinh(\lambda_\gamma-\nu_1+\eta)\times\frac{1}{\Phi_1}\end{aligned}$$

We can now cancel the product over α in the first line with the second product over α in the fourth line and cancel the product over κ in the first line with the first product over κ in the third line, except for the $\sinh(\lambda_\beta-\nu_1+\eta)$ term which we can put into the product over α in the fourth line:

$$\begin{aligned}\det_{N-1}\mathcal{Z} = &\frac{1}{\Phi_1}\sum_{\beta=1}^{N}\Delta_{N-1}^{(\beta)}\times\prod_{\gamma=1}^{N}\sinh(\lambda_\gamma-\lambda_\beta+2\eta)\\ &\times\frac{1}{\prod_{\kappa=1}^{N}\sinh(\lambda_\beta-\nu_\kappa-\eta)}\\ &\times\frac{1}{\prod_{\alpha=1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)}\\ &\times(-1)^{\beta-1}\prod_{\gamma=1}^{N}\sinh(\lambda_\gamma-\nu_1+\eta)\end{aligned}$$

Canceling the product in the third and fourth lines and grouping those in the first and second lines leaves us with:

$$\begin{aligned}\det_{N-1}\mathcal{Z} &= \frac{1}{\Phi_1}\sum_{\beta=1}^{N}(-1)^{\beta-1}\Delta_{N-1}^{(\beta)}\frac{\prod_{\gamma=1}^{N}\sinh(\lambda_\gamma-\lambda_\beta+2\eta)}{\prod_{\kappa=1}^{N}\sinh(\lambda_\beta-\nu_\kappa-\eta)}\\ &= \frac{1}{\Phi_1}\sum_{\beta=1}^{N}(-1)^{\beta-1}g_N(\lambda_\beta)\Delta_{N-1}^{(\beta)}\end{aligned}$$

Which is formula (3.4.5) which we have already showed is correct. This proves that the determinant representation of the partition function (3.2.7) is a solution to the recursion relation (3.3.19).

3.4.2 $H_N^{(M)}$ in Determinant Form

We now wish to derive an expression for $H_N^{(M)}$ using only determinants, which is independent of partition functions. We begin by substituting the expression for Z_{N-1} given by (3.4.10)into (3.3.12). This gives:

$$\begin{aligned}
H_N^{(M)} =& Z_N^{-1} \sinh(2\eta) \prod_{\alpha=1}^{M-1} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \times \sum_{\beta=1}^{M} \prod_{\alpha=2}^{N} \sinh(\lambda_\beta - \nu_\alpha + \eta) \frac{g(\lambda_\beta, \lambda_M)}{f(\lambda_\beta, \lambda_M)} \\
& \times \prod_{\substack{\gamma=1 \\ \gamma\neq\beta}}^{M} f(\lambda_\beta, \lambda_\gamma) \times \frac{\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta) \sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{\substack{1\leq\alpha<\beta\leq N \\ \alpha\neq\beta}} \sinh(\lambda_\beta - \lambda_\alpha) \prod_{2\leq\kappa<j\leq N} \sinh(\nu_\kappa - \nu_j)} \Delta_{N-1}^{(\beta)}
\end{aligned}$$

We expand f and g, and use (3.4.8) to simplify the product over $1 \leq j < i \leq N$ where $i, j \neq \beta$.

$$\begin{aligned}
H_N^{(M)} =& Z_N^{-1} \sinh(2\eta) \prod_{\alpha=1}^{M-1} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \sum_{\beta=1}^{M} \prod_{\alpha=2}^{N} \sinh(\lambda_\beta - \nu_\alpha + \eta) \frac{\sinh(2\eta)}{\sinh(\lambda_M - \lambda_\beta)} \frac{\sinh(\lambda_\beta - \lambda_M)}{\sinh(\lambda_\beta - \lambda_M + 2\eta)} \\
& \times \prod_{\substack{\gamma=1 \\ \gamma\neq\beta}}^{M} \frac{\sinh(\lambda_\gamma - \lambda_\beta + 2\eta)}{\sinh(\lambda_\gamma - \lambda_\beta)} \\
& \times \frac{\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta) \sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{1\leq j<i\leq N} \sinh(\lambda_i - \lambda_j) \prod_{2\leq\kappa<j\leq N} \sinh(\nu_\kappa - \nu_j)} \\
& \times (-1)^{\beta-1} \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \sinh(\lambda_\alpha - \lambda_\beta) \Delta_{N-1}^{(\beta)}
\end{aligned}$$

We can simplify several of these terms. We can cancel part of the denominator of the product in the third line with the product in the fifth line, and then bring the $\sinh(2\eta)$ inside the product on the third line so that it is over all values of γ:

$$\begin{aligned}
H_N^{(M)} =& Z_N^{-1} \sinh(2\eta) \prod_{\alpha=1}^{M-1} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \sum_{\beta=1}^{M} \prod_{\alpha=2}^{N} \sinh(\lambda_\beta - \nu_\alpha + \eta) \frac{1}{\sinh(\lambda_M - \lambda_\beta)} \frac{\sinh(\lambda_\beta - \lambda_M)}{\sinh(\lambda_\beta - \lambda_M + 2\eta)} \\
& \times \prod_{\gamma=1}^{M} \sinh(\lambda_\gamma - \lambda_\beta + 2\eta) \\
& \times \frac{\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta) \sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{1\leq j<i\leq N} \sinh(\lambda_i - \lambda_j) \prod_{2\leq\kappa<j\leq N} \sinh(\nu_\kappa - \nu_j)} \\
& \times (-1)^{\beta-1} \Delta_{N-1}^{(\beta)} \prod_{\substack{\alpha=M+1 \\ \alpha\neq\beta}}^{N} \sinh(\lambda_\alpha - \lambda_\beta)
\end{aligned}$$

We can cancel the $\sinh(\lambda_M - \lambda_\beta)/\sinh(\lambda_\beta - \lambda_M)$ from the second line; use the $1/\sinh(\lambda_\beta - \lambda_M + 2\eta)$ to cancel the M-th term of the product on the third line; bring the denominators of the fourth line out of the summation sign; we can also bring numerator out from the summation sign, except that we will bring it out for all values of α, including β, which by (3.4.8) will leave an additional product within the sum:

$$\begin{aligned}
H_N^{(M)} =& Z_N^{-1} \sinh(2\eta) \prod_{\alpha=1}^{M-1} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \times \frac{\prod_{\alpha=1}^{N} \prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta) \sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{1\leq j<i\leq N} \sinh(\lambda_i - \lambda_j) \prod_{2\leq\kappa<j\leq N} \sinh(\nu_\kappa - \nu_j)} \\
& \sum_{\beta=1}^{M} \prod_{\alpha=2}^{N} \sinh(\lambda_\gamma - \nu_\beta + \eta) \prod_{\gamma=1}^{M-1} \sinh(\lambda_\beta - \lambda_\gamma + 2\eta) \\
& \times (-1)^{\beta-1} \Delta_{N-1}^{(\beta)} \times \frac{\prod_{\substack{\alpha=M+1 \\ \alpha\neq\beta}}^{N} \sinh(\lambda_\alpha - \lambda_\beta)}{\prod_{k=2}^{N} \sinh(\lambda_\beta - \nu_k + \eta) \sinh(\lambda_\beta - \nu_k - \eta)}
\end{aligned}$$

The first product in the sum cancels with part of the product in the fourth line, and we are left with:

$$
\begin{aligned}
H_N^{(M)} =& \sinh(2\eta) \prod_{\alpha=1}^{M-1} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{M} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
&\times \frac{\prod_{\alpha=1}^{N} \prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta) \sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{1\le\alpha<\beta\le N} \sinh(\lambda_\beta - \lambda_\alpha) \prod_{2\le\kappa<j\le N} \sinh(\nu_\kappa - \nu_j)} \\
&\times \sum_{\beta=1}^{M} \frac{\prod_{\gamma=1}^{M-1} \sinh(\lambda_\gamma - \lambda_\beta + 2\eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \lambda_\beta)}{\prod_{\kappa=2}^{N} \sinh(\lambda_\beta - \lambda_\kappa - \eta)} (-1)^{\beta-1} \Delta_{N-1}^{(\beta)}
\end{aligned}
\tag{3.4.11}
$$

We can simplify this equation in steps. Consider the second line and multiply it by the product, $\frac{\prod_{j=2}^{N} \sinh(\nu_1 - \nu_j)}{\prod_{j=2}^{N} \sinh(\nu_1 - \nu_j)}$:

$$
\frac{\prod_{\alpha=1}^{N} \prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta) \sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{1\le\alpha<\beta\le N} \sinh(\lambda_\beta - \lambda_\beta) \prod_{2\le\kappa<j\le N} \sinh(\nu_\kappa - \nu_j)} \times \frac{\prod_{j=2}^{N} \sinh(\nu_1 - \nu_j)}{\prod_{j=2}^{N} \sinh(\nu_1 - \nu_j)} \tag{3.4.12}
$$

Using (3.4.9) we can rewrite (3.4.12) as:

$$
\frac{\prod_{\alpha=1}^{N} \prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta) \sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{1\le\alpha<\beta\le N} \sinh(\lambda_\beta - \lambda_\beta) \prod_{1\le\kappa<j\le N} \sinh(\nu_\kappa - \nu_j)} \times \prod_{j=2}^{N} \sinh(\nu_1 - \nu_j)
$$

Multiplying this result by

$$
\frac{\prod_{\alpha=1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \sinh(\lambda_\alpha - \nu_1 - \eta)}{\prod_{\alpha=1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \sinh(\lambda_\alpha - \nu_1 - \eta)}
$$

and changing the product over κ in the numerator to start from $\kappa = 1$. The second

term of (3.4.11) becomes:

$$\frac{\prod_{\alpha=1}^{N}\prod_{\kappa=1}^{N}\sinh(\lambda_\alpha-\nu_\kappa+\eta)\sinh(\lambda_\alpha-\nu_\kappa-\eta)}{\prod_{1\le\alpha<\beta\le N}\sinh(\lambda_\beta-\lambda_\beta)\prod_{1\le\kappa<j\le N}\sinh(\nu_\kappa-\nu_j)} \tag{3.4.13}$$

$$\times\frac{\prod_{j=2}^{N}\sinh(\nu_1-\nu_j)}{\prod_{\alpha=1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)\sinh(\lambda_\alpha-\nu_1-\eta)} \tag{3.4.14}$$

Which by (3.2.7) is:

$$\frac{\prod_{j=2}^{N}\sinh(\nu_1-\nu_j)}{\prod_{\alpha=1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)\sinh(\lambda_\alpha-\nu_1-\eta)}\times\frac{1}{\det_N\mathcal{Z}} \tag{3.4.15}$$

Consider the first two terms of (3.4.11) now that we have simplified them:

$$\frac{\sinh(2\eta)\prod_{\alpha=1}^{M-1}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=M+1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)}{\prod_{\alpha=1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)\sinh(\lambda_\alpha-\nu_1-\eta)}\times\frac{\prod_{j=2}^{N}\sinh(\nu_1-\nu_j)}{\det_N\mathcal{Z}}$$

Splitting the products in the denominator:

$$=\frac{\sinh(2\eta)\prod_{\alpha=1}^{M-1}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=M+1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)}{\prod_{\alpha=1}^{N}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=1}^{N}\sinh(\lambda_\alpha-\nu_1+\eta)}\times\frac{\prod_{j=2}^{N}\sinh(\nu_1-\nu_j)}{\det_N\mathcal{Z}}$$

Many of the terms in the products will cancel and we may rewrite the terminals as follows:

$$=\frac{\sinh(2\eta)}{\prod_{\alpha=M}^{N}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=1}^{M}\sinh(\lambda_\alpha-\nu_1+\eta)}\times\frac{\prod_{j=2}^{N}\sinh(\nu_1-\nu_j)}{\det_N\mathcal{Z}}$$

$$= \frac{\sinh(2\eta)\prod_{j=2}^{N}\sinh(\nu_1-\nu_j)}{\prod_{\alpha=M}^{N}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=1}^{M}\sinh(\lambda_\alpha-\nu_1+\eta)} \times \frac{1}{\det_N \mathcal{Z}} \tag{3.4.16}$$

Now consider the third term of (3.4.11). We can see that this is a determinant of some $N \times N$ matrix evaluated along the first column. The entries of the $N \times N$ matrix $\mathcal{H}$ are:

$$\mathcal{H}_{\alpha 1} = h_M(\lambda_\alpha), \quad \mathcal{H}_{\alpha\kappa} = \phi(\lambda_\alpha, \nu_\kappa), \quad \kappa = 2 \dots N.$$

and

$$h_M(\lambda_\alpha) = \frac{\prod_{\gamma=1}^{M-1}\sinh(\lambda_\gamma-\lambda_\alpha+2\eta)\prod_{\gamma=M+1}^{N}\sinh(\lambda_\gamma-\lambda_\alpha)}{\prod_{j=2}^{N}\sinh(\lambda_\alpha-\nu_\kappa-\eta)}$$

Note that $h_M(\lambda_\alpha) = 0$ for $\alpha > M$, as the product $\prod_{M+1}^{N}\sinh(\lambda_\gamma - \lambda_\beta)$ would always have a zero when $\lambda_\gamma = \lambda_\alpha$, which would be every term of the sum for $\beta = 1 \dots N$. So when we include it in the determinant sum, we only need the first M terms.

The product of (3.4.16) and our new expression $\det_N \mathcal{H}$, gives us a simplified expressions for (3.4.11):

$$\frac{\sinh(2\eta)\prod_{j=2}^{N}\sinh(\nu_1-\nu_j)}{\prod_{\alpha=M}^{N}\sinh(\lambda_\alpha-\nu_1-\eta)\prod_{\alpha=1}^{M}\sinh(\lambda_\alpha-\nu_1+\eta)} \times \frac{\det_N \mathcal{H}}{\det_N \mathcal{Z}}$$

Which is the expression of $H_N^{(M)}$ using only determinants. We call this, $H_N^{(M)}$ in determinant form.

3.4.3 $G_N^{(M)}$ in Determinant Form

Now we wish to derive an expression for G^M in determinant form. We do this the same way that we found H^M. Substitute the simplified expression for Z_{N-1} into

(3.3.16)

$$\begin{aligned}
G_N^{(M)} =& Z_N^{-1} \prod_{\alpha=1}^{M} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \times \sum_{\beta=1}^{M} \prod_{\alpha=2}^{N} \sinh(\lambda_\beta - \nu_\alpha + \eta) \frac{\sinh(2\eta)}{\sinh(\lambda_\beta - \nu_1 - \eta)} \\
& \times \prod_{\substack{\gamma=1 \\ \gamma\neq\beta}}^{M} \frac{\sinh(\lambda_\gamma - \lambda_\beta + 2\eta)}{\sinh(\lambda_\beta - \lambda_\gamma)} \\
& \times \frac{\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta) \sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{1\leq j<i\leq N} \sinh(\lambda_i - \lambda_j) \prod_{2\leq\kappa<j\leq N} \sinh(\nu_\kappa - \nu_j)} \\
& \times (-1)^{\beta-1} \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \sinh(\lambda_\alpha - \lambda_\beta) \Delta_{N-1}^{(\beta)}
\end{aligned}$$

To simplify this expression, cancel the denominator of the third line with part of the the product in the fifth line; bring the $\sinh(2\eta)$ inside the sum over γ in the third line to make it over all values:

$$\begin{aligned}
G_N^{(M)} =& Z_N^{-1} \prod_{\alpha=1}^{M} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \times \sum_{\beta=1}^{M} \prod_{\alpha=2}^{N} \sinh(\lambda_\beta - \nu_\alpha + \eta) \frac{1}{\sinh(\lambda_\beta - \nu_1 - \eta)} \\
& \times \prod_{\gamma=1}^{M} \sinh(\lambda_\gamma - \lambda_\beta + 2\eta) \\
& \times \frac{\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{N} \prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta) \sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{1\leq j<i\leq N} \sinh(\lambda_i - \lambda_j) \prod_{2\leq\kappa<j\leq N} \sinh(\nu_\kappa - \nu_j)} \\
& \times (-1)^{\beta-1} \Delta_{N-1}^{(\beta)} \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \lambda_\beta)
\end{aligned}$$

We can take the products in the denominator of the fourth term out of the summation; the numerator of the fifth term, over all values, which means we must leaving an additional product by (3.4.1):

$$
\begin{aligned}
G_N^{(M)} =& Z_N^{-1} \prod_{\alpha=1}^{M} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \times \frac{\prod_{\alpha=1}^{N} \prod_{\kappa=2}^{N} \sinh(\lambda_\alpha - \nu_\kappa + \eta) \sinh(\lambda_\alpha - \nu_\kappa - \eta)}{\prod_{1 \le j < i \le N} \sinh(\lambda_i - \lambda_j) \prod_{2 \le \kappa < j \le N} \sinh(\nu_\kappa - \nu_j)} \\
& \times \sum_{\beta=1}^{M} \prod_{\alpha=2}^{N} \sinh(\lambda_\beta - \nu_\alpha + \eta) \frac{1}{\sinh(\lambda_\beta - \nu_1 - \eta)} \\
& \times \prod_{\gamma=1}^{M} \sinh(\lambda_\gamma - \lambda_\beta + 2\eta) \\
& \times (-1)^{\beta-1} \Delta_{N-1}^{(\beta)} \times \frac{\prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \lambda_\beta)}{\prod_{k=2}^{N} \sinh(\lambda_\beta - \nu_k + \eta) \sinh(\lambda_\beta - \nu_k - \eta)}
\end{aligned}
$$

Cancel the first product in the sum, with part of the product in the denominator of fifth line; bring $1/\sinh(\lambda_\beta - \nu_k - \eta)$ into the remainder of the denominator of the same product on the fifth line so that it is a sum over $k = 1 \dots M$ rather than $k = 2 \dots M$; and use steps (3.4.12)- (3.4.14) to write the second line as the RHS of (3.4.15):

$$
\begin{aligned}
G_N^{(M)} =& Z_N^{-1} \prod_{\alpha=1}^{M} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \\
& \frac{\prod_{j=2}^{N} \sinh(\nu_1 - \nu_j)}{\prod_{\alpha=1}^{N} \sinh(\lambda_\alpha - \nu_1 + \eta) \sinh(\lambda_\alpha - \nu_1 - \eta)} \times \frac{1}{\det_N \mathcal{Z}} \\
& \times \sum_{\beta=1}^{M} (-1)^{\beta-1} \Delta_{N-1}^{(\beta)} \times \frac{\prod_{\gamma=1}^{M} \sinh(\lambda_\gamma - \lambda_\beta + 2\eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \lambda_\beta)}{\prod_{k=1}^{N} \sinh(\lambda_\beta - \nu_k - \eta)}
\end{aligned}
$$

Canceling similar terms in the denominator of the second term and numerator of

first term leaves us with:

$$G_N^{(M)} = Z_N^{-1} \frac{\prod_{j=2}^{N} \sinh(\nu_1 - \nu_j)}{\prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=1}^{M} \sinh(\lambda_\alpha - \nu_1 + \eta)} \frac{1}{\det_N \mathcal{Z}}$$
$$\times \sum_{\beta=1}^{M} (-1)^{\beta-1} \Delta_{N-1}^{(\beta)} \times \frac{\prod_{\gamma=1}^{M} \sinh(\lambda_\gamma - \lambda_\beta + 2\eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \lambda_\beta)}{\prod_{k=1}^{N} \sinh(\lambda_\beta - \nu_k - \eta)}$$

Using the same logic as we used for $H_N^{(M)}$, we can represent the sum as the determinant of an $N-1 \times N-1$ matrix, $\mathcal{G}$, so that:

$$G_N^{(M)} = Z_N^{-1} \frac{\prod_{j=2}^{N} \sinh(\nu_1 - \nu_j)}{\prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \nu_1 - \eta) \prod_{\alpha=1}^{M} \sinh(\lambda_\alpha - \nu_1 + \eta)} \frac{\det_N \mathcal{G}}{\det_N \mathcal{Z}}$$

Where the entries of matrix $\mathcal{G}$ are given by

$$\mathcal{G}_{\alpha 1} = g_M(\lambda) \quad \mathcal{G}_{\alpha k} = \phi(\lambda_\alpha, \nu_k), \quad k = 2 \dots N$$

and:

$$g_M(\lambda) = \frac{\prod_{\gamma=1}^{M} \sinh(\lambda_\gamma - \lambda + 2\eta) \prod_{\alpha=M+1}^{N} \sinh(\lambda_\alpha - \lambda)}{\prod_{k=1}^{N} \sinh(\lambda - \nu_k - \eta)}$$

As with $H_N^{(M)}$, the zeros of $g_N(\lambda)$ occur at $\lambda_{M+1} \dots \lambda_N$, meaning the last$N - M$ terms of the first column of $\mathcal{G}$ are zero.

3.5 The Free Fermion Case

3.5.1 Definition and Inhomogeneous Correlation Function $H_N^{(M)}$

We will now consider an application, and evaluate $H_N^{(M)}$ for this particular problem. The free fermion case places the following restriction on the vertex weights [4]

$$a^2(\lambda_\alpha, \nu_\kappa) + b^2(\lambda_\alpha, \nu_\kappa) = c^2(\lambda_\alpha, \nu_\kappa) \quad \alpha, \kappa = 1 \dots N$$

[4] Bogoliubov, N. M.; Pronko, A. G.; Zvonarev, M. B, Boundary correlation functions of the six-vertex model, 13

The equation is satisfied if we let $\eta = i\frac{\pi}{4}$, let $\lambda_\alpha \to i\lambda_\alpha$ and $\nu_\kappa \to i\nu_\kappa$ and rescale $a \to -ia, b \to -ib, c \to -ic$:

$$\begin{aligned}
\sinh^2(\lambda_\alpha - \nu_\kappa + \eta) + \sinh^2(\lambda_\alpha - \nu_\kappa - \eta) &= \sinh^2(2\eta) \\
(-i\sinh(i\lambda_\alpha - i\nu_\kappa + i\pi/4))^2 + (-i\sinh(i\lambda_\alpha - i\nu_\kappa - i\pi/4))^2 &= (-i\sinh(i\pi/2))^2 \\
\sin^2(\lambda_\alpha - \nu_\kappa + \pi/4) + \sin^2(\lambda_\alpha - \nu_\kappa - \pi/4) &= \sin^2(\pi/2) \\
\sin^2(\lambda_\alpha - \nu_\kappa + \pi/4) + \sin^2(\lambda_\alpha - \nu_\kappa - \pi/4) &= 1 \\
\sin^2(\lambda'_\alpha - \nu'_\kappa) + \cos^2(\lambda'_\alpha - \nu'_\kappa) &= 1
\end{aligned}$$

Where we have used $-i\sinh(ix) = \sin(x)$, and made the substitution $\lambda'_\alpha - \nu'_\kappa = \lambda_\alpha - \nu_\kappa + \pi/4$ to satisfy the equation. Hence we have:

$$\begin{aligned}
a(\lambda_\alpha, \nu_\kappa) &= \sin(\lambda_\alpha - \nu_\kappa + \frac{\pi}{4}) \\
b(\lambda_\alpha, \nu_\kappa) &= \sin(\lambda_\alpha - \nu_\kappa - \frac{\pi}{4}) \\
c(\lambda_\alpha, \nu_\kappa) &= 1
\end{aligned}$$

Under this condition, the determinant in equation (3.2.7) becomes the Cauchy determinant, and we can evaluate the partition function explicitly:[5]

$$Z_N = \prod_{1 \le \alpha < \beta \le N} \cos(\lambda_\alpha - \lambda_\beta) \prod_{1 \le k < j \le N} \cos(\nu_k - \nu_j)$$

We can then evaluate the correlation function $H_N^{(M)}$. Substituting this expression for Z_N into (3.3.12) one can obtain, using relations (3.4.1), (3.4.8) and (3.4.9):

$$H_N^{(M)} = \frac{\prod_{\alpha=1}^{M-1} \sin(\lambda_\alpha - \nu_1 - \frac{\pi}{4}) \prod_{\alpha=M+1}^{N} \sin(\lambda_\alpha - \nu_1 + \frac{\pi}{4})}{\prod_{j=1}^{N} \cos(\nu_j - \nu_1)} \tag{3.5.1}$$

$$\times \sum_{\beta=1}^{M} \frac{\prod_{j=2}^{N} \sin(\lambda_\beta - \nu_j + \frac{\pi}{4})}{\prod_{\alpha=M}^{N} \cos(\lambda_\alpha - \lambda_\beta)} \prod_{\substack{\alpha=1 \\ \alpha \neq \beta}}^{M} \frac{1}{\sin(\lambda_\alpha - \lambda_\beta)} \tag{3.5.2}$$

[5] Bogoliubov, N. M.; Pronko, A. G.; Zvonarev, M. B, Boundary correlation functions of the six-vertex model, 14

Which represents the inhomogeneous correlation function $H_N^{(M)}$.[6]

3.5.2 The Homogeneous Limit of $H_N^{(M)}$

We will now evaluate the homogeneous limit of this our expression for $H_N^{(M)}$ in the free fermion case. The homogeneous limit is letting all our parameters become constants: $\lambda_N = \ldots = \lambda_1 \equiv \lambda$ and $\nu_N = \ldots = \nu_1 \equiv \nu$, so that our vertex weights only depend on a constant; $\lambda - \nu$. Letting $\lambda - \nu + \frac{\pi}{4} \to \lambda$ we have the following vertex weights:

$$\begin{aligned} a(\lambda) &= \sin(\lambda + \frac{\pi}{2}) = \cos(\lambda) \\ b(\lambda) &= \sin(\lambda) \\ c(\lambda) &= 1 \end{aligned}$$

Note that the partition function of the homogeneous model becomes $Z_N = 1$.

If we use set all ν and λ to be equivalent at once, (3.5.2) becomes ∞ which is not helpful. This is why we must take the limit. We evaluate $H_N^{(M)}$ by firstly, setting ν to be a constant, keeping λ_α a variable and then taking the limit as $\lambda_\alpha \to \lambda$. Start by letting $\nu_\kappa = \nu = \frac{-\pi}{4}$, and put this into (3.5.2):

$$H_N^{(M)} = \frac{\prod_{\alpha=1}^{M-1} \sin(\lambda) \prod_{\alpha=M+1}^{N} \cos(\lambda_\alpha)}{\prod_{j=1}^{N} \cos(0)} \times \sum_{\beta=1}^{M} \frac{\prod_{j=2}^{N} \cos(\lambda_\beta)}{\prod_{\alpha=M}^{N} \cos(\lambda_\alpha - \lambda_\beta)} \prod_{\substack{\alpha=1 \\ \alpha \neq \beta}}^{M} \frac{1}{\sin(\lambda_\alpha - \lambda_\beta)}$$

$$= \prod_{\alpha=1}^{M-1} \sin(\lambda) \prod_{\alpha=M+1}^{N} \cos(\lambda_\alpha) \times \sum_{\beta=1}^{M} \frac{\cos^{N-1}(\lambda_\beta)}{\prod_{\alpha=M}^{N} \cos(\lambda_\alpha - \lambda_\beta)} \prod_{\substack{\alpha=1 \\ \alpha \neq \beta}}^{M} \frac{1}{\sin(\lambda_\alpha - \lambda_\beta)}$$

We may express the product of $\cos(\lambda_j)$ in the form of a power, because the product

[6] Although we rushed the derivation of $H_N^{(M)}$, this was because we want to focus on the evaluation of the homogeneous limit

in independent of β. Multiplying by $\frac{\prod_{\alpha=1}^{N}\cos(\lambda_\alpha)}{\prod_{\alpha=1}^{N}\cos(\lambda_\alpha)}$ gives:

$$H_N^{(M)} = \frac{\prod_{\alpha=1}^{M-1}\sin(\lambda)\prod_{\alpha=M+1}^{N}\cos(\lambda_\alpha)}{\prod_{\alpha=1}^{N}\cos(\lambda_\alpha)} \times \sum_{\beta=1}^{M}\frac{\cos^{N-1}(\lambda_\beta)\prod_{\alpha=1}^{N}\cos(\lambda_\alpha)}{\prod_{\alpha=M}^{N}\cos(\lambda_\alpha-\lambda_\beta)}\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{M}\frac{1}{\sin(\lambda_\alpha-\lambda_\beta)}$$

$$= \prod_{\alpha=1}^{M-1}\frac{\sin(\lambda_\alpha)}{\cos(\lambda_\alpha)} \times \frac{1}{\cos(\lambda_M)} \times \sum_{\beta=1}^{M}\frac{\prod_{\alpha=N}^{M}\cos(\lambda_\alpha)\cos(\lambda_\beta)\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{M}\cos(\lambda_\alpha)\cos(\lambda_\beta)}{\prod_{\alpha=M}^{N}\cos(\lambda_\alpha-\lambda_\beta)}$$

$$\times \prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{M}\frac{1}{\sin(\lambda_\alpha-\lambda_\beta)}$$

Consider the numerator of the first factor in the sum over β:

$$\cos^{N-1}(\lambda_\beta)\prod_{\alpha=1}^{N}\cos(\lambda_\alpha) = \prod_{\alpha=1}^{N-1}\cos(\lambda_\beta)\times\cos(\lambda_\beta)\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{N}\cos(\lambda_\alpha)$$

$$= \prod_{\alpha=1}^{N}\cos(\lambda_\beta)\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{N}\cos(\lambda_\alpha)$$

$$= \prod_{\alpha=1}^{N}\cos(\lambda_\beta)\frac{1}{\cos(\lambda_M)}\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{M}\cos(\lambda_\alpha)\prod_{\alpha=M}^{N}\cos(\lambda_\alpha)$$

$$= \frac{1}{\cos(\lambda_M)}\prod_{\substack{\alpha=1\\\alpha\neq\beta}}^{M}\cos(\lambda_\alpha)\cos(\lambda_\beta)\prod_{\alpha=M}^{N}\cos(\lambda_\alpha)\cos(\lambda_\beta)$$

Putting this back into our expression for $H_N^{(M)}$ gives:

$$= \prod_{\alpha=1}^{M-1} \frac{\sin(\lambda_\alpha)}{\cos(\lambda_\alpha)} \times \frac{1}{\cos^2(\lambda_M)} \times \sum_{\beta=1}^{M} \frac{\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} \cos(\lambda_\alpha)\cos(\lambda_\beta) \prod_{\alpha=M}^{N} \cos(\lambda_\alpha)\cos(\lambda_\beta)}{\prod_{\alpha=M}^{N} \cos(\lambda_\alpha - \lambda_\beta)}$$

$$\times \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} \frac{1}{\sin(\lambda_\alpha - \lambda_\beta)}$$

$$= \prod_{\alpha=1}^{M-1} \frac{\sin(\lambda_\alpha)}{\cos(\lambda_\alpha)} \times \frac{1}{\cos^2(\lambda_M)} \times \sum_{\beta=1}^{M} \frac{1}{\prod_{\alpha=M}^{N} \frac{\cos(\lambda_\alpha - \lambda_\beta)}{\cos(\lambda_\alpha)\cos(\lambda_\beta)}} \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} \frac{1}{\frac{\sin(\lambda_\alpha - \lambda_\beta)}{\cos(\lambda_\alpha)\cos(\lambda_\beta)}} \tag{3.5.3}$$

Using the standard identities:

$$\sin(x-y) = \sin(x)\cos(y) - \cos(x)\sin(y)$$
$$\cos(x-y) = \cos(x)\cos(y) + \sin(x)\sin(y)$$

we can simplify (3.5.3) to:

$$= \prod_{\alpha=1}^{M-1} \frac{\sin(\lambda_\alpha)}{\cos(\lambda_\alpha)} \times \frac{1}{\cos^2(\lambda_M)} \times \sum_{\beta=1}^{M} \frac{1}{\prod_{\alpha=M}^{N} \frac{\cos(\lambda_\alpha)\cos(\lambda_\beta) - \sin(\lambda_\alpha)\sin(\lambda_\beta)}{\cos(\lambda_\alpha)\cos(\lambda_\beta)}}$$

$$\times \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} \frac{1}{(\frac{\sin(\lambda_\alpha)\cos(\lambda_\beta) - \cos(\lambda_\alpha)\sin(\lambda_\beta)}{\cos(\lambda_\alpha)\cos(\lambda_\beta)})}$$

$$= \prod_{\alpha=1}^{M-1} \tan(\lambda_\alpha) \times \frac{1}{\cos^2(\lambda_M)} \times \sum_{\beta=1}^{M} \frac{1}{\prod_{\alpha=M}^{N} 1 - \tan(\lambda_\alpha)\tan(\lambda_\beta)}$$

$$\times \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} \frac{1}{\tan(\lambda_\alpha) - \tan(\lambda_\beta)}$$

Making the substitution $\tan(\lambda_\gamma) = u_\gamma$:

$$H_N^{(M)} = (1+u_\alpha^2) \prod_{\alpha=1}^{M-1} u_\alpha \times \sum_{\beta=1}^{M} \prod_{\alpha=M}^{N} \frac{1}{1+u_\alpha u_\beta} \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} \frac{1}{u_\alpha - u_\beta}$$

Now the homogenous limit is equivalent to letting $u_\alpha \to u$. If we do this we will still have a singularity in the second product. To resolve this, we will only take the limit of some u_α terms and, for now, let the rest remain variables:

$$\begin{aligned} u_\alpha &\to u, & \alpha &= M \dots N \\ u_\alpha &\to u - (M-\alpha)\epsilon, & \alpha &= 1 \dots M-1 \end{aligned}$$

where the homogeneous case is equivalent to $\epsilon \to 0$. Our expression for $H_N^{(M)}$ becomes:

$$\begin{aligned} H_N^{(M)} =&(1+u^2)\prod_{\alpha=1}^{M-1}(u-(M-\alpha)\epsilon) \times \sum_{\beta=1}^{M}\prod_{\alpha=M}^{N}\frac{1}{(1+u(u-(M-\beta))\epsilon)} \\ &\times \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M}\frac{1}{u-(M-\beta)\epsilon-u-(M+\alpha)\epsilon} \\ =&(1+u^2)\prod_{\alpha=1}^{M-1}(u-(M-\alpha)\epsilon) \times \sum_{\beta=1}^{M}\frac{1}{(1+u(u-(M-\beta))\epsilon)^{N-M+1}} \\ &\times \prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M}\frac{1}{(\alpha-\beta)\epsilon} \end{aligned}$$

We can take $\epsilon \to 0$ in the first product, and take out ϵ^{N-M} in the final product to give

$$\begin{aligned} H_N^{(M)} =&(1+u^2)u^{M-1} \times \sum_{\beta=1}^{M}\frac{1}{(1+u(u-(M-\beta))\epsilon)^{N-M+1}} \times \frac{1}{\epsilon^{M-1}}\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M}\frac{1}{(\alpha-\beta)} \\ =&(1+u^2)u^{M-1}\sum_{\beta=1}^{M}\frac{1}{[1+u^2-u(M-\beta)\epsilon]^{N-M+1}\epsilon^{M-1}}\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M}\frac{1}{\alpha-\beta} \end{aligned}$$

Summing over $M-\beta = M-1, M-2, \dots 0$ is the same as summing from $\beta' = 0, 1, \dots M-1$. Which we can rewrite as:

$$H_N^{(M)} =(1+u^2)u^{M-1}\sum_{\beta=0}^{M-1}\frac{(-1)^{M-1}}{[1+u^2-u\beta\epsilon]^{N-M+1}\epsilon^{M-1}}\prod_{\substack{\alpha=0 \\ \alpha\neq\beta}}^{M-1}\frac{1}{\alpha-\beta}$$

We now use the binomial identity:

$$\prod_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M-1}\frac{1}{\alpha-\beta} = \frac{(-1)^\beta}{\beta!(M-\beta-1)!} = \frac{(-1)^\beta}{(M-1)!}\binom{M-1}{\beta}$$

Our expression for $H_N^{(M)}$ becomes:

$$\begin{aligned}
H_N^{(M)} =&(1+u^2)u^{M-1}\sum_{\beta=0}^{M-1}\frac{(-1)^{M-1}}{[1+u^2-u\beta\epsilon]^{N-M+1}\epsilon^{M-1}}\frac{(-1)^\beta}{(M-1)!}\binom{M-1}{\beta}\\
=&\frac{(1+u^2)u^{M-1}(-1)^{M-1}}{(M-1)!}\sum_{\beta=0}^{M-1}\frac{(-1)^\beta}{[u(\frac{1}{u}(1+u^2)-\beta\epsilon)]^{N-M+1}\epsilon^{M-1}}\binom{M-1}{\beta}\\
=&\frac{(1+u^2)u^{M-1}(-1)^{M-1}}{(M-1)!u^{N-M+1}}\sum_{\beta=0}^{M-1}\frac{(-1)^\beta}{[u^{-1}(1+u^2)-\beta\epsilon]^{N-M+1}\epsilon^{M-1}}\binom{M-1}{\beta}\\
=&\frac{(1+u^2)(-1)^{M-1}}{(M-1)!u^{N-2M+2}}\sum_{\beta=0}^{M-1}\frac{(-1)^\beta}{[u^{-1}(1+u^2)-\beta\epsilon]^{N-M+1}\epsilon^{M-1}}\binom{M-1}{\beta}
\end{aligned}$$

Now we make use of the binomial formula for the M-th derivative[7]:

$$\lim_{\epsilon\to 0}\sum_{\beta=0}^{M-1}\frac{(-1)^\beta f(z-\beta\epsilon)}{\epsilon^{M-1}}\binom{M-1}{\beta}=\frac{\mathrm{d}^{M-1}}{\mathrm{d}z^{M-1}}f(z)$$

Where:

$$f(z)=\frac{1}{z^{N-M+1}},\quad z=\frac{1+u^2}{u}$$

We can rewrite$H_N^{(M)}$:

$$H_N^{(M)}=\frac{(1+u^2)(-1)^{M-1}}{(M-1)!u^{N-2M+2}}\frac{\mathrm{d}^{M-1}}{\mathrm{d}z^{M-1}}z^{N+M-1}$$

The M-1-th derivitive of $f(z)$ is:

$$\begin{aligned}
\frac{\mathrm{d}^{M-1}}{\mathrm{d}z^{M-1}}z^{-(N-M+1)}=&(-1)^{M-1}[N-(M-1)][N-(M-2)]\dots[N-1]N\\
=&\frac{(-1)^{M-1}(N-1)!}{(N-M)!}z^{-N}\\
=&\frac{(-1)^{M-1}(N-1)!}{(N-M)!}(\frac{u}{1+u^2})^N
\end{aligned}$$

[7]Bogoliubov, N. M.; Pronko, A. G.; Zvonarev, M. B, Boundary correlation functions of the six-vertex model

Subbing this into or expression for $H_N^{(M)}$ gives:

$$\begin{aligned} H_N^{(M)} &= \frac{(1+u^2)(-1)^{M-1}}{(M-1)!u^{N-2M+2}} \frac{(-1)^{M-1}(N-1)!}{(N-M)!} (\frac{u}{1+u^2})^N \\ &= \frac{(1+u^2)u^N}{u^{N-2M+2}(1+u^2)^N} \frac{(N-1)!}{(M-1)!(N-M)!} \\ &= \frac{u^{2(M-1)}}{(1+u^2)^F N-1} \binom{N-1}{M-1} \\ &= \frac{\frac{\sin^2(\lambda)}{\cos^2(\lambda)}^{(M-1)}}{(1+\frac{\sin^2(\lambda)}{\cos^2(\lambda)})^{N-1}} \binom{N-1}{M-1} \end{aligned}$$

Multiplying through by $(\cos^2(\lambda))^{N-1}$ leaves us with the result:

$$\begin{aligned} H_N^{(M)} &= \frac{\frac{\sin^2(\lambda)}{\cos^2(\lambda)}^{M-1} (\cos^2(\lambda))^{N-1}}{(cos^2(\lambda)+\sin^2(\lambda))^{N-1}} \binom{N-1}{M-1} \\ &= [\sin^2(\lambda)]^{M-1}[\cos^2(\lambda)]^{(N-1)-(M-1)} \binom{N-1}{M-1} \\ &= [\sin^2(\lambda)]^{M-1}[\cos^2(\lambda)]^{N-M} \binom{N-1}{M-1} \end{aligned}$$

This is the expression for the homogenous limit of the free fermion case.

3.6 Summary and Conclusion

In chapter 1, we introduced the theory of vertex models in statistical mechanics, and explained the graphic representations of partition functions. We considered an $M \times N$ lattice with periodic boundary conditions and found a representation for the partition function using a transfer matrix. We then introduced the 8-vertex model and found a solution to the Yang-Baxter equations in terms of Boltzmann weights that are parameterized by elliptic function.

In chapter 2, we considered the special case of the 6-vertex model, which let Boltzmann weights become parameterized by hyperbolic trigonometric functions. We then constructed algebraic expressions for the eigenvectors of the transfer matrix,

which we could then use to evaluate the partition function in n the homogeneous case (where all vertical rapidities are equal, and all horizontal rapidities are also equal).

In chapter 3, we considered moved to the 6-vertex model in inhomogeneous case (all rapidities are now independent variables) on an $N \times N$ square lattice with domain wall boundary conditions. We then used the determinant form of the partition function to calculate the values of two kinds of boundary 1-point function in the presence of domain wall boundary conditions, and found the exact value of one of these in the homogenous limit for the free fermion case.

Clearly, the algebraic Bethe ansatz (ABA) is a powerful tool for finding exact solutions of physical models and computing physical quantities. This is was only the tip of what is actually a very vast subject.

Bibliography

[1] Baxter, Rodney. J, *Exactly Solved Models in Statistical Mechanics,* (Academic Press, 1982).

[2] Takhatajan, L. J. Introduction to Agebraic Bethe Ansatz, *Lecture Notes in Physics, Exactly solvable problems in condensed matter and relativistic field theory, 242*, (Panchgani, India: Springer-Verlag, 1982)

[3] Bogoliubov, N. M.; Pronko, A. G.; Zvonarev, M. B, Boundary correlation functions of the six-vertex model *J. Phys. A: Math. Gen. 35, 5525-5541 (2002)*