Large N Sachdev-Ye-Kitaev model

Subject: Fourier transforms, iterative methods.

Introduction

The Sachdev-Ye-Kitaev (SYK) model is solved in the large-N limit by the Schwinger-Dyson (SD) equations

$$\frac{1}{G(\omega)} = -i\omega - \Sigma(\omega) \qquad \text{with} \qquad \Sigma(\tau) = J^2 G(\tau)^3 \,, \tag{1}$$

where $G(\omega)$ is the Fourier transform of the Euclidean two-point function $G(\tau) = \langle \mathrm{T}\, \psi^i(\tau)\psi^i(0)\rangle$; $\Sigma(\tau)$ is the self-energy; and $\Sigma(\omega)$ is its Fourier transform.

At finite temperature $T = 1/\beta$, Euclidean time is compactified in the interval $\tau \in [0, \beta)$ and the two-point function is constrained by the KMS conditions to be antisymmetric, $G(\tau + \beta) = -G(\tau)$. Then the SD equations are to be solved over the discrete set of Matsubara-Fourier frequencies

$$\omega_k = \frac{2\pi}{\beta}\left(k - \tfrac{1}{2}\right) \qquad \text{with} \qquad G(\tau) = \frac{1}{\beta}\sum_{k=-\infty}^{\infty} G_k e^{-i\omega_k \tau} \,. \tag{2}$$

Once the SD equations are solved, all thermodynamic properties of the system are immediately known. For example, the entropy can be computed as

$$\frac{S}{N} = \frac{1}{2}\ln 2 - \sum_{k=1}^{\infty} \ln\left|\frac{G_k}{G_k^{(0)}}\right| + \frac{5}{4}\left[1 - \mathrm{Re}\left(\frac{G_k}{G_k^{(0)}}\right)\right] \,, \tag{3}$$

where $G_k^{(0)} = (-i\omega_k)^{-1}$ are the Matsubara-Fourier coefficients of the free fermion solution $G^{(0)}(\tau) = \frac{1}{2}\mathrm{sign}(\tau)$. Note that this expression depends on the temperature and coupling only through the two-point function's Matsubara-Fourier coefficients G_k. It has the correct high-temperature limit $S_\infty = \frac{N}{2}\ln 2$, since in this limit the fermions are essentially free and $G_k \approx G_k^{(0)}$, but its low-temperature limit is *a priori* unclear. A low-energy analytic calculation, exploiting the fact that the theory becomes nearly conformal in the IR, results in

$$S_0 = \frac{\mathrm{Cat}}{2\pi} + \frac{\log 2}{8} \approx 0.23\,. \tag{4}$$

Problem statement

At finite temperature the SD equations (1) become an infinite set of coupled equations for the Matsubara-Fourier coefficients G_k of the finite-temperature two-point function,

$$G_k = (-i\omega_k - \Sigma_k)^{-1} \qquad \text{for} \qquad k \in \mathbb{Z}\,, \tag{5}$$

where $\Sigma_k \equiv \Sigma(\omega_k)$ could in principle be obtained by plugging the sum of (2) in the definition of the self-energy in (1). Doing so we would get

$$\Sigma_k = \left(\frac{J}{\beta}\right)^2 \sum_{k_1,k_2=-\infty}^{\infty} G_{k_1} G_{k_2} G_{k-k_1-k_2+1}\,, \tag{6}$$

which makes it evident that there is only one parameter β/J in this problem. We will therefore set $J = 1$ hereafter in order to fix the energy units.

Note that the system of equations we have obtained is somewhat reminiscent of a fixed-point equation $x = f(x)$. One could hope to solve it by repeated iteration, $x^{[i+1]} = f(x^{[i]})$, given an initial guess $x^{[0]}$. In order to proceed in this direction, we will first introduce a frequency cutoff K such that we only consider the $2K$ frequencies ω_k for $k = -K+1, \ldots, K$. Starting with an initial guess $G_k^{[0]}$ for the solution, we could then compute successive approximations $G_k^{[i]}$ for $i = 1, 2, \ldots$ by iterating (5) with the self-energy obtained from the previous step,

$$G_k^{[i+1]} = \left(-i\omega_k - \Sigma_k^{[i]}\right)^{-1} \qquad \text{for} \qquad k = -K+1, \ldots, K\,. \tag{7}$$

However, the coupled system we are dealing with is unstable, so in practice we need to introduce a weighted update instead,

$$G_k^{[i+1]} = (1-\alpha) G_k^{[i]} + \alpha \left(-i\omega_k - \Sigma_k^{[i]}\right)^{-1} \qquad \text{for} \qquad k = -K+1, \ldots, K\,. \tag{8}$$

The parameter α appearing above is *a priori* arbitrary, and needs to by chosen through trial-and-error.

We will assume the repeated iteration algorithm has achieved convergence when the average change of the Matsubara-Fourier coefficients is smaller than some accuracy parameter ε, *i.e.*

$$\frac{1}{2K}\sum_{k=-K+1}^{K}\left|G_k^{[i]}-G_k^{[i-1]}\right|<\varepsilon \qquad \text{un units of } J=1\,. \tag{9}$$

Your task is to write a function `SDSolve`, taking the physical parameter β and the algorithmic parameters K, α and ε, to implement the algorithm described above for solving the SD equations. For reference, you should be able to obtain solutions similar to those in Fig. 1 fairly quickly.

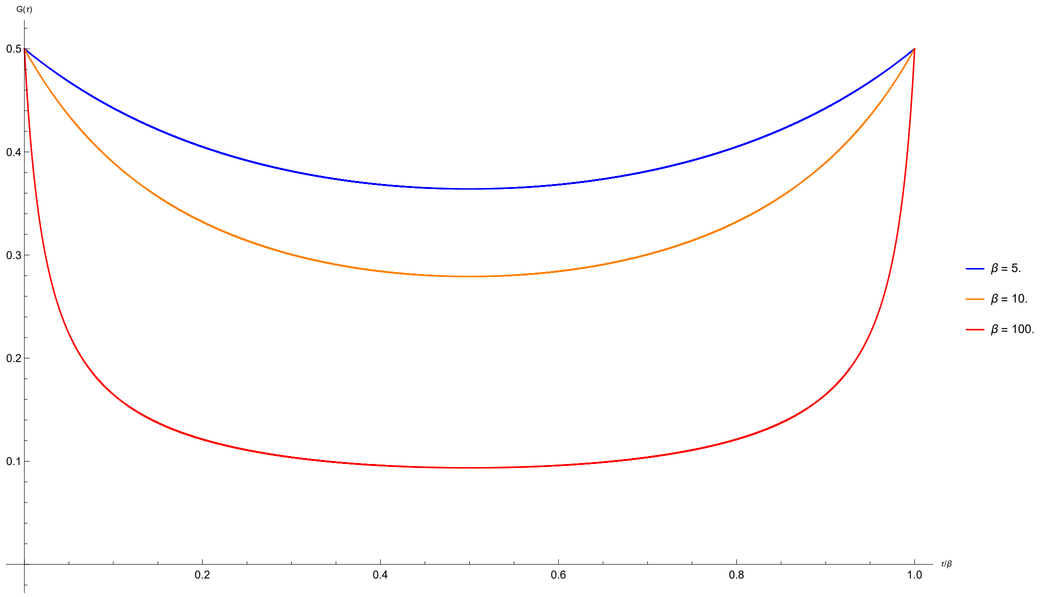

Figure 1: Examples of solutions to the SD equations (1) for $\beta = 5, 10, 100$ with $J = 1$.

Once implemented, your solution should allow you to obtain a fairly good approximation of SYK's zero-temperature entropy (4).

Details

The algorithm as described in the previous section has a running time that is proportional to $\mathcal{O}(K^3)$: for each value of $k = -K+1, \ldots, K$, equation (6) requires us to take a double sum over $k_1, k_2 = -K+1, \ldots, K$. Even if the number of iterations required to reach convergence were to be rather small, say of order $\mathcal{O}(1)$, we would still have an $\mathcal{O}(K^3)$ algorithm and be limited to take $K \sim 10^2 - 10^3$ in practice.

However, we can do much better if we notice that the calculation of the self-energy is simpler in coordinate space than it is in Fourier space. Indeed, if we can obtain the two-point function $G(\tau)$ at a set of $2K$ points τ_j for $j = 1, \ldots, 2K$, the computation of the associated self-energy becomes an $\mathcal{O}(K)$ calculation: we can simply take the second equation in (1) on a point-by-point basis.

Now, obtaining $G(\tau_j)$ from G_k amounts to performing the inverse discrete Fourier transform defined in (2), and the Fast Fourier Transform (FFT) algorithms achieve exactly this goal with running time $\mathcal{O}(K \log K)$. They are implemented in `Mathematica` by `Fourier` and `InverseFourier`, so it is particularly simple to take advantage of them. With this improvement, the inner workflow of the algorithm becomes

$$G_k^{[i]} \xrightarrow[\mathcal{O}(K\log K)]{\texttt{InverseFourier}} G^{[i]}(\tau_j) \xrightarrow[\mathcal{O}(K)]{\text{Eq. (1)}} \Sigma^{[i]}(\tau_j) \xrightarrow[\mathcal{O}(K\log K)]{\texttt{Fourier}} \Sigma_k^{[i]} \xrightarrow[\mathcal{O}(K)]{\text{Eq. (8)}} G_k^{[i+1]} . \tag{10}$$

The algorithm's running time is then $\mathcal{O}(I \times K \log K)$, where I is the number of iterations required to reach convergence (depending mostly on α and ϵ, but also less prominently on K). This should make it possible to reach $K \sim 10^5$, resulting in greater precision and the ability to reach lower temperatures.

Some comments are in order:

- `Mathematica` implements `Fourier` to perform the transformation

  $$G_k = \frac{1}{n^{(1-a)/2}} \sum_{j=1}^{n} G(\tau_j) e^{2\pi i b (j-1)(k-1)/n} , \tag{11}$$

  where a and b are set by `FourierParameters`. You should choose the $n = 2K$ discretization points $\tau_j \in [0, \beta)$ in order to match this definition, and be prepared to "shift" your arrays before and after applying

`Fourier/InverseFourier` (these details are to be worked out beforehand in pen and paper).

- The KMS condition $G(\tau+\beta) = -G(\tau)$ means the two-point function is discontinuous at $\tau = n\beta$ for $n \in \mathbb{Z}$. The introduction of a high-frequency cutoff therefore results in numerical artifacts (oscillations) appearing close to these discontinuities. In order to produce nice plots as in Fig. 1 you will need to correct this upon the final step to obtain $G(\tau)$. The easiest way to do it consists in subtracting a function with the same discontinuities before using FFT, and adding back it's (exact) transform afterwards.

- In order to reproduce the zero-temperature entropy (4), you can either evaluate (3) at a very low temperature, or extrapolate to $T = 0$ a few results obtained at not-so-low temperatures.

Optional: part I

Up to now we have only glossed over a hidden parameter of the algorithm, namely the initial guess $G_k^{[0]}$ for the Matsubara-Fourier coefficients we use to start the iterative process. This can affect the convergence of the algorithm, since starting too far away from the actual solution may lead down a non-converging or "explosive" path through which our tentative solution keeps getting worse instead of improving. We can diagnose this by keeping track of the average change in the computed coefficients, *e.g.* using `Dynamic`, and sometimes mitigate the effect by playing with the α weighting factor.

For greater control, we may decide to include a new argument in our `SDSolve` function to specify what the initial guess $G_k^{[0]}$ should be. This will allow us to first compute a high-temperature solution, for which $G_k^{[0]} = G_k^{(0)}$ should be a good initial guess, and then slowly decrease the temperature by using the solution found in one step as a starting point for the next step.

Optional: part II

A complex-fermion generalization of the SYK model has the Schwinger-Dyson equations

$$\frac{1}{G(\omega)} = m - i\omega - \Sigma(\omega) \qquad \text{with} \qquad \Sigma(\tau) = -J^2 G(\tau)^2 G(-\tau)\,, \tag{12}$$

where m is a mass parameter that we can introduce for Dirac fermions, and now the entropy is computed as[1]

$$\begin{aligned}\frac{S}{N} = \log(1+e^{\beta m}) - \frac{\beta m}{e^{-\beta m}+1} \\ - 2\sum_{k=1}^{\infty} m\mathrm{Re}\left(G_k - G_k^{(0)}\right) + \ln\left|\frac{G_k}{G_k^{(0)}}\right| + \frac{5}{4}\left[1 - \mathrm{Re}\left(\frac{G_k}{G_k^{(0)}}\right)\right]\,.\end{aligned} \tag{13}$$

Making the corresponding changes to your `SDSolve` function, use it to solve the SD equations for various values of the mass and the temperature, "tracing solutions" as the parameters are varied. For low enough temperatures, you should be able to observe the hysteresis phenomenon first reported in [2], see Fig. 7 of [3].

Useful functions

You may find it useful to read the `Mathematica` help pages of the following functions:

- **FFT:** `Fourier`, `InverseFourier`, `FourierParameters`, ...

References

[1] J. Maldacena and D. Stanford, "Remarks on the Sachdev-Ye-Kitaev model," Phys. Rev. D **94** (2016) no.10, 106002 [arXiv:1604.07818 [hep-th]].

[1] Note that N Dirac fermions have twice as many degrees of freedom as N Majorana fermions, so now $S_\infty = N\log 2$.

[2] T. Azeyanagi, F. Ferrari and F. I. Schaposnik Massolo, “Phase Diagram of Planar Matrix Quantum Mechanics, Tensor, and Sachdev-Ye-Kitaev Models,” Phys. Rev. Lett. **120** (2018) no.6, 061602 [arXiv:1707.03431 [hep-th]].

[3] F. Ferrari and F. I. Schaposnik Massolo, “Phases Of Melonic Quantum Mechanics,” Phys. Rev. D **100** (2019) no.2, 026007 [arXiv:1903.06633 [hep-th]].