HHH three point functions at strong coupling

In this exercise we will determine the leading strong coupling behavior of certain 3-point functions in $\mathcal{N}=4$ SYM by reproducing Figure 3 of 1109.6262. If you are interested in details on the set-up of this computation see 1109.6262. Hint: study in detail the notebook of Pedro's first lecture on solving the Liouville PDE before starting the second part of the Exercise.

1 Analytics (Very easy)

The (AdS contribution to the) regularized Area relevant for computing the structure constant can actually be computed without knowing the precise shape of the surface!, see 1109.6262. It is given by

$$\begin{aligned}\frac{\text{Area}_{reg}}{\pi} &= -2\Delta h(2\Delta)+h\left(2\Delta-\Delta_\infty\right)\left(\Delta-\frac{\Delta_\infty}{2}\right)+h\left(2\Delta+\Delta_\infty\right)\left(\Delta+\frac{\Delta_\infty}{2}\right)\\ &\quad +h\left(\Delta_\infty\right)\Delta_\infty-h\left(2\Delta_\infty\right)\Delta_\infty+\frac{1}{6}\end{aligned} \tag{1}$$

where $h(a)=\int\limits_{-\infty}^{+\infty}\frac{d\theta}{\pi}\cosh\theta\log\left(1-e^{-\pi a\cosh\theta}\right)$.

- Reproduce the right column of Figure 3 of 1109.6262

2 Numerics (Medium)

The PDE

The goal of this exercise is to use Mathematica to numerically solve the nonlinear PDE

$$\partial\bar{\partial}\gamma-\sqrt{T\bar{T}}\sinh\gamma=0 \tag{2}$$

where $\partial=\frac{1}{2}\left(\partial_x-i\partial_y\right)$, $\bar{\partial}=\frac{1}{2}\left(\partial_x+i\partial_y\right)$ and T is the energy-momentum tensor

$$T\left(w\right)=\frac{\Delta_\infty}{4}\frac{w^2+a^2}{\left(1-w^2\right)^2}\,,\quad \bar{T}\left(\bar{w}\right)=\frac{\Delta_\infty}{4}\frac{\bar{w}^2+a^2}{\left(1-\bar{w}^2\right)^2},\qquad a^2=4\frac{\Delta^2}{\Delta_\infty^2}-1 \tag{3}$$

with $w = x + iy$, $\bar{w} = x - iy$. These are the equations relevant for a three point function with two operators of dimension Δ and a third of dimension Δ_∞. For concreteness set $\Delta = \Delta_\infty = 1$. You can change it later. The boundary conditions to be imposed on the solution of the PDE are

$$\gamma \to \log \sqrt{T\overline{T}}, \qquad \text{(at zeros of } T) \tag{4}$$

and

$$\gamma \to 0, \qquad \text{(at poles of } T) \tag{5}$$

In particular, one of the poles of T is at infinity so that $\gamma \to 0$ at infinity.

Numerics

- Code the left hand side of (2) in `Mathematica`.
- We define

$$\gamma_{reg} = \frac{1}{2}\gamma + L_{reg}, \qquad L_{reg} = -\frac{1}{4}\log\left[\frac{(w^2+a^2)\,(\bar{w}^2+a^2)}{w^2\bar{w}^2+a^2}\right] \tag{6}$$

  This new function is more convenient for numerics since it is regular at the zeros of T (it's actually regular everywhere on the Riemann sphere). The price to pay is that the equation becomes much uglier. Find the equation for γ_{reg}.

- Next we change variables

$$\begin{cases} w = x + iy \\ \bar{w} = x - iy \end{cases} \longrightarrow \begin{cases} s = \dfrac{x}{1+x} \\ t = \dfrac{y}{1+y} \end{cases} \tag{7}$$

  which are quite useful because they live in a finite domain, $s \in [0,1]$, $t \in [0,1]$ (note that in this exercise we are interested in the quadrant $x, y > 0$). How does the equation look in these new coordinates?

  Hint: It is useful to first play with a simpler toy model. For example, under $x = e^y$ the differential equation $f + xf' + x^2 f'' = 0$ becomes a simple Harmonic oscillator ODE. We could easily see this by using

```
F[x]+x D[F[x],x]+x^2 D[F[x],{x,2}]==0/.
F->(f[Log[#]]&)/.x->Exp[y]//PowerExpand//Simplify
```

Understand this piece of code and then change equation (2) to (s, t) coordinates.

- Now we discretize.

$$\left[\partial_s^2 \gamma_{reg}\left(s, t\right)\right]^{(n)} \rightarrow \frac{\left(\gamma_{reg}^{\mathbf{(n-1)}}\left(s+\varepsilon, t\right) - 2\gamma_{reg}^{\mathbf{(n)}}\left(s, t\right) + \gamma_{reg}^{\mathbf{(n-1)}}\left(s-\varepsilon, t\right)\right)}{\varepsilon^2} \quad (8)$$

  Make this replacement in your PDE (in s and t variables, of course). Do similar for $\partial_t^2 \gamma_{reg}\left(s, t\right)$. For single derivatives use $\gamma_{reg}^{\mathbf{(n-1)}}$ only. Also replace all γ_{reg}'s appearing elsewhere by $\gamma_{reg}^{\mathbf{(n-1)}}$. Solve for $\gamma_{reg}^{\mathbf{(n)}}\left(s, t\right)$ in terms of $\gamma_{reg}^{\mathbf{(n-1)}}$, that is in terms of γ_{reg} at the previous iteration. Hint: This is very similar to the lecture on the Liouville equation.

- The boundary conditions in the s and t variables read

$$\gamma\left(s, 0\right) = 0 \quad (9)$$
$$\gamma\left(1, t\right) = 0 \quad (10)$$
$$\gamma\left(s, 1\right) = 0 \quad (11)$$
$$\partial_s \gamma\left(t, s\right)|_{s=0} = 0 \quad (12)$$

  What do they become for γ_{reg}? For Example, for the condition with the derivative you should find

$$\partial_s \gamma_{reg}\left(t, s\right)|_{s=0} = 0 \quad (13)$$

- Code the boundary conditions for γ_{reg}. Hint: again, see lecture on Liouville. One way of implementing (13) is (γ_{reg} =g)

```
g[n_][0, t_] := g[n - 1][\[Epsilon], t]
```

  Explain why.

- Code some reasonable $\gamma^{\mathbf{(0)}}(s, t)$ to start the iterations.

- Define some number of points in our lattice discretization, for example $\Lambda = 17$ and $\epsilon = 1/\Lambda$.

- Compute $\gamma_{reg}((\Lambda+1)/(2\Lambda), (\Lambda+1)/(2\Lambda))$ after 10 iterations.

  At this point the difficulty level of the exercise changes. It is now *Hard*

- Compute Compute $\gamma_{reg}((\Lambda+1)/(2\Lambda), (\Lambda+1)/(2\Lambda))$ for 100, 1000 and even 2000 iterations. The last ones should take a few minutes! If you use $\gamma_{reg} = 0$ for starting point you should find something like

  ```
  {{10,0.0592509},{200,0.324321},{1000,0.377323},{2000,0.380725}}
  ```

- `ListPlot` the sequence of numbers above (or whatever you find). To observe that the iterations are indeed converging.

- The regularized area is given by

$$\text{Area}_{reg} = \int\limits_{x,y>0} dw d\bar{w}\, 4\sqrt{T\bar{T}}(\cosh\gamma - 1) \tag{14}$$

  Compute the integrand in terms of s, t and also γ_{reg}. Don't forget the Jacobian from the change of variables.

- Now we need to compute this integral. Compute it by replacing it by a Riemann sum. As a function of the number of iterations you should find something like

  ```
  {{200, 0.2456}, {500, 0.368051}, {1000, 0.414301}, {2000, 0.432258}}
  ```

  If you want to do slightly better define an `Interpolation` to create a numerical function γ_{reg} and then use `NIntegrate`. Compare with the third line in Figure 3 (page 31) of arXiv:1109.6262. In this way you should be able to get at least 2 digits of accuracy.

- Compare with the analytic predictions for these particular values of Δ's.

- Very hard: Compute the other values of Δ in Figure 3 of arXiv:1109.6262.

- Really very hard: Google for other relaxation methods. Try implementing non-local ones and compare the performance and accuracy compared with the local one used so far.