Computing scalar products in the Bethe Ansatz

Juan Miguel Nieto García

University of Surrey

LMS lectures
October 2020

Scalar products, why do we need them?

Given a quantum integrable system, there exists a set of well defined methods to simultaneously diagonalize all the conserved charges of the system. These are the different versions of the Bethe asantz (coordinate, algebraic, analytic, nested, off-diagonal...)

Scalar products, why do we need them?

Given a quantum integrable system, there exists a set of well defined methods to simultaneously diagonalize all the conserved charges of the system. These are the different versions of the Bethe asantz (coordinate, algebraic, analytic, nested, off-diagonal...)
All these methods provide us not only with the complete spectrum but also with a recipe to systematically construct the eigenstates of the tower of conserved charges.

Scalar products, why do we need them?

Given a quantum integrable system, there exists a set of well defined methods to simultaneously diagonalize all the conserved charges of the system. These are the different versions of the Bethe asantz (coordinate, algebraic, analytic, nested, off-diagonal...)
All these methods provide us not only with the complete spectrum but also with a recipe to systematically construct the eigenstates of the tower of conserved charges.
However, the computation of correlation functions and expectation values is muddied due to the way the eigenstates are constructed.

Scalar products, why do we need them?

Given a quantum integrable system, there exists a set of well defined methods to simultaneously diagonalize all the conserved charges of the system. These are the different versions of the Bethe asantz (coordinate, algebraic, analytic, nested, off-diagonal...)

All these methods provide us not only with the complete spectrum but also with a recipe to systematically construct the eigenstates of the tower of conserved charges.

However, the computation of correlation functions and expectation values is muddied due to the way the eigenstates are constructed.

As these quantities can be related to different kinds of scalar products, we are interested in obtaining simple expressions for such scalar products.

Scalar products, why do we need them?

Given a quantum integrable system, there exists a set of well defined methods to simultaneously diagonalize all the conserved charges of the system. These are the different versions of the Bethe asantz (coordinate, algebraic, analytic, nested, off-diagonal...)

All these methods provide us not only with the complete spectrum but also with a recipe to systematically construct the eigenstates of the tower of conserved charges.

However, the computation of correlation functions and expectation values is muddied due to the way the eigenstates are constructed.

As these quantities can be related to different kinds of scalar products, we are interested in obtaining simple expressions for such scalar products.

I will be focusing in the case of the Algebraic Bethe Ansatz, but I will also dedicate some time at the beginning to the Coordinate version.

Outline

Coordinate Bethe ansatz for the Lieb-Liniger model

A bit of history, the Lieb-Liniger model

The original version of the Bethe ansatz [Hans Bethe, 1931], known today as the Coordinate Bethe Ansatz, was a method to find the spectrum and the eigenstates of the "linear atomic chain" (known today as the Heisenberg Hamiltonian).

A bit of history, the Lieb-Liniger model

The original version of the Bethe ansatz [Hans Bethe, 1931], known today as the Coordinate Bethe Ansatz, was a method to find the spectrum and the eigenstates of the "linear atomic chain" (known today as the Heisenberg Hamiltonian).
However, I am more interested in another model, the one dimensional gas of interacting bosons, described by

$$\sum_j \frac{\partial^2 \psi}{\partial x_j^2} + 2c \sum_{i<j} \delta(x_i - x_j)\psi = E\psi \ .$$

This model is also known as the Lieb-Liniger model, as they were the first to solve it thanks to Bethe's method [Lieb, Liniger, 1963].

The basic intuition

The basic idea goes as follows: if we have only two bosons, they behave like free particles unless they meet, so we can first separate the wave function into four terms

$$\psi_{k_1,k_2}(x) = Ae^{i(k_1x_1+p_2x_2)}\theta(x_1-x_2) + Be^{i(k_1x_2+k_2x_1)}\theta(x_1-x_2) \\ + Be^{i(k_1x_1+k_2x_2)}\theta(x_2-x_1) + Ae^{i(k_1x_2+k_2x_1)}\theta(x_2-x_1) \ ,$$

where θ is the Heaviside function. After substituting and solving the Schrödinger equation, we get $\frac{A}{B} = \frac{k_1-k_2+ic}{k_1-k_2-ic}$ and $E = k_1^2 + k_2^2$.

The basic intuition

The basic idea goes as follows: if we have only two bosons, they behave like free particles unless they meet, so we can first separate the wave function into four terms

$$\psi_{k_1,k_2}(x) = Ae^{i(k_1x_1+p_2x_2)}\theta(x_1-x_2) + Be^{i(k_1x_2+k_2x_1)}\theta(x_1-x_2) \\ + Be^{i(k_1x_1+k_2x_2)}\theta(x_2-x_1) + Ae^{i(k_1x_2+k_2x_1)}\theta(x_2-x_1) \ ,$$

where θ is the Heaviside function. After substituting and solving the Schrödinger equation, we get $\frac{A}{B} = \frac{k_1-k_2+ic}{k_1-k_2-ic}$ and $E = k_1^2 + k_2^2$.

The procedure is the same for any number of excitations, so

$$\psi_{\{k\}}(x) = \sum_{\sigma\in S_N} \left(\prod_{i<j} \frac{k_{\sigma(i)} - k_{\sigma(j)} + ic}{k_{\sigma(i)} - k_{\sigma(j)}} \right) \exp\left[i \sum_j x_j k_{\sigma(j)} \right] \ ,$$

for the ordering $x_i < x_j$ if $i < j$ (the others are obtained from symmetry), and $E = \sum k_i^2$.

Periodicity and BAE

The final step is to put everything in a box (because we want our things well quantized) with periodic boundary conditions. This means

$$\psi_{\{k\}}(x_1, x_2, \ldots, x_N) = \psi_{\{k\}}(x_2, \ldots, x_N, x_1 + L) \ ,$$

because of the ordering condition on our wavefunction.

Periodicity and BAE

The final step is to put everything in a box (because we want our things well quantized) with periodic boundary conditions. This means

$$\psi_{\{k\}}(x_1, x_2, \ldots, x_N) = \psi_{\{k\}}(x_2, \ldots, x_N, x_1 + L) \ ,$$

because of the ordering condition on our wavefunction.

Then, the periodicity condition imposes the following condition on the momenta

$$e^{ik_1 L} = \prod_{j \neq 1} \frac{k_1 - k_j + ic}{k_1 - k_j - ic} \ ,$$

and similarly for the other momenta. These are called Bethe Ansatz Equations (BAE).

Normalization of eigenstates [M. Gaudin, 1972]

However, these wave functions are not normalized, and the amount of terms ($N!$) makes the direct computation of scalar products tricky.

M. Gaudin proposed the following normalization for this particular wave function

$$\int_D \frac{1}{(2\pi)^N G(k)} \psi^*_{\{k'\}}(x) \psi_{\{k\}}(x) d^N x = \prod_j \delta(k_j - k'_j) \ ,$$

where $G(k) = \prod_{i<j} \left(1 + \frac{c^2}{(k_i - k_j)^2} \right)$, and the momenta and positions are ordered increasingly $k_1 < k_2 < \cdots < k_N$ and $x_1 < x_2 < \cdots < x_N$.

Sketch of proof

To prove the previous statement, first we need the integral

$$F(\{k\}) = \int_D \exp\left[-i\sum_j k_j x_j\right] d^N x = \frac{2\pi i^{N-1}\delta\left(\sum_j k_j\right)}{(k_1+i0)(k_1+k_2+i0)\dots(k_1+k_2+\dots+k_{N-1}+i0)} ,$$

which has the properties

$$\sum_{\sigma\in S_N} F(\{k_\sigma\}) = (2\pi)^N \prod_j \delta(k_j) = 2\pi i^{N-1}\delta\Big(\sum_j k_j\Big)\sum_l \left(\frac{k_l+i0}{\prod_j (k_j+i0)}\right) .$$

From that, the proof is just (tedious and non-trivial) rewritings.

Summary

- The CBA gives us the wavefunction and the energy in a very simple and compact manner, but with the price of solving the BAE.
- The wavefunction is simple to write, but it involver a large number of terms if we have too many excitations ($N!$ terms for N excitations).
- We cannot compute scalar products in a direct way except for few excitations, so we have to rely in the symmetry and the properties of the integrals to find a closed form.
- The normalization proposed by Gaudin actually can be generalized to other systems, but that will have to wait.

Introduction to the Algebraic Bethe Ansatz

Introduction

The Algebraic Bethe Ansatz, developed by the Leningrad School, is a complete shift of gears from the Coordinate version. The most drastic differences are that there would be hardly any mention of the Hamiltonian and absolutely no one of the actual wavefunctions of the eigenstates.

Introduction

The Algebraic Bethe Ansatz, developed by the Leningrad School, is a complete shift of gears from the Coordinate version. The most drastic differences are that there would be hardly any mention of the Hamiltonian and absolutely no one of the actual wavefunctions of the eigenstates.

The first happens because the method treats all the conserved charges (even the non-local ones) at the same level, so we will be working all the time with the generating function of the conserved charges.

Regarding the eigenstates, this method provide a straightforward and simple construction which can be proven to be equivalent to the one of the Coordinate Bethe Ansatz. However, it is not easy to write the operators involved in the Algebraic Bethe Ansatz in terms of creation and annihilation operators. (Interestingly, this computation is also related to the computation of scalar products).

The Algebraic Bethe Ansatz (1)

Let us define L copies of a Hilbert space $\mathfrak{H}_i$ and an "auxiliary space" $\mathfrak{A}$. For the moment we are going to take both to be $\mathbb{C}^2$.
The two main objects involved in the Algebraic Bethe Ansatz are the so called "Lax matrix", acting on $\mathfrak{A} \otimes \mathfrak{H}_i$, and "R-matrix", acting on $\mathfrak{A} \otimes \mathfrak{A}$, which fulfil

$$R_{12}(\lambda-\mu)[\mathfrak{L}(\lambda)\otimes 1][1\otimes\mathfrak{L}(\mu)] = [1\otimes\mathfrak{L}(\mu)][\mathfrak{L}(\lambda)\otimes 1]R_{12}(\lambda-\mu) \ ,$$
$$R_{12}(\lambda-\mu)R_{13}(\lambda-\xi)R_{23}(\mu-\xi) = R_{23}(\mu-\xi)R_{13}(\lambda-\xi)R_{12}(\lambda-\mu) \ .$$

(In the fundamental representation we choose $\mathfrak{L} = R$)

From the Lax matrix we define the monodromy matrix

$$T(\lambda) = \begin{pmatrix} A(\lambda) & B(\lambda) \\ C(\lambda) & D(\lambda) \end{pmatrix} = \prod_{j=1}^{L} \mathfrak{L}_j(\lambda-\nu_j) \ .$$

and the transfer matrix $\tau(\lambda) = A(\lambda) + D(\lambda)$. A, B, C and D are operators defined on $\bigotimes_{i=1}^{L} \mathfrak{H}_i$.

The Algebraic Bethe Ansatz (2)

The operators A, B, C and D satisfy a set of commutation relations (inherited from the Lax matrix) codified by the RTT relation

$$R_{12}(\lambda-\mu)[T(\lambda)\otimes 1][1\otimes T(\mu)] = [1\otimes T(\mu)][T(\lambda)\otimes 1]R_{12}(\lambda-\mu) \ ,$$

$$R(\lambda) = \begin{pmatrix} f(\lambda) & 0 & 0 & 0 \\ 0 & 1 & g(\lambda) & 0 \\ 0 & g(\lambda) & 1 & 0 \\ 0 & 0 & 0 & f(\lambda) \end{pmatrix} \ .$$

In particular, we will need the following commutation relations

$$\begin{aligned}
[\tau(\lambda),\tau(\mu)] = 0 \ , \quad & A(\mu)B(\lambda) = f(\mu,\lambda)B(\lambda)A(\mu) + g(\lambda,\mu)B(\mu)A(\lambda) \ , \\
[B(\lambda),B(\mu)] = 0 \ , \quad & D(\mu)B(\lambda) = f(\lambda,\mu)B(\lambda)D(\mu) + g(\mu,\lambda)B(\mu)D(\lambda) \ , \\
[C(\lambda),C(\mu)] = 0 \ , \quad & [C(\mu),B(\lambda)] = g(\mu,\lambda)\,[A(\mu)D(\lambda) - A(\lambda)D(\mu)] \ .
\end{aligned}$$

(Be careful with the definitions: $f_{\text{some people}}(\mu,\lambda) = f_{\text{other people}}(\lambda,\mu)$)

The Algebraic Bethe Ansatz (3)

$[\tau(\lambda), \tau(\mu)] = 0$ is a very important relation, as it means that the operators we obtain by expanding $\tau(\lambda)$ in powers of λ all commute with each other. In fact, for some particular R-matrices it can be proven that they actually form a complete basis of operators. Thus, if we diagonalize τ, we can simultaneously diagonalize all the operators it contains.

The Algebraic Bethe Ansatz (3)

$[\tau(\lambda), \tau(\mu)] = 0$ is a very important relation, as it means that the operators we obtain by expanding $\tau(\lambda)$ in powers of λ all commute with each other. In fact, for some particular R-matrices it can be proven that they actually form a complete basis of operators. Thus, if we diagonalize τ, we can simultaneously diagonalize all the operators it contains.

Assuming that there exists a pseudovacuum $|0\rangle$ that is an eigenvector of $\tau(\mu)$ and is annihilated by C, $C(\mu)|0\rangle = 0 \ \forall \mu$, then the state $\prod_j B(\lambda_j)|0\rangle$ is an eigenvector of $\tau(\lambda)$ with eigenvalue $\Lambda(\mu, \{\lambda\})$, provided that the set $\{\lambda\}$ fulfils the *Bethe equations*

$$\frac{a(\lambda_n)}{d(\lambda_n)} \prod_{n \neq j} \frac{f(\lambda_n - \lambda_j)}{f(\lambda_j - \lambda_n)} = 1 \ ,$$

where

$$A(\lambda)|0\rangle = a(\lambda)|0\rangle \ , \qquad D(\lambda)|0\rangle = d(\lambda)|0\rangle \ .$$

(For a little bit of concreteness)

Most of the time I will be referring to the so called XXX spin chain model or Heisenberg spin chain. The Hamiltonian associated to this model is $H = \sum_{i=i}^{L} \vec{S}_i \cdot \vec{S}_{i+1}$ with periodic boundary conditions. The R matrix associated to it is given by

$$\mathfrak{L}(\lambda) = R(\lambda) = \mathbb{I} + \frac{i}{\lambda}\mathbb{P} \Longrightarrow \begin{cases} f(\lambda) = \frac{\lambda+i}{\lambda} \\ g(\lambda) = \frac{i}{\lambda} \end{cases}$$

(The Lieb-Liniger has the same R matrix as the XXX spin chain, but the $\mathfrak{L}$ matrix is slightly different)

(For a little bit of concreteness)

Most of the time I will be referring to the so called XXX spin chain model or Heisenberg spin chain. The Hamiltonian associated to this model is $H = \sum_{i=i}^{L} \vec{S}_i \cdot \vec{S}_{i+1}$ with periodic boundary conditions. The R matrix associated to it is given by

$$\mathfrak{L}(\lambda) = R(\lambda) = \mathbb{I} + \frac{i}{\lambda}\mathbb{P} \Longrightarrow \begin{cases} f(\lambda) = \frac{\lambda+i}{\lambda} \\ g(\lambda) = \frac{i}{\lambda} \end{cases}$$

(The Lieb-Liniger has the same R matrix as the XXX spin chain, but the $\mathfrak{L}$ matrix is slightly different)

All the results apply also for the XXZ spin chain, $H = \sum_{i=i}^{L} [S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + \Delta S_i^z S_{i+1}^z]$, just by using instead

$$f(\lambda) = \frac{\sinh(\lambda + \eta)}{\sinh(\lambda)} , \qquad g(\lambda) = \frac{\sinh(\eta)}{\sinh(\lambda)} .$$

Freedom in a and d

Although the functions a and d might seem to be completely determined when we choose the Lax matrix to be equal to the R-matrix, this is far from true.

First of all, thank to the inhomogeneities ν_i we introduced in the definition of the monodromy matrix, we will have L free parameters to play with. This means that in the $L \to \infty$ limit we have enough parameters to consider the ration $a(\lambda)/d(\lambda)$ as completely independent of the functions f and g in the R-matrix.

Furthermore, as $T(\lambda)$ and $h(\lambda)T(\lambda)$, with h an arbitrary function, give raise to the same Bethe equations, then not only their ratio but both functions can be set to be any arbitrary function.

Inverse scattering formulas

This formalisms is all well and good, but at the end of the day we want to compute correlation lengths, magnetizations, etc.
In order to do that, we have to go back from the monodromy matrix operators to operators in our Hilbert space. The relation between them is relatively simple [Maillet, Terras, 1999]

$$E_n^{ij} = \left(\prod_{i=1}^{n-1} \tau(\nu_i) \right) tr \left[E_{\mathfrak{A}}^{ij} T(\nu_n) \right] \left(\prod_{i=1}^{n} \tau(\nu_i) \right)^{-1} ,$$

where $E^{ij} = \delta_{ij}$.
This recipe also provide us with a method to find the explicit form of the wave function as

$$\psi(\{x_i\}) = \langle 0| \prod_{\{x_i\}} E_{x_i}^{21} \prod B(u_j) |0\rangle ,$$

although computing it is very difficult task.

Notation

In order to reduce the number of products appearing in the formulas, I'll be using this shorthand notation

$$\vec{\lambda} \to \{\lambda_j\}_{j=1}^N \ , \qquad \vec{\lambda}_i \to \{\lambda_j\}_{j=1}^N - \{\lambda_i\}$$

$$f(\vec{\lambda}) \to \prod_{i=1}^N f(\lambda_i) \ , \qquad f(\vec{\lambda}, \vec{\mu}) \to \prod_{i,j=1}^N f(\lambda_i, \mu_j) \ ,$$

$$f^<(\vec{\lambda}, \vec{\lambda}) \to \prod_{i<j} f(\lambda_i, \lambda_j) \ , \qquad f^>(\vec{\lambda}, \vec{\lambda}) \to \prod_{i>j} f(\lambda_i, \lambda_j) \ .$$

Classification of scalar products

Hermitian conjugation relate the operators as $B^\dagger(\lambda^*) = \pm C(\lambda)$, so computing scalar products reduces to computing matrix elements of the form

$$\langle 0| \prod_{i=1}^{N} C(\lambda_i) \prod_{j=1}^{N} B(\mu_j)|0\rangle = \langle C(\vec{\lambda})B(\vec{\mu})\rangle \ .$$

Classification of scalar products

Hermitian conjugation relate the operators as $B^\dagger(\lambda^*) = \pm C(\lambda)$, so computing scalar products reduces to computing matrix elements of the form

$$\langle 0| \prod_{i=1}^{N} C(\lambda_i) \prod_{j=1}^{N} B(\mu_j)|0\rangle = \langle C(\vec{\lambda}) B(\vec{\mu})\rangle \ .$$

We will distinguish three kinds of scalar products

1. Off-shell-off-shell scalar products, $\langle 0| \prod_i C(\lambda_i) \prod_j B(\mu_j)|0\rangle$, where none of the two sets of rapidities satisfy the BE.
2. On-shell-off-shell scalar products, $\langle 0| \prod_i C(\lambda_i) \prod_j B(\mu_j)|0\rangle$, where only one of the two sets of rapidities satisfies the BE.
3. Norms, computed as $\lim_{\vec{\mu}\to\vec{\lambda}} \langle 0| \prod_i C(\lambda_i) \prod_j B(\mu_j)|0\rangle$, and where the set $\{\lambda\}$ satisfies the BE.

Brute force computation

We can compute scalar products by brute force via the commutation relations, but this is a very inefficient process and scale out of control very fast. Remember that

$$A(\mu)B(\lambda) = f(\mu,\lambda)B(\lambda)A(\mu) + g(\lambda,\mu)B(\mu)A(\lambda) \ ,$$
$$[C(\mu), B(\lambda)] = g(\mu,\lambda)\left[A(\mu)D(\lambda) - A(\lambda)D(\mu)\right] \ ,$$

and similarly for DB. Then, schematically,

$$\begin{aligned}
\langle CB \rangle &= \langle AD - AD \rangle = ad - ad \ , \\
\langle CCBB \rangle &= \langle C(AD - AD + BC)B \rangle = \\
&= \langle 4CBAD - 4CBAD + CBCB \rangle = 20 \text{ terms} \ , \\
\langle CCCBBB \rangle &\approx 800 \text{ terms} \ .
\end{aligned}$$

The amount of terms involved grow out of proportion very fast.

Brute force computation

Let me show how much time a compute takes to do all the commutation relations involved

# excitations	1	2	3	4	5
Time numerical (s)	0	0.005	0.06	11.4	8920.7 ($\approx$ 2.5 h)
Time symbolic (s)	0	0	0.172	33.28	n/a
Time simplify (s)	0	2.59	n/a	n/a	n/a

Brute force computation

Let me show how much time a compute takes to do all the commutation relations involved

# excitations	1	2	3	4	5
Time numerical (s)	0	0.005	0.06	11.4	8920.7 ($\approx$ 2.5 h)
Time symbolic (s)	0	0	0.172	33.28	n/a
Time simplify (s)	0	2.59	n/a	n/a	n/a

This shows that we need a more intelligent approach if we want to get somewhere.

Summary

- The ABA explains why the CBA results were so "simple": there exists a particular basis that diagonalizes the conserved charges and has well defined creation and annihilation operators.
- The ABA is equivalent to the CBA. We did not showed it in depth, but in both cases we ended with a set of "Bethe equations".
- However, we have to pay the price of less straightforward formulas for the wave function. Which also makes the computation of scalar products more involved.
- In particular, we already saw that the brute force computation is very inefficient, so we have to find alternative methods.

Outline

1. Coordinate Bethe ansatz for the Lieb-Liniger model
2. Introduction to the Algebraic Bethe Ansatz
3. Domain wall partition function
4. Norm of Bethe vectors
5. Scalar products of Bethe vectors: Slavnov and Korepin-Izergin
6. How to efficiently find recursion relations?

   [Hutsalyuk, Liashyk, Pakuliak, Ragoucy, Slavnov, 2017]
7. Example 1: massless relativistic AdS_3 [Nieto, Torrielli, 2019]
8. Example 2: flux-deformed massless relativistic AdS_3 [NT, 2019]
9. Example 3: rational $SU(N)$ [HLPRS, 2017]
10. Why do we get determinants? [Belliard, Slavnov, 2019]

Domain wall partition function

Domain wall partition function

Before trying to make any computation regarding scalar products, we are going to talk first about a related quantity called "domain wall partition function" defined as

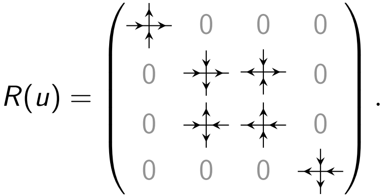
$$Z_N(\vec{u}, \vec{\nu}) = \langle \tilde{0} | \prod_{i=1}^{L} B(u_i) | 0 \rangle$$

where $|\tilde{0}\rangle$ is usually known as the "dual vacuum vector" (all sites occupied instead of empty, or all spins up instead of all spins down).

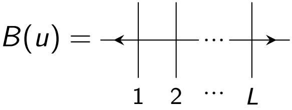
To show the recurrence properties of this object, I'm going to make a small detour and talk about the pictorial representation (as it makes them extremely clear).

Pictorial representation

As our Hilbert and auxiliary spaces are $\mathbb{C}^2$, we can represent them as spins up and down (or left and right, or labels 1 and 2). Then, the matrix elements of the R-matrix can be represented pictorially as



So, for example, we have

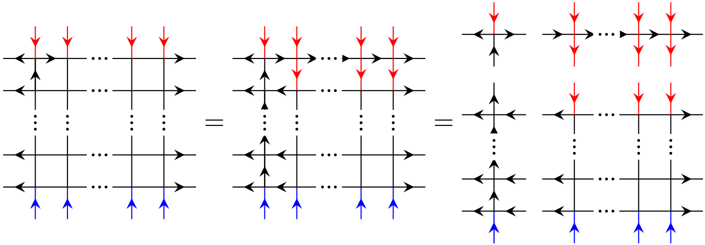


(Image from [Lamers, 2018]).

Recurrence relation

It can be proven [Korepin, 1982] that if we choose a very particular value of the inhomonegenities, the the domain wall partition function for length L can be written in terms of the one for length $L-1$

$$Z_L(\vec{u};\vec{\nu})\big|_{u_L=\nu_1-i} =$$



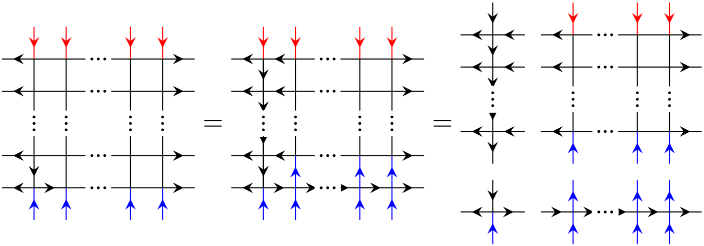
$$= g(-i)\frac{\prod_{i=1}^{L-1} f(u_i-\nu_1)\prod_{j=2}^{L} f(\nu_1-i-\nu_j)}{\prod_{i=1}^{L-1} g(u_i-\nu_1)\prod_{j=2}^{L} g(\nu_1-i-\nu_j)} \times Z_{L-1}(u_1,\dots,u_{L-1};\nu_2,\dots,\nu_L).$$

(Image from [Lamers, 2018]).

A second recurrence relation

Similarly, for a different particular value of the parameter we find a second recurrence relation

$$Z_L(\vec{u};\vec{\nu})\big|_{u_1=\nu_1} =$$



$$= \operatorname{Res} g(0) \prod_{i=2}^{L} f(u_i-\nu_1) f(\nu_1-\nu_i) \times Z_{L-1}(u_2,\dots,u_L;\nu_2,\dots,\nu_L).$$

(Image from [Lamers, 2018]).

Solving the recurrence relation

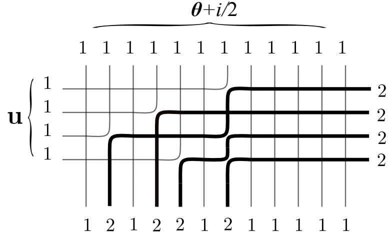
Using the previous recurrence relations and symmetry properties of the DWPF, a closed expression was found [Izergin, 1987]

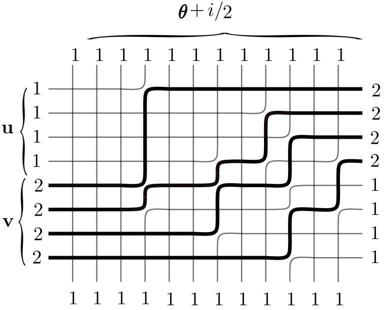
$$Z_L \propto \frac{f(\vec{u},\vec{\nu})f(\vec{\nu},\vec{u})g^{<}(\vec{u},\vec{u})g^{<}(\vec{\nu},\vec{\nu})}{g(\vec{u},\vec{\nu})g(\vec{\nu},\vec{u})} \det\left[\frac{g(u_i,\nu_j)g(\nu_j,u_i)}{f(u_i,\nu_j)f(\nu_j,u_i)}\right] \ .$$

But for us only the recurrence relations will be important, as we will be relating recurrence properties of the scalar product with them.

Pictorial representation of the wave function

The wave function and the scalar products can also be expressed pictorially. Using a different representation (instead of representing the arrows, it marks the directed path created by the down and left arrows), one of the configurations we have to sum over in each case take the form

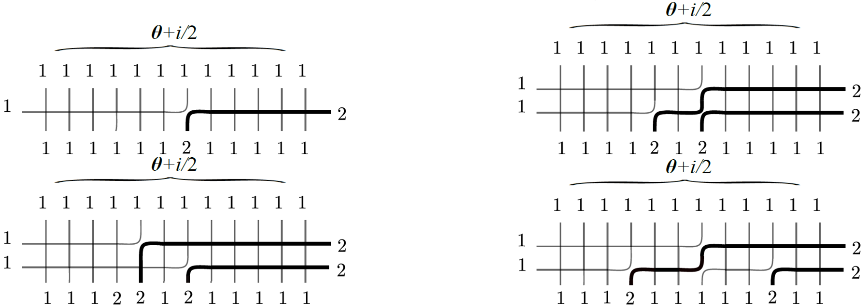




(Image from [Foda, Jiang, Kostov, Serban, 2013]).

Is this better than brute force? No

For the case of one excitation is still just one term, but for the case of two excitations we have three different kinds of diagrams



Furthermore, in the last diagram we still have to sum over the all possible placements of the “up arrow”/bold line.

Norm of Bethe vectors

A theorem [Korepin, 1982]

Given the matrix element

$$F(\vec{\lambda}) = \frac{\langle C(\vec{\lambda})B(\vec{\lambda})\rangle}{\left[\prod_j a(\lambda_j)d(\lambda_j)\right]\left[\prod_{j\neq k} f(\lambda_j,\lambda_k)\right]\langle 0|0\rangle},$$

with $\vec{\lambda}$ fulfilling the BE, and $X_p = i\frac{\partial}{\partial\lambda_p}\log\left(\frac{a(\lambda_p)}{d(\lambda_p)}\right)$, if the following properties are fulfilled

1. F is invariant under swapping of λ's and swapping of X's.
2. F is linear in X_1, i.e. $F = U_1X_1 + V_1$.
3. $U_1 \propto F(\vec{\lambda}_1)^{\text{mod}}$, where *mod* means replacing $a(\lambda) \to a(\lambda)f(\lambda,\lambda_1)$ and $d(\lambda) \to d(\lambda)f(\lambda_1,\lambda)$
4. $F = 0$ if all $X_i = 0$ at fixed $\bar{\lambda}$.
5. $F(\lambda) = X$.

then $F(\vec{\lambda}) = \kappa^N \det\left[i\frac{\partial}{\partial_{\lambda_j}}\log\left(\frac{a(\lambda_k)}{d(\lambda_k)}\prod_{l\neq k}\frac{f(\lambda_k,\lambda_l)}{f(\lambda_l,\lambda_k)}\right)\right]$, where $\kappa = \operatorname*{Res}_{\lambda=0} g(\lambda)$.

(Proven by induction using some analytic properties)

And its proof [Korepin, 1982]

The proof is done by induction. For that we need two things: the base case and the induction step.

We can use as base case the norm of one particle states encoded is property 5, and can be checked by using the commutation relations.

The induction step goes as follows: first we construct $\Delta = F - \kappa^N \det$. If we compute $\frac{\partial \Delta}{\partial X_1}$, we can check that it is zero by combining properties 2 and 3. This is true for any X_i thanks to property 1. So it only remain to prove that $\Delta = 0$ if all $X_i = 0$, which is just property 4.

Rewriting the scalar product

To prove that F has those properties, we need to dig a little bit into the properties of scalar products of N excitations. It is easy to prove that a general scalar product can be expanded as

$$\langle C(\vec{\lambda}^C)B(\vec{\lambda}^B)\rangle = \sum_{\text{part.}} K_N(\vec{\lambda}^A, \vec{\lambda}^D | \vec{\lambda}^C, \vec{\lambda}^B) a(\vec{\lambda}^A) d(\vec{\lambda}^D) \ ,$$

where the sum is over the partitions of the set $\vec{\lambda}^C \cup \vec{\lambda}^B$ into two sets $\vec{\lambda}^A$ and $\vec{\lambda}^D$. The function K is non-zero only if $|\vec{\lambda}^A| = |\vec{\lambda}^D|$.

A very important property is that K only depends on the form of the commutation relations (i.e., on the R-matrix). Thus it is independent of the length of the system, inhomogeneities, and even the functions $a(\lambda)$ and $d(\lambda)$.

Putting independence to a good use (1)

Actually, this independence property of the K_N's can be used to complete fix them. First, the independence of the particular $a(\lambda)$ and $d(\lambda)$ allows us to pick, for example, the inhomogeneous trigonometric 6-vertex model

$$a(\lambda) = \prod_{i=1}^{L} \sinh(\lambda - \nu_i - i\eta) = d(\lambda - 2i\eta) ,$$

and pick inhomogeneities $\nu_j = \lambda_j^C + i\eta$. Thus only one term of the sum over partitions contributes, as $d(\lambda_i^C) = 0$

$$\langle C(\vec{\lambda}^C) B(\vec{\lambda}^B) \rangle = K_N(\vec{\lambda}^C, \vec{\lambda}^B | \vec{\lambda}^C, \vec{\lambda}^B) a(\vec{\lambda}^C) d(\vec{\lambda}^B) .$$

This particular K_N is usually referred in the literature as *highest weight*. We can also define the *lowest weight* as $K_N(\vec{\lambda}^B, \vec{\lambda}^C | \vec{\lambda}^C, \vec{\lambda}^B)$.

Putting independence to a good use(2)

The same idea can be applied using instead $\nu_j = \lambda_j^A + i\eta$, with $\vec{\lambda}^A$ begin a combination of elements of $\vec{\lambda}^B$ and $\vec{\lambda}^C$, in order to pick a particular partition.

Putting independence to a good use(2)

The same idea can be applied using instead $\nu_j = \lambda_j^A + i\eta$, with $\vec{\lambda}^A$ begin a combination of elements of $\vec{\lambda}^B$ and $\vec{\lambda}^C$, in order to pick a particular partition.

But also, we can pick any length of the spin chain we want, so we can pick exactly $L = N$, and relate our scalar product with the so-called domain-wall partition function

$$\langle 0|C(\vec{\lambda}^C)B(\vec{\lambda}^B)|0\rangle = \langle 0|C(\vec{\lambda}^C)|\tilde{0}\rangle\langle\tilde{0}|B(\vec{\lambda}^B)|0\rangle \ .$$

Putting independence to a good use(2)

The same idea can be applied using instead $\nu_j = \lambda_j^A + i\eta$, with $\vec{\lambda}^A$ begin a combination of elements of $\vec{\lambda}^B$ and $\vec{\lambda}^C$, in order to pick a particular partition.

But also, we can pick any length of the spin chain we want, so we can pick exactly $L = N$, and relate our scalar product with the so-called domain-wall partition function

$$\langle 0|C(\vec{\lambda}^C)B(\vec{\lambda}^B)|0\rangle = \langle 0|C(\vec{\lambda}^C)|\tilde{0}\rangle\langle\tilde{0}|B(\vec{\lambda}^B)|0\rangle \ .$$

At this point V. E. Korepin uses the recurrence properties of the DWPF to derive recurrence relations for K, which he later uses to prove that $F \propto F(\vec{\lambda}_1)^{\text{mod}} X_1 + V_1$. I will not comment much about it, as later we will talk about better ways to find recursion relations directly for the scalar products.

Back to the properties

Properties 1 and 5 are obvious from the commutation relations and the functions involved.
Properties 2 and 3 are a consequence of properties of the DWPF (in particular the first recurrence relation).
Property 4 is a little bit more odd and related to what the X's are.

Back to the properties

Properties 1 and 5 are obvious from the commutation relations and the functions involved.
Properties 2 and 3 are a consequence of properties of the DWPF (in particular the first recurrence relation).
Property 4 is a little bit more odd and related to what the X's are.

For a fixed solution of the BE, $\vec{\lambda}$, we can always construct new functions $\hat{a}$ and $\hat{d}$ such that

$$\frac{\hat{a}(\lambda_i)}{\hat{d}(\lambda_i)} = \frac{a(\lambda_i)}{d(\lambda_i)} \quad \forall \lambda_i \in \bar{\lambda} \ .$$

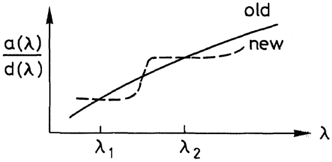
Therefore, $X_p = i \frac{\partial}{\partial \lambda_p} \log \left(\frac{a(\lambda_p)}{d(\lambda_p)} \right)$ can change freely for a fixed $\vec{\lambda}$, and for any given $\hat{a}$ and $\hat{d}$ we can always construct a monodromy matrix $T(\lambda)$.

Deriving property 4 (1)

To prove property 4, pick $S = \frac{1}{4} \min |\lambda_i - \lambda_j|$, so we can define

$$\frac{\hat{a}(\lambda)}{\hat{d}(\lambda)} = \frac{a(\lambda_i)}{d(\lambda_i)} \text{ if } |\lambda - \lambda_i| < S \ .$$

In this variables, $F(\vec{\lambda})$ is the same as the original one but with all X_i set to zero.



Furthermore, with these new functions if $\vec{\lambda}$ is a solution of the Bethe equations, so it is $\{\lambda + y\}$ provided that $|y| < S$ and $f(\lambda, \mu) = f(\lambda - \mu)$.

Deriving property 4 (2)

However, if we want $\langle C(\vec{\lambda})\tau(\mu)B(\{\lambda_i + y\})\rangle$ to be consistent we need that

$$\left[\Lambda(\mu, \vec{\lambda}) - \Lambda(\mu, \{\lambda_i + y\})\right] \langle C(\vec{\lambda})B(\{\lambda_i + y\})\rangle = 0 \ .$$

Using that for the XXX and XXZ R-matrix we have

$$\Lambda(\mu, \vec{\lambda}) = a(\mu)f(\mu, \vec{\lambda}) + d(\mu)f(\vec{\lambda}, \mu) \ ,$$

we can see that the term between brackets cannot cancel, so the only possibility is to have $F(\vec{\lambda}) \propto \langle C(\vec{\lambda})B(\vec{\lambda})\rangle = 0$ when $X_i = 0$.

Summary

- Computing the norm of Bethe vectors reduces at the end to finding some key properties.
- These properties at the end boil down to symmetry arguments and recursion relations.
- We can apply the same symmetry arguments to the computation of scalar products, but the recursion relations are different (either because we are imposing the Bethe equations or because we computed the norm limit).
- So, what we need is an efficient way of computing (and solving) the recurrence relations. But, before that, let me show the expressions for scalar products for completeness.

Scalar products of Bethe vectors: Slavnov and Korepin-Izergin

Slavnov determinant [Slavnov, 1989]

Given a set of variables $\vec{\lambda}^C$ that fulfil the Bethe equations and a set of complex numbers $\vec{\lambda}^B$ (on-shell-off-shell scalar product), the explicit form of the scalar product $\langle C(\vec{\lambda}^C) B(\vec{\lambda}^B) \rangle$ is given by

$$S_N = \langle C(\vec{\lambda}^C) B(\vec{\lambda}_B) \rangle = d(\vec{\lambda}^B) d(\vec{\lambda}^C) G_N \det M_{lk} \ ,$$

$$G_N = g^{>}(\vec{\lambda}^B, \vec{\lambda}^B) g^{<}(\vec{\lambda}^C, \vec{\lambda}^C) \frac{f(\vec{\lambda}^C \vec{\lambda}^B)}{g(\vec{\lambda}^C, \vec{\lambda}^B)} \ ,$$

$$M_{lk} = \frac{g(\lambda_k^C, \lambda_l^B)^2}{f(\lambda_k^C, \lambda_l^B)} - \frac{a(\lambda_l^B)}{d(\lambda_l^B)} \frac{g(\lambda_l^B, \lambda_k^C)^2}{f(\lambda_l^B, \lambda_k^C)} \prod_m \frac{f(\lambda_l^B, \lambda_m^C)}{f(\lambda_m^C, \lambda_l^B)} \ .$$

Slavnov determinant [Slavnov, 1989]

Given a set of variables $\vec{\lambda}^C$ that fulfil the Bethe equations and a set of complex numbers $\vec{\lambda}^B$ (on-shell-off-shell scalar product), the explicit form of the scalar product $\langle C(\vec{\lambda}^C)B(\vec{\lambda}^B)\rangle$ is given by

$$S_N = \langle C(\vec{\lambda}^C)B(\vec{\lambda}_B)\rangle = d(\vec{\lambda}^B)d(\vec{\lambda}^C)G_N \det M_{lk} \ ,$$

$$G_N = g^{>}(\vec{\lambda}^B,\vec{\lambda}^B)g^{<}(\vec{\lambda}^C,\vec{\lambda}^C)\frac{f(\vec{\lambda}^C\vec{\lambda}^B)}{g(\vec{\lambda}^C,\vec{\lambda}^B)} \ ,$$

$$M_{lk} = \frac{g(\lambda_k^C,\lambda_l^B)^2}{f(\lambda_k^C,\lambda_l^B)} - \frac{a(\lambda_l^B)}{d(\lambda_l^B)}\frac{g(\lambda_l^B,\lambda_k^C)^2}{f(\lambda_l^B,\lambda_k^C)}\prod_m \frac{f(\lambda_l^B,\lambda_m^C)}{f(\lambda_m^C,\lambda_l^B)} \ .$$

Interestingly and luckily, the expression is the same for the off-shell-on-shell ($\vec{\lambda}^B$ satisfies the BAE while $\vec{\lambda}^C$ is a set of complex numbers). Naïvely, we would expect that we would have to change the B and C labels, but that's not necessary.

Slavnov determinant (2)

To prove that Slavnov used the following properties

1. S_N is invariant under swapping of λ's and swapping of X's.
2. S_N is linear in X_1, i.e. $F = U_1 X_1 + V_1$.
3. $S_N \to \frac{1}{\lambda_i^X}$ when $\lambda_i^X \to \infty$.
4. Recurrence equation relating S_N and S_{N-1} (obtained in this case from the second recurrence relation for the DWPF).
5. S_1 can be computed from commutation relations.

With all of those properties, the Slavnov formula can be proven by induction.

Slavnov determinant (2)

To prove that Slavnov used the following properties

1. S_N is invariant under swapping of λ's and swapping of X's.
2. S_N is linear in X_1, i.e. $F = U_1 X_1 + V_1$.
3. $S_N \to \frac{1}{\lambda_i^X}$ when $\lambda_i^X \to \infty$.
4. Recurrence equation relating S_N and S_{N-1} (obtained in this case from the second recurrence relation for the DWPF).
5. S_1 can be computed from commutation relations.

With all of those properties, the Slavnov formula can be proven by induction. Because S_N and the Slavnov determinant have the same recurrence relation, and their difference is a bounded function on the complex plain for all its variables, this difference has to be zero.

Simplifying the determinant

The Slavnov determinant can be simplified more if we look at it more closely. First, the matrix M can be written as the derivative of the eigenvalues of the transfer matrix with respect to the rapidities. It is very simple to check that, if we take $\Lambda(u)$, we have

$$\frac{\partial \Lambda_(\lambda_l^B)}{\partial \lambda_k^C} = f(\vec{\lambda}_C \vec{\lambda}_B) d(\lambda_l^B) M_{lk} \ .$$

In addition, we can use the Cauchy matrix theorem (or its hyperbolic generalization) in the factors of g in front, so we can write the scalar product as

$$S_N = \langle C(\vec{\lambda}^C) B(\vec{\lambda}_B) \rangle = d(\vec{\lambda}_C) \frac{\det \frac{\partial \Lambda_(\lambda_l^B)}{\partial \lambda_k^C}}{\det g(\lambda_l^C - \lambda_k^B)} \ .$$

Symmetric representation

One can prove that this scalar product is completely symmetric under the exchange of the total set of variables, not $\vec{\lambda}^C$ and $\vec{\lambda}^B$ separately.
The prove of this statement is pretty simple to understand

$$\langle C(\vec{\lambda}^C)B(\vec{\lambda}_B)\rangle \cong \langle C(\vec{\lambda}^C)C(\vec{\lambda}^B)(\sum_i S_i^+)^{2\#\{\vec{\lambda}^B\}}\rangle \ ,$$

where all the terms that break this equivalence do not contribute to the scalar product. (To prove that we use that $\sum_i S_i^+ = \lim_{u\to\infty} \frac{u}{i} B(u)$)

Symmetric representation

One can prove that this scalar product is completely symmetric under the exchange of the total set of variables, not $\vec{\lambda}^C$ and $\vec{\lambda}^B$ separately.
The prove of this statement is pretty simple to understand

$$\langle C(\vec{\lambda}^C)B(\vec{\lambda}_B)\rangle \cong \langle C(\vec{\lambda}^C)C(\vec{\lambda}^B)(\sum_i S_i^+)^{2\#\{\vec{\lambda}^B\}}\rangle \ ,$$

where all the terms that break this equivalence do not contribute to the scalar product. (To prove that we use that $\sum_i S_i^+ = \lim_{u\to\infty} \frac{u}{i} B(u)$)

There exists a representation that makes this symmetry explicit, although we are not going to prove its equivalence

$$S_N = d(\vec{\lambda}^C)a(\vec{\lambda}^B)\frac{\det\left(w_j^{k-1} - \frac{d(w_j)}{a(w_j)}(w_j+i)^{k-1}\right)}{\det w_j^{k-1}} \ ,$$

where $w = \vec{\lambda}^C \cup \vec{\lambda}^B$.

Korepin-Izergin determinant

The K_N we define in the previous section can be related to the DWPF if we choose the length equal to the number of B operators we have. Thus

$$\langle 0|C(\vec{\lambda}^C)B(\vec{\lambda}^B)|0\rangle = \langle 0|C(\vec{\lambda}^C)|\tilde{0}\rangle\langle\tilde{0}|B(\vec{\lambda}^B)|0\rangle \ .$$

After some non-trivial algebra and use of the properties of K_N, one can prove that

$$S_N \propto \sum_{\text{part.}} (\text{signs and factors}) \det\left[\frac{g(\lambda_j^C, \lambda_k^B)^2}{f(\lambda_j^C, \lambda_k^B)}\right] \det\left[\frac{g(\lambda_j^C, \lambda_k^B)^2}{f(\lambda_j^C, \lambda_k^B)}\right] \ .$$

which is valid also for off-shell rapidities.

Korepin-Izergin determinant

The K_N we define in the previous section can be related to the DWPF if we choose the length equal to the number of B operators we have. Thus

$$\langle 0|C(\vec{\lambda}^C)B(\vec{\lambda}^B)|0\rangle = \langle 0|C(\vec{\lambda}^C)|\tilde{0}\rangle\langle\tilde{0}|B(\vec{\lambda}^B)|0\rangle \ .$$

After some non-trivial algebra and use of the properties of K_N, one can prove that

$$S_N \propto \sum_{\text{part.}} (\text{signs and factors}) \det\left[\frac{g(\lambda_j^C,\lambda_k^B)^2}{f(\lambda_j^C,\lambda_k^B)}\right] \det\left[\frac{g(\lambda_j^C,\lambda_k^B)^2}{f(\lambda_j^C,\lambda_k^B)}\right] \ .$$

which is valid also for off-shell rapidities.

However, in contrast with the two previous cases, this scalar product cannot be written in terms of a single determinant.

Outline

How to efficiently find recursion relations?

[Hutsalyuk, Liashyk, Pakuliak, Ragoucy, Slavnov, 2017]

The composite model

Going back to the definition of the monodromy matrix

$$T(\lambda) = \begin{pmatrix} A(\lambda) & B(\lambda) \\ C(\lambda) & D(\lambda) \end{pmatrix} = \prod_{j=1}^{L} \mathfrak{L}_j(\lambda - \nu_j) \ ,$$

we can think about separating the product over Lax matrices into two separate products

$$T(\lambda) = \prod_{j=1}^{L} \mathfrak{L}_j(\lambda - \nu_j) = \prod_{j=1}^{r} \mathfrak{L}_j(\lambda - \nu_j) \prod_{j=r+1}^{L} \mathfrak{L}_j(\lambda - \nu_j) = T_1(\lambda) T_2(\lambda) \ .$$

(The ordering of T_1 and T_2 depends on if we are using the forward or backward ordering of the Lax matrices, but the results do not depend on that.)

This splitting is usually called *composite model* or *two-site generalized model*.

Breaking the operators

Straightforwardly from the definition can see that the B and C operators become

$$B(\lambda) = A_1(\lambda)B_2(\lambda) + B_1(\lambda)D_2(\lambda) \ ,$$
$$C(\lambda) = C_1(\lambda)A_2(\lambda) + D_1(\lambda)C_2(\lambda) \ .$$

Breaking the operators

Straightforwardly from the definition can see that the B and C operators become

$$B(\lambda) = A_1(\lambda)B_2(\lambda) + B_1(\lambda)D_2(\lambda) \; ,$$
$$C(\lambda) = C_1(\lambda)A_2(\lambda) + D_1(\lambda)C_2(\lambda) \; .$$

Let us see what happens when we have two creation operators applied to the pseudo vacuum

$$\begin{aligned} B(\lambda)B(\mu)|0\rangle &= [A_1(\lambda)B_2(\lambda) + B_1(\lambda)D_2(\lambda)] \\ &\times [A_1(\mu)B_2(\mu) + B_1(\mu)D_2(\mu)]\,|0\rangle \\ &= [A_1(\lambda)A_1(\mu)B_2(\lambda)B_2(\mu) + A_1(\lambda)B_1(\mu)B_2(\lambda)D_2(\mu) \\ &+B_1(\lambda)A_1(\mu)D_2(\lambda)B_2(\mu) + B_1(\lambda)B_1(\mu)D_2(\lambda)D_2(\mu)]\,|0\rangle \; . \end{aligned}$$

More on breaking

Now, we would like to move the A and D operators to the right. Operators from with different subindices commute, but we still have the relation $A_i(\mu)B_i(\lambda) = f(\mu,\lambda)B_i(\lambda)A_i(\mu) + g(\lambda,\mu)B_i(\mu)A_i(\lambda)$, and similarly for D. After some work we get

$$B(\lambda)B(\mu)|0\rangle = [B_2(\lambda)B_2(\mu)A_1(\lambda)A_1(\mu) + f(\lambda,\mu)B_1(\mu)B_2(\lambda)A_1(\lambda)D_2(\mu)$$
$$+f(\mu,\lambda)B_1(\lambda)B_2(\mu)A_1(\mu)D_2(\lambda) + B_1(\lambda)B_1(\mu)D_2(\lambda)D_2(\mu)]\,|0\rangle \ .$$

Notice that only f's appear and all *unwanted* terms cancel.

More on breaking

Now, we would like to move the A and D operators to the right. Operators from with different subindices commute, but we still have the relation $A_i(\mu)B_i(\lambda) = f(\mu,\lambda)B_i(\lambda)A_i(\mu) + g(\lambda,\mu)B_i(\mu)A_i(\lambda)$, and similarly for D. After some work we get

$$B(\lambda)B(\mu)|0\rangle = [B_2(\lambda)B_2(\mu)A_1(\lambda)A_1(\mu) + f(\lambda,\mu)B_1(\mu)B_2(\lambda)A_1(\lambda)D_2(\mu)$$
$$+f(\mu,\lambda)B_1(\lambda)B_2(\mu)A_1(\mu)D_2(\lambda) + B_1(\lambda)B_1(\mu)D_2(\lambda)D_2(\mu)]\,|0\rangle\ .$$

Notice that only f's appear and all *unwanted* terms cancel.
This can be generalized to any number of creation operators, giving us

$$B(\vec{\lambda})|0\rangle = \sum_{\{\beta\}\cup\{\bar{\beta}\}=\vec{\lambda}} w_B(\beta,\bar{\beta})B_1(\beta)B_2(\bar{\beta})|0\rangle$$

where $w_B(\beta,\bar{\beta}) = f(\bar{\beta},\beta)a_1(\bar{\beta})d_2(\beta)$.

Breaking the scalar product

A similar expression can be constructed for the annihilation operators

$$\langle 0| C(\vec{\lambda}) = \sum_{\{\gamma\}\cup\{\bar{\gamma}\}=\vec{\lambda}} \langle 0| C_1(\gamma) C_2(\bar{\gamma}) w_C(\gamma, \bar{\gamma})$$

where $w_C(\gamma, \bar{\gamma}) = f(\gamma, \bar{\gamma}) d_1(\bar{\gamma}) a_2(\gamma)$.

Breaking the scalar product

A similar expression can be constructed for the annihilation operators

$$\langle 0|C(\vec{\lambda}) = \sum_{\{\gamma\}\cup\{\bar{\gamma}\}=\vec{\lambda}} \langle 0|C_1(\gamma)C_2(\bar{\gamma})w_C(\gamma,\bar{\gamma})$$

where $w_C(\gamma,\bar{\gamma}) = f(\gamma,\bar{\gamma})d_1(\bar{\gamma})a_2(\gamma)$.

And, substituting into the definition of a general scalar product we find

$$S_N(\vec{\lambda}^C,\vec{\lambda}^B) = \sum_{\text{part.}} w_C(\gamma,\bar{\gamma})w_B(\beta,\bar{\beta})S_n(\{\gamma\},\{\beta\})S_{N-n}(\{\bar{\gamma}\},\{\bar{\beta}\})$$

Recursion of the K_N

Now we can apply a modification of the trick from Korepin to this recursion relation of the scalar product. By writing the scalar products as

$$\langle C(\vec{\lambda}^C)B(\vec{\lambda}^B)\rangle = \sum_{\text{part.}} K_N(\vec{\lambda}^A, \vec{\lambda}^D | \vec{\lambda}^C, \vec{\lambda}^B) a(\vec{\lambda}^A) d(\vec{\lambda}^D) \ .$$

Now, as the K's are independent from the choice of a and d, we will choose them such that $a_1(\lambda_i^C) = a_2(\lambda_j^B) = 0$ for some particular choices $\lambda_i^C \in \hat{\gamma} \subset \vec{\lambda}^C$ and $\lambda_j^B \in \hat{\beta} \subset \vec{\lambda}^B$, such that $|\hat{\gamma}| + |\hat{\beta}| = N$. With this choice we can see that the previous formula reduces to

$$K_N\left(\overline{\hat{\beta} \cup \hat{\gamma}}, \hat{\beta} \cup \hat{\gamma} | \vec{\lambda}^C, \vec{\lambda}^B\right) = \left[w_C(\hat{\gamma}, \vec{\lambda}^C - \hat{\gamma}) w_B(\vec{\lambda}^B - \hat{\beta}, \hat{\beta}) \right]_{\text{without a and d}} LW_n(\hat{\gamma}, \vec{\lambda}^B - \hat{\beta}) HW_{N-n}(\vec{\lambda}^C - \hat{\gamma}, \hat{\beta}) \ .$$

So **any** K_N can be obtained if we know the highest (and lowest) weight.

Now, what do we do with the highest weight?

The answer is: more recurrence relations.

Now, what do we do with the highest weight?

The answer is: more recurrence relations.

For that we need to compute the commutation relation between a single C operator and a stack of B operators

$$C(\mu)\prod_{i=1}^{N} B(\lambda_i)|0\rangle = \Big(B(\lambda_N)C(\mu)+g(\mu,\lambda_N)\left[A(\mu)D(\lambda_N)-A(\lambda_N)D(\mu)\right]\Big)\prod_{i=1}^{N-1} B(\lambda_i)|0\rangle \, .$$

However, instead of computing all the terms that can contribute to a highest weight (the weight associated to all $\vec{\lambda}^B$ appearing in d functions) in the RHS, we can just compute those associated to $\prod_{i=1}^{N-1} B(\lambda_i)|0\rangle$ and get the remaining ones from the symmetry between the original B's.

Now, what do we do with the highest weight? (2)

$$C(\mu)\prod_{i=1}^{N}B(\lambda_i)|0\rangle = \Big(B(\lambda_N)C(\mu)+g(\mu,\lambda_N)\left[A(\mu)D(\lambda_N)-A(\lambda_N)D(\mu)\right]\Big)\prod_{i=1}^{N-1}B(\lambda_i)|0\rangle\,.$$

With that in mind, we can completely ignore the first term. In addition, the third term cannot contribute, as we cannot move the μ out of the D operator without using another λ_i from the stack of B's. Finally, we only have to commute the A and D in a *wanted* way through the stack of B's in the second term.

Now, what do we do with the highest weight? (2)

$$C(\mu)\prod_{i=1}^{N}B(\lambda_i)|0\rangle = \Big(B(\lambda_N)C(\mu)+g(\mu,\lambda_N)\left[A(\mu)D(\lambda_N)-A(\lambda_N)D(\mu)\right]\Big)\prod_{i=1}^{N-1}B(\lambda_i)|0\rangle\ .$$

With that in mind, we can completely ignore the first term. In addition, the third term cannot contribute, as we cannot move the μ out of the D operator without using another λ_i from the stack of B's. Finally, we only have to commute the A and D in a *wanted* way through the stack of B's in the second term.

$$C(\mu)\prod_{i=1}^{N}B(\lambda_i)|0\rangle \propto g(\mu,\lambda_N)\left(\prod_{i=1}^{N-1}f(\mu,\lambda_i)f(\lambda_i,\lambda_N)B(\lambda_i)\right)A(\mu)D(\lambda_N)|0\rangle\ .$$

Now, what do we do with the highest weight? (3)

This gives us a recurrence relation between a highest weight with N excitations and a highest weight with $N-1$ excitations

$$HW_N(\vec{\lambda}^C, \vec{\lambda}^B) = \sum_{j=1}^{N} g(\lambda_N^C, \lambda_j^B) \prod_{i \neq j} f(\lambda_N^C, \lambda_i^B) f(\lambda_i^B, \lambda_j^B) HW_{N-1}(\vec{\lambda}_N^C, \vec{\lambda}_j^B) \ .$$

Now, what do we do with the highest weight? (3)

This gives us a recurrence relation between a highest weight with N excitations and a highest weight with $N-1$ excitations

$$HW_N(\vec{\lambda}^C, \vec{\lambda}^B) = \sum_{j=1}^{N} g(\lambda_N^C, \lambda_j^B) \prod_{i \neq j} f(\lambda_N^C, \lambda_i^B) f(\lambda_i^B, \lambda_j^B) HW_{N-1}(\vec{\lambda}_N^C, \vec{\lambda}_j^B) \; .$$

Notice that, although not explicitly symmetric under exchange of the λ^C arguments, it has to be so.

A similar construction can be done for the lowest weights, but they can also be obtained from symmetry arguments.

"Sums all the way down"

Thus, the four steps to compute a general (off-shell-off-shell) scalar product are

1. Construct the recurrence relation for the highest weight from the ABA algebra.
2. Solve the recurrence relation for the highest weight.
3. Use the composite model to construct the K_N from the highest weights.
4. Construct the scalar product from weighted sums of K_N

Now, let me illustrate better how it works with a couple of non-trivial examples: a weird R-matrix and the $SU(N)$ spin chain.

Example 1: massless relativistic AdS_3 [Nieto, Torrielli, 2019]

R-matrix

The first example we are going to be looking at is the R-matrix for the massless relativistic limit of excitations in AdS_3 with pure R-R flux, given by

$$R(\theta) = E_{11} \otimes E_{11} - E_{22} \otimes E_{22}$$
$$- \tanh \frac{\theta}{2} \left(E_{11} \otimes E_{22} - E_{22} \otimes E_{11} \right) - \text{sech} \frac{\theta}{2} \left(E_{12} \otimes E_{21} - E_{21} \otimes E_{12} \right) ,$$

[Fontanella, Ohlsson Sax, Stefański, Torrielli, 2019] where the index 2 is a fermionic index, so we have to be careful as some extra minus signs would appear in our construction. This particular R-matrix is very reminiscent of (but not the same as) a trigonometric $su(1|1)$ spin chain.

Commutation relations

For the case of the first R-matrix, the commutation relations we are interested in are

$$[B(\mu), B(\lambda)] = 0 \ , \qquad A(\mu)B(\lambda) = -\coth\frac{\mu-\lambda}{2} B(\lambda)A(\mu) + \operatorname{csch}\frac{\mu-\lambda}{2} B(\mu)A(\lambda) \ ,$$

$$[C(\mu), C(\lambda)] = 0 \ , \qquad D(\mu)B(\lambda) = -\coth\frac{\mu-\lambda}{2} B(\lambda)D(\mu) + \operatorname{csch}\frac{\mu-\lambda}{2} B(\mu)D(\lambda) \ ,$$

$$[C(\mu), B(\lambda)] = \operatorname{csch}\frac{\mu-\lambda}{2} \left[D(\lambda)A(\mu) - D(\mu)A(\lambda)\right] \ ,$$

with $\tau(\lambda) = A(\lambda) - D(\lambda)$.

Commutation relations

For the case of the first R-matrix, the commutation relations we are interested in are

$$[B(\mu), B(\lambda)] = 0 \, , \qquad A(\mu)B(\lambda) = -\coth\frac{\mu-\lambda}{2}B(\lambda)A(\mu) + \operatorname{csch}\frac{\mu-\lambda}{2}B(\mu)A(\lambda) \, ,$$

$$[C(\mu), C(\lambda)] = 0 \, , \qquad D(\mu)B(\lambda) = -\coth\frac{\mu-\lambda}{2}B(\lambda)D(\mu) + \operatorname{csch}\frac{\mu-\lambda}{2}B(\mu)D(\lambda) \, ,$$

$$[C(\mu), B(\lambda)] = \operatorname{csch}\frac{\mu-\lambda}{2}\left[D(\lambda)A(\mu) - D(\mu)A(\lambda)\right] \, ,$$

with $\tau(\lambda) = A(\lambda) - D(\lambda)$.

The behaviour under hermitian conjugation of the B operators is given by

$$B^{\dagger}(u|\{\nu_j\}) = -[B^{st_{\text{phys}}}(u)]^* = \frac{-i\, C(u^*|\{\nu_j^* - i\pi\})}{\prod_{j=1}^{L}\tanh\frac{u^*-\nu_j^*+i\pi}{2}} \, .$$

Thus, as long as we choose the inhomogeneities such that $\text{Im}[\nu_j] = -i\pi/2$, we end up in the same spin chain under hermitian conjugation.

Computing a the highest weight

The recurrence relation for the highest weight of the first R-matrix is

$$HW_N(\vec{u}|\vec{v}) = \sum_j \operatorname{csch}\left(\frac{u_N - v_j}{2}\right)$$

$$\prod_{i\neq j} \coth\left(\frac{u_N - v_i}{2}\right) \coth\left(\frac{v_i - v_i}{2}\right) HW_{N-1}(\bar{u}_N|\bar{v}_j) \ .$$

Computing a the highest weight

The recurrence relation for the highest weight of the first R-matrix is

$$HW_N(\vec{u}|\vec{v}) = \sum_j \operatorname{csch}\left(\frac{u_N - v_j}{2}\right) \prod_{i\neq j} \coth\left(\frac{u_N - v_i}{2}\right) \coth\left(\frac{v_i - v_i}{2}\right) HW_{N-1}(\bar{u}_N|\bar{v}_j) \ .$$

After rewriting $HW(\vec{u}|\vec{v}) = \coth^{<}\left(\frac{\vec{v}-\vec{v}}{2}\right) \coth\left(\frac{\vec{u}-\vec{v}}{2}\right) \overline{HW}(\vec{u}|\vec{v})$, this recursion relation is nothing more than the Laplace expansion of a determinant

$$\overline{HW}_N(\vec{u}|\vec{v}) = \sum_j (-1)^{j-1} \operatorname{sech}\left(\frac{u_N - v_j}{2}\right) \prod_{k\neq N} \tanh\left(\frac{u_k - v_j}{2}\right) \overline{HW}_{N-1}(\bar{u}_N|\bar{v}_j) \ ,$$

Simplifying it

We get the determinant

$$\overline{HW}_N(\vec{u}|\vec{v}) = \det\left[\operatorname{sech}\left(\frac{u_{N+1-i} - v_j}{2}\right)\prod_{k=1}^{N-i}\tanh\left(\frac{u_k - v_j}{2}\right)\right] .$$

but it can be simplified even more if we use that

$$\operatorname{sech}\left(\frac{u_i - v_j}{2}\right)\tanh\left(\frac{u_k - v_j}{2}\right) + \operatorname{csch}\left(\frac{u_k - u_i}{2}\right)\operatorname{sech}\left(\frac{u_k - v_j}{2}\right) = \operatorname{sech}\left(\frac{u_i - v_j}{2}\right)\coth\left(\frac{u_k - u_i}{2}\right) ,$$

which give us

$$\overline{HW}_N(\vec{u}|\vec{v}) = \coth^{<}\left(\frac{\vec{u} - \vec{u}}{2}\right)\det\left[\operatorname{sech}\left(\frac{u_{N+1-i} - v_j}{2}\right)\right] .$$

Even more simplifications

But we are not done with the simplifications. The matrix inside the determinant looks like a hyperbolic version of a Cauchy matrix ($M_{ij} = \frac{1}{x_i - y_j}$), and indeed satisfies a similar theorem

$$\det \left[\operatorname{sech} \left(\frac{u_{N+1-i} - v_j}{2} \right) \right] = (-1)^{\frac{N(N-1)}{2}} \sinh^{<} \left(\frac{\vec{u} - \vec{u}}{2} \right) \sinh^{<} \left(\frac{\vec{v} - \vec{v}}{2} \right) \operatorname{sech} \left(\frac{\vec{u} - \vec{v}}{2} \right) \ ,$$

Even more simplifications

But we are not done with the simplifications. The matrix inside the determinant looks like a hyperbolic version of a Cauchy matrix ($M_{ij} = \frac{1}{x_i - y_j}$), and indeed satisfies a similar theorem

$$\det \left[\operatorname{sech} \left(\frac{u_{N+1-i} - v_j}{2} \right) \right] = (-1)^{\frac{N(N-1)}{2}} \sinh^{<} \left(\frac{\vec{u} - \vec{u}}{2} \right) \sinh^{<} \left(\frac{\vec{v} - \vec{v}}{2} \right) \operatorname{sech} \left(\frac{\vec{u} - \vec{v}}{2} \right) ,$$

thus

$$HW_{|u|}(\vec{u}|\vec{v}) = (-1)^{\frac{|u|(|u|-1)}{2}} \cosh^{<} \left(\frac{\vec{u} - \vec{u}}{2} \right) \cosh^{<} \left(\frac{\vec{v} - \vec{v}}{2} \right) \operatorname{csch} \left(\frac{\vec{u} - \vec{v}}{2} \right) .$$

For this R-matrix, the weights from breaking the B and C operators into two are

$$w_B(\beta,\bar{\beta}) = \coth\left(\frac{\bar{\beta}-\beta}{2}\right) a_1(\bar{\beta}) d_2(\beta) \ , \quad w_C(\beta,\bar{\beta}) = \coth\left(\frac{\beta-\bar{\beta}}{2}\right) d_1(\bar{\beta}) a_2(\beta) \ .$$

(Be careful when computing them, as the tensor products here are graded so, e.g., $B_2 C_1 = -C_1 B_2$).

Putting everything together, we have

$$\begin{aligned} &K_N(\gamma\cup\tilde{\beta},\beta\cup\tilde{\gamma}|\gamma\cup\tilde{\gamma},\beta\cup\tilde{\beta}) \\ &= (-1)^{\frac{|\gamma|(|\gamma|-1)}{2}+\frac{(N-|\gamma|)(N-|\gamma|-1)}{2}+|\gamma|(N-|\gamma|)} \cosh^{<}\left(\frac{\vec{u}-\vec{u}}{2}\right) \cosh^{<}\left(\frac{\vec{v}-\vec{v}}{2}\right) \\ &\times \operatorname{csch}\left(\frac{\gamma-\tilde{\gamma}}{2}\right) \operatorname{csch}\left(\frac{\tilde{\beta}-\beta}{2}\right) \operatorname{csch}\left(\frac{\beta-\gamma}{2}\right) \operatorname{csch}\left(\frac{\tilde{\gamma}-\tilde{\beta}}{2}\right) \ . \end{aligned}$$

Final expression

After applying again the hyperbolic version of the Cauchy matrix theorem, we surprisingly get just

$$\langle 0| \prod_{j=1}^{N} C(u_j) \prod_{j=1}^{N} B(v_j) |0\rangle = \coth^{<}\left(\frac{\vec{v}-\vec{v}}{2}\right) \coth^{<}\left(\frac{\vec{u}-\vec{u}}{2}\right) \det\left[\operatorname{csch}\left(\frac{u_i - v_j}{2}\right) [a(u_i)d(v_j) - d(u_i)a(v_j)]\right] .$$

Notice that we have not use the Bethe Equations at any point, so this is an off-shell-off-shell scalar product.

Example 2: flux-deformed massless relativistic AdS_3 [NT, 2019]

The other R-matrix

The same procedure can be done for the case of massless relativistic limit of excitations in AdS_3 with R-R and NS-NS flux

$$R_{LL}(\theta) = E_{11} \otimes E_{11} + \frac{e^{\theta} - e^{\frac{2\pi i}{k}}}{e^{\frac{2\pi i}{k}+\theta} - 1} E_{22} \otimes E_{22}$$
$$+ \frac{e^{\frac{\pi i}{k}}(e^{\theta} - 1)}{e^{\frac{2\pi i}{k}+\theta} - 1}(E_{11} \otimes E_{22} + E_{22} \otimes E_{11}) + \frac{e^{\frac{\theta}{2}}(e^{\frac{2\pi i}{k}} - 1)}{e^{\frac{2\pi i}{k}+\theta} - 1}(E_{21} \otimes E_{12} - E_{12} \otimes E_{21}) ,$$

although it requires a little bit of care. This is because, for example, the B do not commute any more. They fulfil instead

$$B(u_1)B(u_2) + \frac{e^{u_1 - u_2} - e^{\frac{2\pi i}{k}}}{e^{\frac{2\pi i}{k}+u_1-u_2} - 1} B(u_2)B(u_1) = 0 .$$

But more importantly, the definition of $B^{\dagger}$ is anything but trivial!

Conjugation

Let me write R_{LL} as

$$R_{LL}(\theta) = E_{11} \otimes E_{11} + c_{LL}(\theta) E_{22} \otimes E_{22} \\ + b_{LL}(\theta)(E_{11} \otimes E_{22} + E_{22} \otimes E_{11}) + a_{LL}(\theta)(E_{21} \otimes E_{12} - E_{12} \otimes E_{21}) \ ,$$

and introduce another R-matrix

$$R_{RL}(\theta) = a_{RL}(\theta) E_{11} \otimes E_{11} + E_{11} \otimes E_{22} \\ + b_{RL}(\theta) \left[E_{21} \otimes E_{21} - E_{12} \otimes E_{12} \right] + E_{22} \otimes E_{11} + c_{RL}(\theta) E_{22} \otimes E_{22} \ ,$$

with

$$a_{RL}(\theta) = -i \frac{\sin\left(\frac{\pi}{k} + \frac{i\theta}{2}\right)}{\sinh\frac{\theta}{2}} \ , \quad b_{RL}(\theta) = \frac{\sin\frac{\pi}{k}}{\sinh\frac{\theta}{2}} \ , \quad c_{RL}(\theta) = \cos\frac{\pi}{k} + i \coth\frac{\theta}{2} \sin\frac{\pi}{k} \ .$$

which are related via supertransposition

$$R_{LL}(\theta)\, \sigma_0^{-1} R_{RL}^{st_0}(\theta)\, \sigma_0 = a_{RL}(\theta) \mathbb{1} \otimes \mathbb{1} \ , \qquad \sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} \otimes \mathbb{1} \ .$$

Relating monodromy matrices

From these R-matrices, we can define two different monodromy matrices with the same physical space

$$T_{LL}(\lambda) = \prod_i R_{LL;0,i}(\lambda - \nu_i) \ , \qquad T_{RL}(\lambda) = \prod_i \frac{R_{RL;0,i}(\lambda - \nu_i)}{a_{RL}(\lambda - \nu_i)} \ ,$$

which are related then as

$$T_{LL}^{st_{\text{phys}}}(u|\vec{\nu}) = \sigma_0^{-1} T_{RL}(-u|-\vec{\nu})\sigma_0 \ .$$

If we focus on the entry for the B operator, we get that

$$B^\dagger(u) = -\left[B^{st_{\text{phys}}}\left(u|\vec{\theta}\right)\right]^* = i\,\left[\tilde{B}\left(-u|-\vec{\theta}\right)\right]^* \ .$$

Eigenfunctions and splitting

To compute the recursion relation in this case we have to go around the non-commutativity of the B operators. This can be done by defining the eigenstates instead as

$$|u_1, \dots, u_M\rangle = \frac{B(u_1|\vec{\theta}) \dots B(u_M|\vec{\theta})|0\rangle}{\prod_{i<j} d_{LL}(u_i - u_j)} \ ,$$

with $d_{LL}(\theta) = e^{-\theta/2}\left(e^{2\pi i/k} - e^{\theta}\right)$, crafted so $d_{LL}(\theta) = -c_{LL}(\theta) d_{LL}(-\theta)$, making this eigenstate symmetric under the exchange of rapidities.

Eigenfunctions and splitting

To compute the recursion relation in this case we have to go around the non-commutativity of the B operators. This can be done by defining the eigenstates instead as

$$|u_1, ..., u_M\rangle = \frac{B(u_1|\vec{\theta})...B(u_M|\vec{\theta})|0\rangle}{\prod_{i<j} d_{LL}(u_i - u_j)} ,$$

with $d_{LL}(\theta) = e^{-\theta/2}\left(e^{2\pi i/k} - e^{\theta}\right)$, crafted so $d_{LL}(\theta) = -c_{LL}(\theta)d_{LL}(-\theta)$, making this eigenstate symmetric under the exchange of rapidities. With that redefinition, we can compute the weights from splitting the B operators (keeping into account the d_{LL} factors in the definition), which in this case amount to

$$w_B(\beta, \bar{\beta}) = \frac{Y(\beta, \bar{\beta})}{d_{LL}(\beta, \bar{\beta})} , \qquad w_B(\gamma, \bar{\gamma}) = \frac{Y(\bar{\gamma}, \gamma)}{d_{LL}(\bar{\gamma}, \gamma)} ,$$

with $Y(\theta) = \frac{c_{LL}(\theta)}{b_{LL}(\theta)}$.

Highest weight (1)

The recurrence relation for the highest weight (defined without the d_{LL} factors) is given by

$$HW_N(\vec{u}|\vec{v}) = \sum_{j=1}^{N} \frac{-b_{RL}(u_M - v_j)}{a_{RL}(u_M - v_j)} \prod_{k \neq j} \frac{Y(v_j - v_k)}{a_{RL}(u_m - v_k)} \prod_{l>j} \left[-c(v_k - v_j)\right] HW_{N-1}(\bar{u}_N|\bar{v}_j) \, .$$

Again, we can get a Laplace expansion of a determinant if we take out a factor $(-1)^M \frac{a_{RL}(\vec{u}-\vec{v})}{Y^{>}(\vec{v}-\vec{v})}$, and use it reconstruct the following intermediate expression

$$HW_N(\vec{u}|\vec{v}) = (-1)^M \frac{a_{RL}(\vec{u} - \vec{v})}{Y^{>}(\vec{v} - \vec{v})} \det \left[b_{RL}(u_i - v_j) \prod_{l=1}^{i-1} a_{RL}(u_k - v_j) \right] \, .$$

Highest weight (2)

This expression can be simplified if we make use of the relations

$$b_{RL}(u_i - v_j)a_{RL}(u_k - v_j) + ib_{RL}(u_i - u_k)b_{RL}(u_k - v_j) = a_{RL}(u_k - u_i)b_{RL}(u_i - v_j) \ ,$$

$$\frac{1}{b_{RL}(u_1, v_j)b_{RL}(u_i, v_1)} - \frac{1}{b_{RL}(u_1, v_1)b_{RL}(u_i, v_j)} = \frac{-1}{b_{RL}(u_1 - u_i)b_{RL}(v_1 - v_j)} \ ,$$

and the relation $a_{RL}b_{LL} = c_{LL}$, so

$$HW_{|u|}(\vec{u}|\vec{v}) = (-1)^{|u|}LW_{|u|}(\vec{u}|\vec{v}) = (-1)^{|u|}\frac{b_{RL}(\vec{u} - \vec{v})a^{<}_{RL}(\vec{u} - \vec{u})a^{>}_{RL}(\vec{v} - \vec{v})}{a_{RL}(\vec{u} - \vec{v})b^{<}_{RL}(\vec{u} - \vec{u})b^{>}_{RL}(\vec{v} - \vec{v})} \ .$$

Scalar product

If we define the scalar product as

$$S(\vec{u}|\vec{v}) = \frac{\langle 0|\tilde{B}(u_1|\vec{\theta}) \dots \tilde{B}(u_M|\vec{\theta}) B(v_M|\vec{\theta}) \dots B(v_1|\vec{\theta})|0\rangle}{d_{LL}^{<}(\vec{u}-\vec{u}) d_{LL}^{>}(\vec{v}-\vec{v})} ,$$

we have

$$\frac{S(\vec{u}|\vec{v})}{\Delta(\vec{u})} = \sum_{\text{part.}} (-1)^{|\gamma|(M-|\gamma|)} \frac{Y(\bar{\gamma}-\gamma) Y(\bar{\beta}-\beta)}{d_{LL}(\bar{\gamma}-\gamma) d_{LL}(\bar{\beta}-\beta)} \frac{LW(\gamma|\beta) HW(\bar{\gamma}|\bar{\beta}) \Delta^{-1}(\gamma) d(\bar{\beta})}{d_{LL}^{<}(\gamma-\gamma) d_{LL}^{<}(\bar{\gamma}-\bar{\gamma}) d_{LL}^{>}(\beta-\beta) d_{LL}^{>}(\bar{\beta}-\bar{\beta})} ,$$

where Δ is the equivalent to $d(\lambda)$ associated to T_{RL}.

Cantilenam eandem canis

After using some relations between the functions a_{RL} and b_{RL}, we get that

$$S(\vec{u}|\vec{v}) = \frac{Y^{<}(\vec{u}-\vec{u})Y^{>}(\vec{v}-\vec{v})}{d_{LL}^{<}(\vec{u}-\vec{u})d_{LL}^{>}(\vec{v}-\vec{v})} \det \left[\frac{b_{RL}(u-v)}{a_{RL}(u-v)} \left[a(v)\alpha(u) - d(v)\Delta(u) \right] \right] .$$

Cantilenam eandem canis

After using some relations between the functions a_{RL} and b_{RL}, we get that

$$S(\vec{u}|\vec{v}) = \frac{Y^{<}(\vec{u}-\vec{u})Y^{>}(\vec{v}-\vec{v})}{d_{LL}^{<}(\vec{u}-\vec{u})d_{LL}^{>}(\vec{v}-\vec{v})} \det \left[\frac{b_{RL}(u-v)}{a_{RL}(u-v)} \left[a(v)\alpha(u) - d(v)\Delta(u) \right] \right] .$$

Notice that we can get rid of the d_{LL}'s in the denominator and write

$$\langle 0|\tilde{B}(u_1|\vec{\theta}) \dots \tilde{B}(u_M|\vec{\theta}) B(v_M|\vec{\theta}) \dots B(v_1|\vec{\theta})|0\rangle = Y^{<}(\vec{u}-\vec{u})Y^{>}(\vec{v}-\vec{v}) \det \left[\frac{b_{RL}(u-v)}{a_{RL}(u-v)} \left[a(v)\alpha(u) - d(v)\Delta(u) \right] \right] .$$

Example 3: rational $SU(N)$ [HLPRS, 2017]

The R-matrix

The final example we are going to look at is the XXX Heisenberg spin chain with $SU(N)$ symmetry. This is a straightforward generalization of the usual Heisenberg spin chain, so direct that even the R-matrix is the same when written in a very particular form

$$R_{12}^{N}(u) = \mathbb{I}_N + \frac{i}{u}\mathbb{P}_{12}^{N}$$

where $\mathbb{P}_{12}$ permutes spaces 1 and 2. The difference is that $\mathfrak{A} = \mathfrak{H}_i = \mathbb{C}^N$ in this case.

Bethe vectors and Nested Bethe Ansatz

The Algebraic Bethe Ansatz is a very powerful tool, but it only works for systems whose symmetry is of rank 1 ($SU(2)$, $SL(2)$ and $SU(1|1)$). For higher ranks we need to use what is called "nesting procedure". The basic idea is to apply some modification of the ABA to each level of the filtration defined by nested $SU(n)$ groups.

The method [Kulish, Reshetikhin, 1983] goes as follows, we define

$$T^N(\lambda) = \begin{pmatrix} A_{11}(\lambda) & B_{1i}(\lambda) \\ C_{i1}(\lambda) & D_{ij}(\lambda) \end{pmatrix} , \qquad |\vec{u}\rangle = \prod_{j=1}^{n_1} B_{1i_j}(u_j^1)|\{i_j\}\rangle \ .$$

Bethe vectors and Nested Bethe Ansatz

The Algebraic Bethe Ansatz is a very powerful tool, but it only works for systems whose symmetry is of rank 1 ($SU(2)$, $SL(2)$ and $SU(1|1)$). For higher ranks we need to use what is called "nesting procedure". The basic idea is to apply some modification of the ABA to each level of the filtration defined by nested $SU(n)$ groups.

The method [Kulish, Reshetikhin, 1983] goes as follows, we define

$$T^N(\lambda) = \begin{pmatrix} A_{11}(\lambda) & B_{1i}(\lambda) \\ C_{i1}(\lambda) & D_{ij}(\lambda) \end{pmatrix} , \qquad |\vec{u}\rangle = \prod_{j=1}^{n_1} B_{1i_j}(u_j^1)|\{i_j\}\rangle .$$

The vector $|\vec{u}\rangle$ is an eigenvector of $\text{tr}[T^N]$ provided that the vector $|\{i_j\}\rangle$ is an eigenvector of $\text{tr}[D_{ij}(u)\prod_{j=1}^{n_1} R^{N-1}(u-u_j^1)]$ and the rapidities fulfil (a modified version of) the Bethe equations.

Bethe equations and other representations

In particular we will have a set of Bethe equations for each level of the filtration. Luckily all the sets can be written as

$$\frac{\lambda^k(u_j^k)}{\lambda^{k+1}(u_j^k)} = \prod_{l \neq j} \frac{f(u_j^k, u_l^k) f(\bar{u}^{k+1}, u_l^k)}{f(u_l^k, u_j^k) f(u_l^k, \bar{u}^{k-1})}$$

where $A_{kk}(x)|0\rangle = \lambda^k(x)|0\rangle$.

Bethe equations and other representations

In particular we will have a set of Bethe equations for each level of the filtration. Luckily all the sets can be written as

$$\frac{\lambda^k(u_j^k)}{\lambda^{k+1}(u_j^k)} = \prod_{l\neq j} \frac{f(u_j^k, u_l^k) f(\bar{u}^{k+1}, u_l^k)}{f(u_l^k, u_j^k) f(u_l^k, \bar{u}^{k-1})}$$

where $A_{kk}(x)|0\rangle = \lambda^k(x)|0\rangle$.

There exists other ways of computing the Bethe vectors in the nested Bethe Ansatz, like the Tarasov-Varchenko Trace formula or the projection of Drinfeld currents. However, all of them are present an important level of lengthiness and cumbersomeness at some particular point of the computation.

Breaking the Bethe vector

Despite these problems, the Bethe vectors still behave well in the composite model (Bethe vectors are broken into Bethe vectors). In particular

$$\mathbb{B}(\vec{\lambda})|0\rangle = \sum f(\bar{\beta}^N, \beta^N) \prod_{k=1}^{N-1} \frac{\lambda_2^k(\beta^k) f(\bar{\beta}^k, \beta^k)}{\lambda_2^{k+1}(\beta^{k+1}) f(\bar{\beta}^{k+1}, \beta^k)} \mathbb{B}_1(\beta)\mathbb{B}_2(\bar{\beta})|0\rangle$$

where $\mathbb{B}$ are actually sums of products of the different B_{ij} we have and a normalization factor. Similarly for the $\mathbb{C}$ operator. At the end, we can repeat again the same process as before and write

$$K_m = \frac{\prod_{k=1}^{N} f(\bar{\gamma}^k, \gamma^k) f(\beta^k, \bar{\beta}^k)}{\prod_{k=1}^{N-1} f(\bar{\gamma}^{k+1}, \gamma^k) f(\beta^{k+1}, \bar{\beta}^k)} HW(\gamma, \beta) LW(\bar{\gamma}, \bar{\beta}) \ .$$

Generalized Gaudin determinant

After some work (which I am not going to reproduce here due to the lack of time and abundance of indices), one can see that

$$\langle 0|\mathbb{C}(\vec{u})\mathbb{B}(\vec{u})|0\rangle = \frac{\prod f^{\neq}(\vec{u}^k, \vec{u}^k)}{\prod f(\vec{u}^{k+1}, \vec{u}^k)} \det G \ ,$$

$$G_{ij}^{kl} = -u_i^k \frac{\partial}{\partial u_i^k} \left[\frac{\lambda^l(u_j^l)}{\lambda^{l+1}(u_j^l)} \prod_{q\neq j} \frac{f(u_q^l, u_j^l) f(u_q^l, \bar{u}^{l-1})}{f(u_j^l, u_q^l) f(\bar{u}^{l+1}, u_q^l)} \right] \ ,$$

notice the striking similarity of the final answer with the usual Gaudin determinant.

Why do we get determinants? [Belliard, Slavnov, 2019]

A determinant for everyone

We have just seen that for this particular R-matrix, the scalar product of Bethe vectors can be written in terms of determinant. In [Hutsalyuk, Liashyk, Pakuliak, Ragoucy, Slavnov, 2017], the authors proved that the norm of on-shell Bethe vectors in models with $U_q(\hat{\mathfrak{gl}}_m)$ symmetry can be written in terms of a determinant.

A determinant for everyone

We have just seen that for this particular R-matrix, the scalar product of Bethe vectors can be written in terms of determinant. In [Hutsalyuk, Liashyk, Pakuliak, Ragoucy, Slavnov, 2017], the authors proved that the norm of on-shell Bethe vectors in models with $U_q(\hat{\mathfrak{gl}}_m)$ symmetry can be written in terms of a determinant.

Why do determinants appear so often when computing scalar products of Bethe vectors?

Back to linear algebra

An interesting argument was given in [Belliard, Slavnov, 2019]. Take a set of N rapidities that fulfil the Bethe equations $\bar{v}$ and a set of $N+1$ arbitrary complex numbers $\bar{u}$, and define the scalar product $X_j = \langle C(\vec{v}) B(\vec{u}_j) \rangle$. The matrix element

$$\langle C(\vec{v}) | \tau(u_j) | B(\vec{u}_j) \rangle \ ,$$

can be computed in two ways. Acting on the left state it is just $\Lambda(u_j, \vec{v}) X_j$, while if we act in the right state, we will get a linear combination of all X_j's

$$\langle C(\vec{v}) | \tau(u_j) | B(\vec{u}_j) \rangle = \sum_{k=1}^{N+1} L_{j,k}(\vec{v}, \vec{u}) X_k \ .$$

Consistency forces this set of scalar products to satisfy the homogeneous linear system of equations

$$\sum_k \left(L_{j,k} - \delta_{j,k}\Lambda(u_j, \vec{v})\right) X_k = \sum_k \mathfrak{M}_{jk} X_k = 0 \ ,$$

and a solution (if exists) can be written in terms of determinants of the minors of the matrix.

Consistency forces this set of scalar products to satisfy the homogeneous linear system of equations

$$\sum_k \left(L_{j,k} - \delta_{j,k} \Lambda(u_j, \vec{v}) \right) X_k = \sum_k \mathfrak{M}_{jk} X_k = 0 \ ,$$

and a solution (if exists) can be written in terms of determinants of the minors of the matrix.
As the matrix L_{jk} can be computed from the residues of $\Lambda(u_j, \vec{v})$, we do not need to actually construct the states for this method to work. This allows us to extend the reasoning to cases where we do not have a pseudovacuum.

The six-vertex model, again

If we consider again the rational six vertex model

$$R(u,v) = \mathbb{1} + g(u-v)\mathbb{P} \ , \qquad g(u,v) = \frac{c}{u-v} \ ,$$

we have that

$$\Lambda(z,\vec{v}) = g(z,\vec{v})\mathfrak{Y}(z,\vec{v}) \ , \qquad L_{jk} = g(u_k,\vec{u}_k)\mathfrak{Y}(u_k,\vec{u}_j)$$

where $\mathfrak{Y}(z,\vec{v})$ is a linear combination of symmetric polynomials of $\vec{v}$ with coefficients that depend on z.

The six-vertex model, again

If we consider again the rational six vertex model

$$R(u,v) = \imath + g(u-v)\mathbb{P} \ , \qquad g(u,v) = \frac{c}{u-v} \ ,$$

we have that

$$\Lambda(z,\vec{v}) = g(z,\vec{v})\mathfrak{Y}(z,\vec{v}) \ , \qquad L_{jk} = g(u_k,\vec{u}_k)\mathfrak{Y}(u_k,\vec{u}_j)$$

where $\mathfrak{Y}(z,\vec{v})$ is a linear combination of symmetric polynomials of $\vec{v}$ with coefficients that depend on z.

One can check that indeed the matrix of coefficients $\mathfrak{M}_{jk}$ indeed has vanishing determinant as there exists a non-trivial linear combination of the columns that gives a zero

$$\sum_{j=1}^{N+1} g(u_{N+1},\vec{u}_{N+1})\frac{g(u_k,w_j)}{g(u_k,\vec{w})}\mathfrak{M}_{jl} = 0 \ ,$$

where $\bar{w}$ is a set of $N+1$ generic pair-wise distinct complex numbers.

The six-vertex model, again

After some manipulations, the system can be rewritten as

$$\frac{X_l}{g^{<}(\vec{u}_l, \vec{u}_l)\Omega_l} = \frac{X_r}{g^{<}(\vec{u}_r, \vec{u}_r)\Omega_r}$$

for any pair l and r. Here Ω_l is the determinant of the $N \times N + 1$ matrix

$$\Omega_{jk} = \frac{c}{g(u_k, \vec{v})} \frac{\partial \Lambda(u_k, \vec{v})}{\partial v_j} \ ,$$

computed after removing column l. This equation can be understood as some kind of separation of variables (as the lhs does not depend on u_l while the rhs does not depend on r, thus

$$X_l = \Phi(\vec{v}) g^{<}(\vec{u}_l, \vec{u}_l)\Omega_l \ ,$$

with $\Phi(\vec{v})$ a general function.