Anharmonic Oscillator and Thermodynamic Bethe Ansatz [Medium]

This problem explores a curious duality that exists between quantum mechanics in $0+1$ dimensions (i.e. Schroedinger equation) and certain integrable $1+1$ dimensional QFT's. It will allow us to find the spectrum of Schroedinger equation with x^4 potential using TBA methods. The original reference is the classic paper of Dorey and Tateo hep-th/9812211. We will consider the spectral problem associated with the Schroedinger equation

$$\left(-\frac{d^2}{dx^2}+x^4-E_k\right)\psi_k(x)=0\,. \tag{1}$$

The dual system is a 1+1 dimensional QFT associated with a thermal perturbation of a system of $\mathbb{Z}_4$ parafermions. This is an integrable theory. It contains 3 particle species with masses $m_1=m_2/\sqrt{2}=m_3\equiv m$. The key object is then the (so-called massless limit of the) TBA equation governing the ground state energy of this theory on a circle of radius

$$R=m^{-1}\sqrt{2\pi}\frac{\Gamma(5/4)}{\Gamma(7/4)}\,.$$

The TBA is a set of three nonlinear integral equations of the form

$$\log Y_a(\theta)=m_aRe^\theta-\sum_{b=1}^{3}K_{ab}(\theta-\theta')*\log(1+1/Y_b(\theta'))\ ,\qquad a=1,2,3\,, \tag{2}$$

where star stands for convolution

$$f*g=\int_{-\infty}^{+\infty}d\theta' f(\theta-\theta')g(\theta')\,. \tag{3}$$

The kernels are given by $K_{ab}=\frac{1}{2\pi i}\partial_\theta\log S_{ab}(\theta)$ where $S_{ab}(\theta)$ is the infinite volume $2\to 2$ scattering matrix given by

$$S_{ab}(\theta)=\prod_{p=|a-b|+1\ (\text{step }2)}^{a+b-1}\frac{\sinh(\frac{\theta}{2}+i\pi\frac{p-1}{8})\sinh(\frac{\theta}{2}+i\pi\frac{p+1}{8})}{\sinh(\frac{\theta}{2}-i\pi\frac{p-1}{8})\sinh(\frac{\theta}{2}-i\pi\frac{p+1}{8})} \tag{4}$$

where θ is the relative particle rapidity.

The remarkable observation of Dorey and Tateo is that the zeros of the Y-function are directly related to the spectrum of the anharmonic oscillator! More precisely, the first Y-function has zeros in the line $\mathrm{Im}(\theta)=3i\pi/4$ and

$$Y_1(\frac{3i\pi}{4}+\theta_k)=0\qquad\text{with}\qquad E_k=e^{3/4\theta_k}\,. \tag{5}$$

In the following exercise we will numerically solve (2), find the zeros of $Y_1(\theta)$, and compare them to the eigenvalues of the Schroedinger problem. We will solve (2) by iteration, which means we will need to evaluate the convolution on the RHS many times.

1 Efficiently evaluating convolutions

As mentioned above we will solve (2) by an iterative method and thus we will need to evaluate the convolution on the RHS many times. The purpose of this section is to demonstrate one method for efficiently evaluating convolutions. Recall the convolution theorem for Fourier transforms which states

$$F[f * g] = F[f]F[g] \tag{6}$$

where $F[f]$ is the Fourier transform of f. Thus we can evaluate the convolutions in (2) as

$$K * \log(1 + 1/Y) = F^{-1}[F[K]F[\log(1 + 1/Y)]] \tag{7}$$

This equality converts the evaluation of the convolution into matrix multiplication which is much more efficient than using `NIntegrate` to evaluate the convolution for each value of θ.

We will now implement the operations F and F^{-1} in Mathematica. We will discretize θ so that all functions of θ are represented by a list of numbers. Define a cutoff $\Theta = 10$ and a grid-spacing $\epsilon = 2^{-8}$ and the function

$$\texttt{l[f\_]:=Table[f,\{}\theta\texttt{,-}\Theta\texttt{,}\Theta - \epsilon\texttt{,}2\Theta\epsilon\texttt{\}]} \tag{8}$$

which evaluates the function $f(\theta)$ on the grid.

1) Implement `l[f_]` and evaluate it on the function $f(\theta) = \theta$ to see the gridpoints. How many points are in the grid? The `Fourier` command is fastest when the size of the grid is a power of 2. Now use `l[f_]` on your favorite function of θ. Use the `ListPlot` command to compare with the `Plot` of the function.

The Fourier and inverse Fourier commands are implemented as

$$\texttt{F[+1][X\_]} \ := \ \texttt{Fourier[X]} \tag{9}$$

$$\texttt{F[-1][X\_]} \ := \ \texttt{RotateRight[}2\Theta/\sqrt{\texttt{Length[X]}}\ \texttt{InverseFourier[X], Length[X]/2]}$$

where the input `X` has already been evaluated on the θ-grid (i.e. `X=l[f]` where `f` is some function of θ). The normalization and RotateRight in the inverse transform are just needed to account for Mathematica's conventions. Now let's test this equipment in an example.

2) Plot the function

$$f(\theta) = \int_{-\infty}^{+\infty} d\theta' \frac{e^{-\cosh\theta'}}{\cosh(\theta - \theta')} \tag{10}$$

in two ways. First using `NIntegrate` at each gridpoint; use `ListPlot` to plot the result. Second using the `Fourier` commands as in (7). Use the command `Timing` to compare the time efficiency. The fourier method should be about 100 times faster.

2 TBA: solution by iteration

In this section we will be solving the TBA (2) by iteration.

1) Implement the S-matrix (4) and the corresponding kernels. You should find there are two different types of kernels. One of the form $1/\cosh\theta$ and another of the form $\cosh\theta/\cosh 2\theta$. You may need to use the command `FullSimplify`.

It follows from (2) that $Y_a \sim e^{m_a e^{\theta}}$ as $\theta \to \infty$ and $Y_a \sim c_a$ as $\theta \to -\infty$ where c_a is θ-independent.

2) Find the constants c_a by setting the Y-functions to constants in (2) and solving the resulting algebraic equations. Why is it OK to set the Y-function in the convolution to a constant in the limit $\theta \to -\infty$? You should find the values $c_1 = c_3 = 2$, $c_2 = 3$.

This difference in the $\theta \to \pm\infty$ asymptotics means that the Y-functions are not approximately periodic on the interval $(-\Theta, \Theta)$ which is not good for the `Fourier` command. For this reason it is useful to add a term to the integrand which smoothly cuts off the $\log(1+1/Y_a)$ for large negative θ and which does not affect the behavior at large positive θ. For example, one could replace

$$K_{a,b} * \log(1+1/Y_b) = K_{a,b} * \mathrm{cut}(\theta)\log(1+1/c_b) + K_{a,b} * [\log(1+1/Y_b) - \mathrm{cut}(\theta)\log(1+1/c_b)] \tag{11}$$

where, for example, one can use

$$\mathrm{cut}(\theta) = 1/2(1+\tanh(-\theta)) \tag{12}$$

so that the term in square brackets in the second term in (11) decays for large negative and positive θ and is thus in prime condition for evaluation using `Fourier`. The first term can be evaluated once and for all.

3) Evaluate the first term on the RHS of (11) on the θ-grid. Since this only needs to be done once it is fine to just use `NIntegrate` for each point. Or you can do it analytically.

4) Write a recursion which computes the nth approximation to the Y-functions `Y[a][n]` by evaluating the RHS of (2) on the previous approximation `Y[a][n-1]` starting with the large θ asymptotic

$$\texttt{Y[a\_][0]= 1[}e^{\texttt{m[a]R}\ e^{\theta}}\texttt{]} \tag{13}$$

You should implement the convolution in the TBA equations using the fourier methods discussed in the previous section. Also, be sure to recall the "memorization trick" so that the LHS of your recursion should start like `Y[a_][n_]:=Y[a][n]= RHS`. If you are not familiar with this trick, see appendix A. Compute several iterations and check that the boundary condition at large negative θ is satisfied. Your Y-functions should converge after around 10 to 20 iterations; compare with figure 1 where I plotted $1/Y_a$ after 20 iterations.

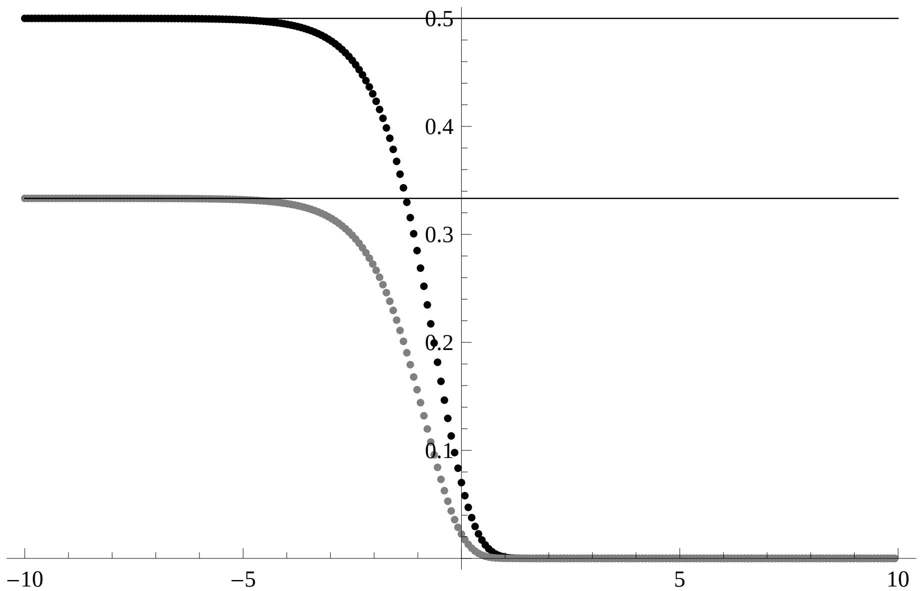

Figure 1: Plot of $1/Y_a(\theta)$ with $1/Y_1 = 1/Y_3$ shown in black and $1/Y_2$ in gray.

3 Extracting spectrum

According to (5) the eigenvalues of the Schroedinger problem appear as zeros of $Y_1(\theta)$, which are located along the line $\text{Im}\theta = i3\pi/4$. In this section we will located these zeros and verify (5).

Consider replacing $\theta \to \theta + i\gamma$ in (2) where here we take θ and γ real, and start with gamma close to zero. Analytically continuing in γ we can determine the Y-functions anywhere in the complex plane in terms of the Y-functions on the real axis (which we have just solved for in the previous problem). Note, however, that as we move γ outside the strip $-\pi/4 \le \gamma \le \pi/4$ poles of the kernels will cross the integration contour. One should analytically continue by deforming the integration contour and picking the residues of these poles. There is a nice way to implement this calculation. The TBA equations (2) imply the functional equations

$$
\begin{aligned}
Y_1(\theta + i\pi/4)Y_1(\theta - i\pi/4) &= (1 + Y_2(\theta)) \\
Y_2(\theta + i\pi/4)Y_2(\theta - i\pi/4) &= (1 + Y_1(\theta))(1 + Y_3(\theta)) \\
Y_3(\theta + i\pi/4)Y_3(\theta - i\pi/4) &= (1 + Y_2(\theta))
\end{aligned}
\tag{14}
$$

We can always use these equations (called the Y-system) to write a Y-function evaluated anywhere in the complex plane in terms of Y-functions evaluated within the strip $-\pi/4 \le \gamma \le \pi/4$.

2) Write $Y_1(\theta + i3\pi/4)$ in terms of Y-functions whose arguments are within $i\pi/4$ of the real axis (θ is real here). Your answer should be a rational function of $Y_1(\theta + i\pi/4)$, $Y_3(\theta + i\pi/4)$ and $Y_2(\theta)$.

We will evaluate $Y_1(\theta + i\pi/4)$ and $Y_3(\theta + i\pi/4)$ in terms of the $Y_a(\theta)$ by replacing $\theta \to \theta + i\gamma$

in the TBA and smoothly continuing γ from zero to $\pi/4$. Note that after the continuation in γ there will be poles of the kernels K_{12} and K_{32} sitting an infinitesimal distance from the integration contour and one needs to use principal value integration to evaluate such integrals and add the appropriate (half) residue by hand.

3) Consider the function

$$f(\theta) = \int_{-\infty}^{+\infty} d\theta' \frac{\cosh(\theta-\theta')}{\cosh 2(\theta-\theta')} e^{-\cosh\theta'} \tag{15}$$

replace $\theta \to \theta + i\gamma$ and analytically continue up to $\gamma = \pi/4$. Plot $f(\theta + i\pi/4)$ on the θ-grid using `NIntegrate` with the option `"Method" -> "PrincipalValue"` and adding the (half) residue by hand. To determine the sign of the residue, carefully consider how the poles of the kernel move when you increase γ (i.e. is the pole above or below the contour when $\gamma \to \pi/4$). You should find the value $f(0.1 + i\pi/4) = 0.638185 - 0.0296788i$.

4) Compute $Y_1(\theta+i\pi/4)$ and $Y_3(\theta+i\pi/4)$ as described just below problem **2)** of this section. You should find the value $Y_1(0.1 + i\pi/4) = Y_3(0.1 + i\pi/4) = -1.95731 + 7.72487i$.

5) Using the results of **2)** and **4)** evaluate $Y_1(\theta + i3\pi/4)$ for several values in the θ-interval $(0, 3)$. Create an interpolation and find the first several zeros. Compare with the values in the table I of hep-th/9812211. Remember the conversion $E = e^{4/3\theta}$.

A Memorization Trick

Here we show a useful way of implementing a function that we want to define with delayed evaluate (`:=`) when we want Mathematica to remember the values of this function she has already computed. Consider computing the Fibonicci numbers defined by the recursion

$$Q_n = Q_{n-1} + Q_{n-2}, \qquad Q_0 = 0,\ Q_1 = 1 \tag{16}$$

Compute the first 30 Fibonnaci numbers using the recursion

```
Qslow[n_]:=Qslow[n-1]+Qslow[n-2]; Qslow[0]=0; Qslow[1]=1;    (17)
```

Now implement using the "Memorization trick"

```
Qfast[n_]:=Qfast[n]=Qfast[n-1]+Qfast[n-2]; Qfast[0]=0; Qfast[1]=1;    (18)
```

The key with this implementation is that Mathematica stores each value of `Qfast` when she computes it. In this way she does not need to recompute it each time it is called. Compare the two methods using the `Timing` command. Try to compute the first 40 Fibonacci numbers with the two different implementations.