

Question

第六章作业（必做，12分）

已知自旋1/2 XXZ反铁磁链 $H = \sum (S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + \Delta S_i^z S_{i+1}^z)$ 在 $\Delta \in [-1.5, 1.5]$ 存在两个（量子）相变点。试用Lanczos算法，计算周期边界条件下 H 的基态能量 E_g ，并以 Δ 为横坐标 E_g 为纵坐标作图，找出相变点的位置。链长可以根据机器内存大小，取 $L=12$ ，或者14，或者16，皆可。

1. 不要使用对称性；
2. 写文档介绍计算过程，并确定相变点（找出一个即可）；
3. 程序要求能在linux下运行；
3. 注意交作业的时间。

理论分析

1. Krylov subspace

在介绍lanczos算法之前，我们先介绍一下Krylov子空间。它可以说是处理大型矩阵迭代方法，如Arnoldi、lanczos等，的基本原理。

对已知的矩阵A和任意的矢量b，Krylov子空间定义为：

$$\mathcal{K}(A, q_1, k) = \text{span} \{q_1, Aq_1, \dots, A^{k-1}q_1\} = \text{span} \{q_1, q_2, \dots, q_k\}$$

当然，只有前半部分涉及对Krylov子空间的定义，后面是该子空间的一组正交完备基；这是我们进行大型矩阵对角化的一般思路。至于为什么这样取Krylov子空间，可以参考MATRIX COMPUTATIONS^[1]的第十章。

2. Lanczos algorithm

对于厄米矩阵H，和任意给定向量f；我们可以构造 $HF = F\Lambda + r f_{k+1}$ ，进而可以用F的本征值来估计H的本征值，理论上可以证明，最小和最大本征值是可以很好的估计的。

根据Krylov子空间的定义，我们可以证明下述引理：

$$Hf_k = \beta_{k-1}f_{k-1} + \alpha_k f_k + \beta_k f_{k+1}$$

Proof:

$$\begin{aligned} F_{ij} &= f_i^\dagger H f_j \\ F_{ij}^* &= f_j^\dagger H^\dagger f_i = f_j^\dagger H f_i \end{aligned}$$

由于 $Hf_j \in \text{span} \{f_1, f_2, \dots, f_{j+1}\}$ ，因此当 $i > j + 1$ 或 $j > i + 1$ 时， $F_{ij} = 0$ 。

上面我们直接给出了F矩阵元的表达式，当然只是为了方便起见，下面我们将给出Lanczos算法的构造过程。

Given Hermite matrix H and arbitrary vector f :

$$\begin{aligned} f_1 &= f / \|f\|_2 \\ \omega_1 &= H f_1 \\ \alpha_1 &= \omega_1^\dagger f_1 \\ \omega'_1 &= \omega_1 - \alpha_1 f_1 \end{aligned}$$

$$\begin{aligned}\beta_1 &= \|\omega'_2\|_2 \\ f_2 &= \omega'_1/\beta_1 \\ \rightarrow Hf_1 &= \alpha_1 f_1 + \beta_1 f_2\end{aligned}$$

上面其实就是简单的Gram-Schmit正交化的过程。需要注意的是，由于H是厄米的，因此 α_1 一定是一个实数。还有就是 β_1 的表达式可以转换一下：
 $\beta_1^2 = (\omega - \alpha_1 f_1)^\dagger (\omega - \alpha_1 f_1) = (Hf_1)^\dagger Hf_1 - \alpha^2 = (\alpha_1 f_1 + \beta_1 f_2)^\dagger Hf_1 \Rightarrow \beta_1 = f_2^\dagger Hf_1$ 。这里我们另 β 为实数。

我们再推几个：

$$\begin{aligned}\omega_2 &= Hf_2 \\ \alpha_2 &= \omega_2^\dagger f_2, \beta'_1 = f_1^\dagger \omega = f_1^\dagger Hf_2 \\ \omega'_2 &= \omega_2 - \beta'_1 f_1 - \alpha_2 f_2 \\ \beta_2 &= \|\omega'_2\|_2 \\ f_3 &= \omega'_2/\beta_2 \\ \rightarrow Hf_2 &= \beta'_1 f_1 + \alpha_2 f_2 + \beta_2 f_3\end{aligned}$$

注意到 $\beta_1^{\dagger} = \beta_1$ and $\beta_1 \in \mathbb{R}$, so $\beta_1' = \beta_1$ 。

剩下的我就不写了，我们最后可以得到：

$HF = F\Lambda + \beta_k I f_{k+1}^\top$

其中 $F = \begin{bmatrix} f_1 & f_2 & \cdots & f_k \end{bmatrix}, \Lambda = \begin{bmatrix} \alpha_1 & \beta_1 & & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{k-1} \\ 0 & \cdots & & \beta_{k-1} & \alpha_k \end{bmatrix}$ 。

设 \mathbf{x} 是 Λ 的本征矢，则 $HF\mathbf{x} = \lambda F\mathbf{x} + \beta_k I f_{k+1}^\top \mathbf{x}$ 。

3. pseudocode

```
beta0 = (0), q(0) = 0, b = arbitrary, q(1) = b/norm(b)

for n = 1, 2, 3, ...
    v = Aq(n) // of Aq(n) - beta(n-1)q(n-1) for greater stability
    alpha(n) = \dag{q(n)}*v
    v = v - beta(n-1)q(n-1) - alpha(n)q(n)
    beta(n) = norm(v)
    q(n+1) = v/beta(n)
```

4. spin-1/2 XXY Antiferromagnetism

$$H = \sum (S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + \Delta S_i^z S_{i+1}^z)$$

在周期性边界条件下，及第一个和最后一个的自旋进行耦合。

程序编写

1. function m_Hf

我采用"black box"的形式，不给出哈密顿量对应的矩阵形式，而是直接用位运算flapBit。

这样我们可以节省存储哈密顿量的那个超大空间。但是这只适用于spin-1/2。

自旋1/2的粒子与计算机的二进制正好对应，因此，我们将采用位运算的方式。另外注意到：

$$S^x|\uparrow\rangle = 1/2|\downarrow\rangle, S^x|\downarrow\rangle = 1/2|\uparrow\rangle; S^y|\uparrow\rangle = i/2|\downarrow\rangle, S^y|\downarrow\rangle = -i/2|\uparrow\rangle$$

正好对应于位运算中的翻转运算，因此我们避免使用梯子算符来计算而是直接使用自旋分量算符。

对于海森堡链：

$$\begin{aligned}\sum_i (S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + \Delta S_i^z S_{i+1}^z) |\dots \uparrow_i \uparrow_{i+1} \dots\rangle &= \sum_i (\Delta/4 |\dots \uparrow_i \uparrow_{i+1} \dots\rangle) \\ \sum_i (S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + \Delta S_i^z S_{i+1}^z) |\dots \downarrow_i \downarrow_{i+1} \dots\rangle &= \sum_i (\Delta/4 |\dots \downarrow_i \downarrow_{i+1} \dots\rangle) \\ \sum_i (S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + \Delta S_i^z S_{i+1}^z) |\dots \uparrow_i \downarrow_{i+1} \dots\rangle &= \sum_i (-\Delta/4 |\dots \uparrow_i \downarrow_{i+1} \dots\rangle + 1/2 |\dots \downarrow_i \uparrow_{i+1} \dots\rangle) \\ \sum_i (S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + \Delta S_i^z S_{i+1}^z) |\dots \downarrow_i \uparrow_{i+1} \dots\rangle &= \sum_i (-\Delta/4 |\dots \downarrow_i \uparrow_{i+1} \dots\rangle + 1/2 |\dots \uparrow_i \downarrow_{i+1} \dots\rangle)\end{aligned}$$

2. function m_lanczos

首先，在存储alpha和beta的算法上，我们有两种选择：1. 增加时间复杂度，我们先赋一个较小的空间；在迭代过程中进行判断是否需要扩充。 2. 增加空间复杂度，将alpha、beta的大小就设为Hilbert空间的维数。
我选择后者。

关于的存储，2个就够了，来回访问。因为我们不算特征向量。

需要注意的是循环结束的条件，这里需要计算三对角矩阵的本征值，最小的那个及对应基态能量的估计值；因此当两次迭代出来的E相差不大时，我们认为其收敛。

当然，这种中间涉及到一些理论，即为什么最小的本征值对应基态能量？在^[1:1]的书中有介绍，是有定理保证的。

3. NOTE

最后说一下我在编写时遇到的问题：

- 1. 在构造哈密顿量作用时，要注意系数的对应，否则会出问题。
- 2. calloc才是会把变量初始化为零的函数，malloc不是。这个我用了好久都没有问题，这次有问题了，不过是放在循环内部，可能局部变量空间比较小。

结果

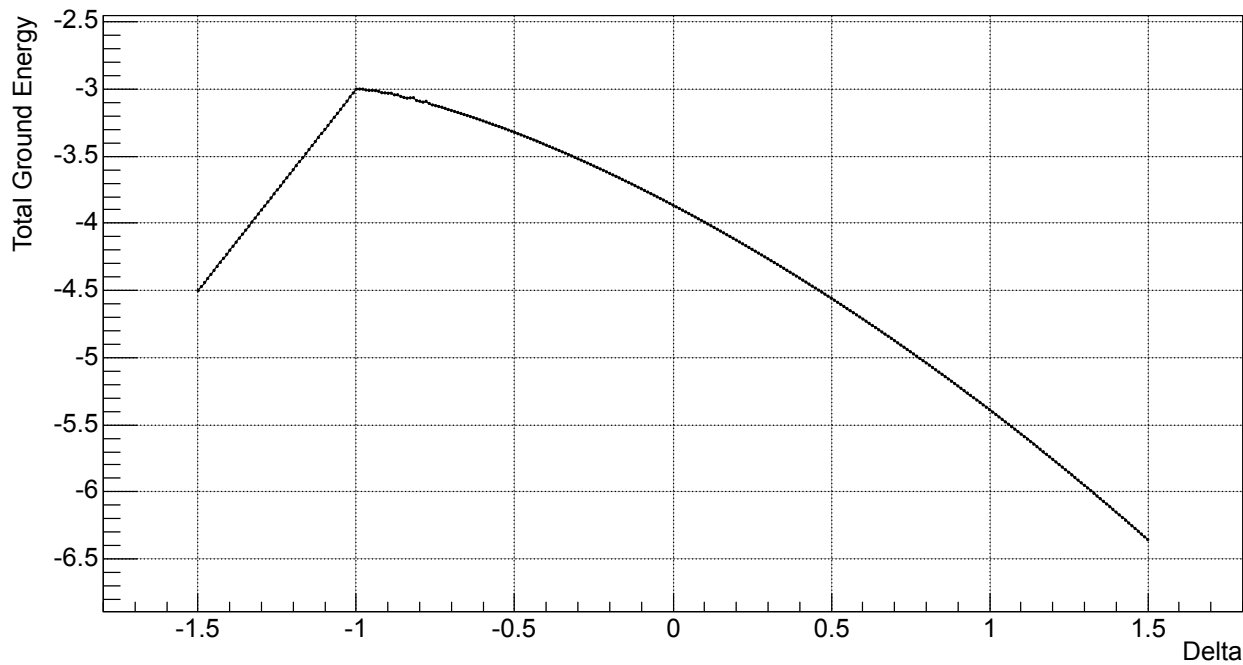
通过改变参数计算，得出了相变点为 $\Delta = -1.0$ 的结论。

参数：

- ☒ L 周期内的粒子数
- ☒ range Delta取值区间
- ☒ error 允许误差
- ☒ num 样本数

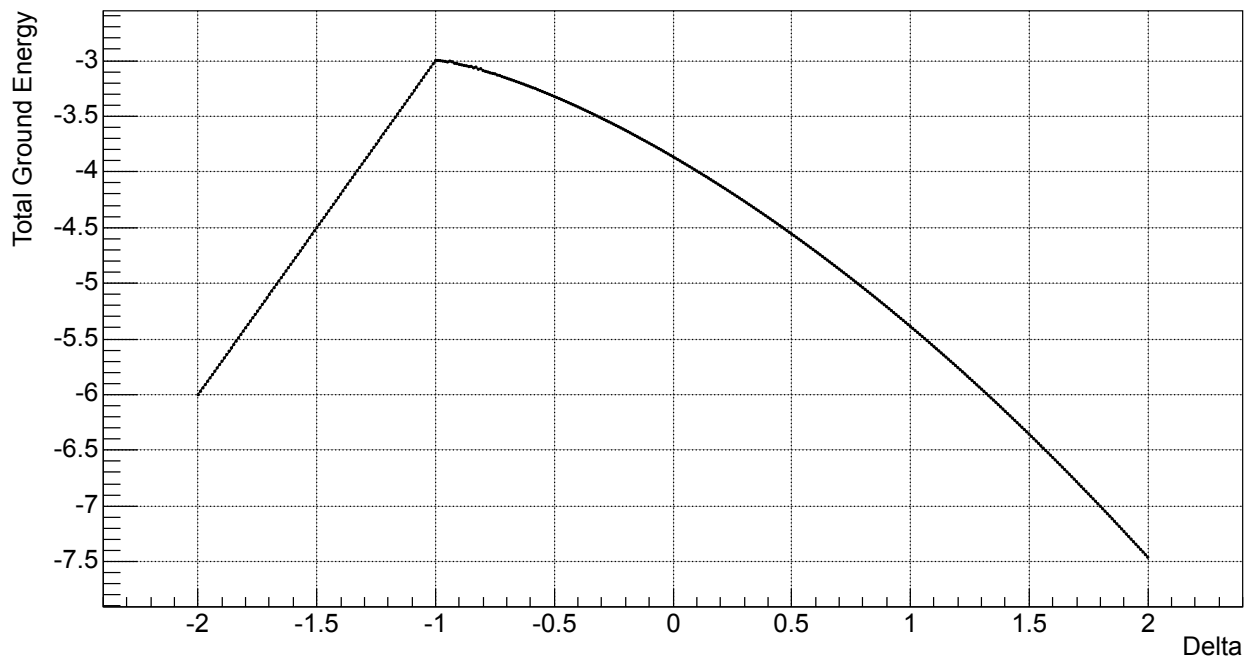
1. L = 12, range = [-1.5,1.5], error = 1e-13, num = 301

Lanczos alogrithm, num = 12, error = 1e-13



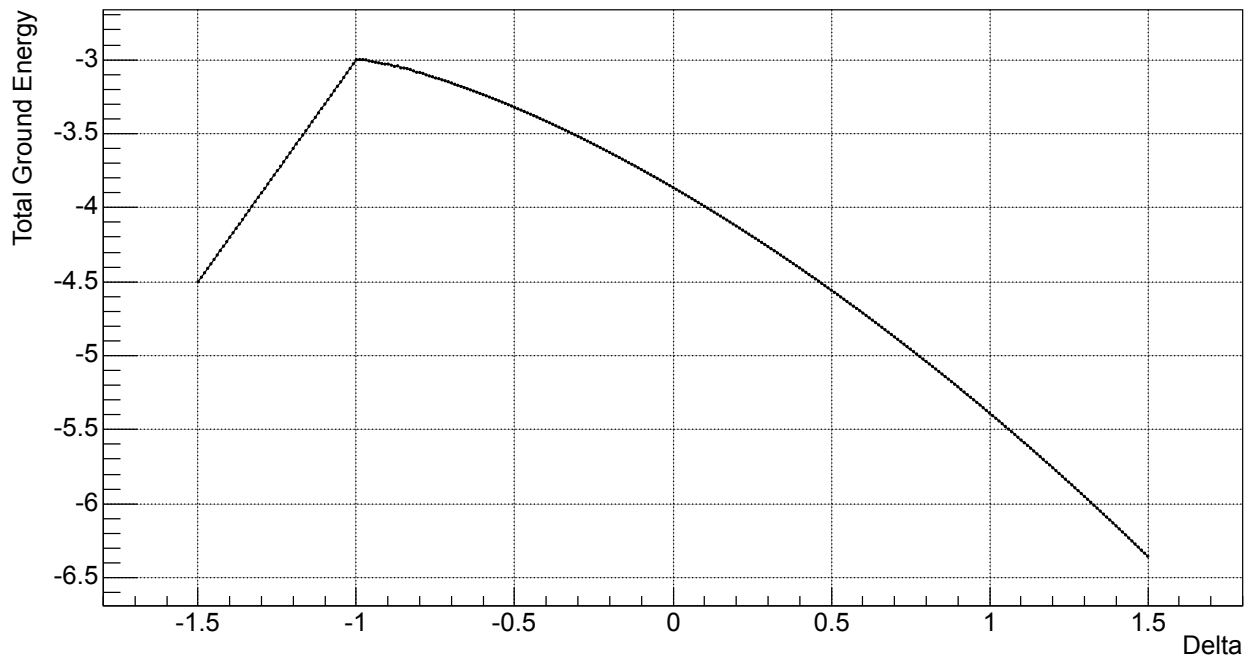
1. L = 12, range = [-2.0,2.0], error = 1e-13, num = 401

Lanczos algorithm, num = 12, error = 1e-13



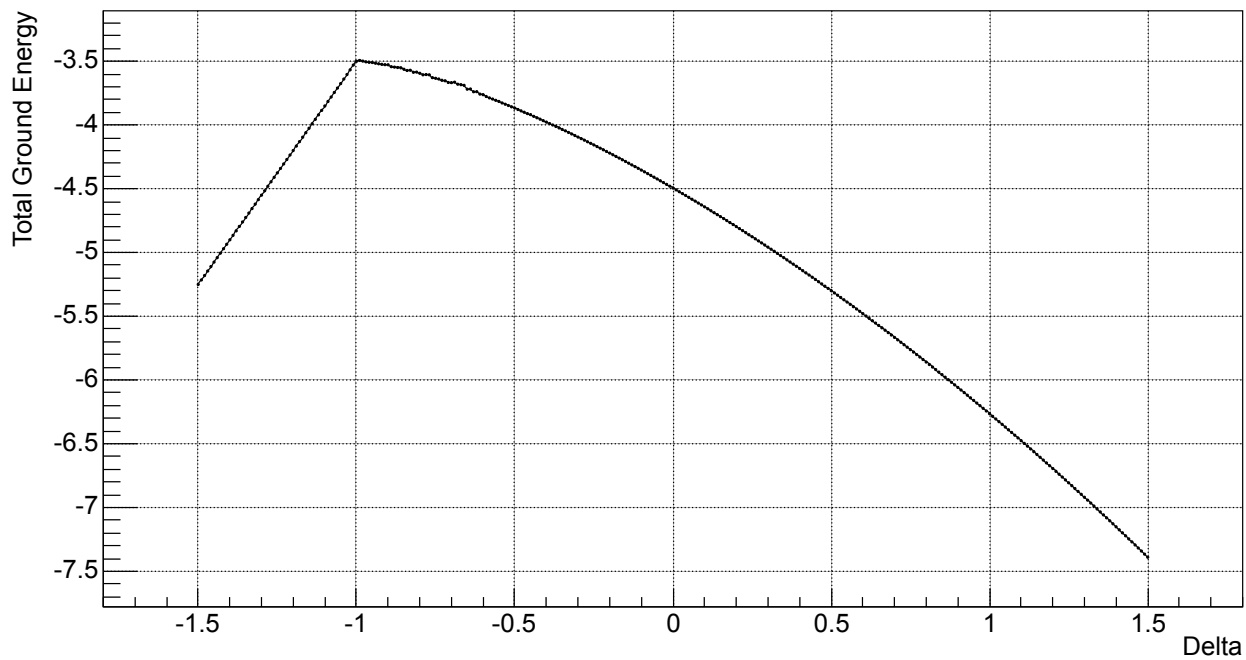
1. L = 12, range = [-1.5,1.5], error = 1e-15, num = 301

Lanczos algorithm, num = 12, error = 1e-15



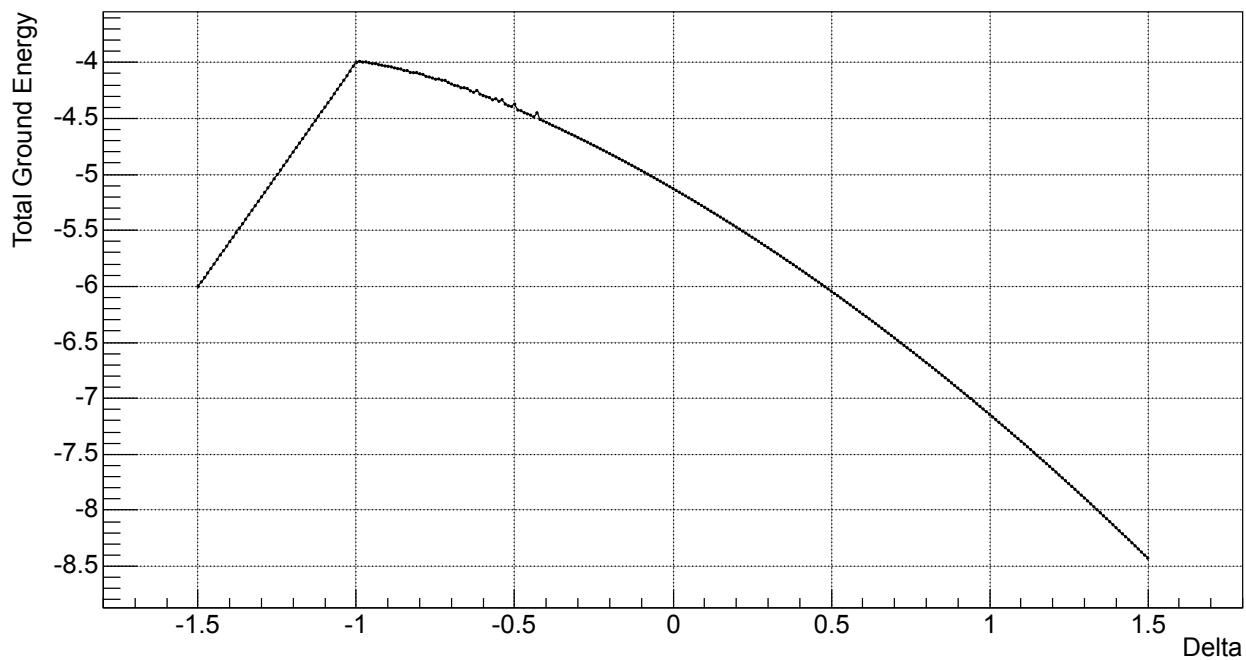
1. L = 14, range = [-1.5,1.5], error = 1e-13, num = 301

Lanczos algorithm, num = 14, error = 1e-13



1. L = 16, range = [-1.5,1.5], error = 1e-13, num = 301

Lanczos algorithm, num = 16, error = 1e-13



1. Golub, G. H., and Van Loan, C. F., *Matrix Computations*, 4th ed. (Johns Hopkins University Press, Baltimore, 2013). [↩](#) [↩](#)