量子物理计算方法选讲实验报告

李昊恩　2021010312

Exact Diagonalization, Task 2

目录

1　引言 **1**

2　代码实现 **2**

2.1　精确对角化 2

2.2　计算时间演化 2

3　结果 **4**

1　引言

本实验的目的是对一个单粒子高斯波包在梯度场一维 tight-binding 下的时间演化。体系的哈密顿量为：

$$H = -\sum_{j=1}^{N-1} (|j\rangle\langle j+1| + |j+1\rangle\langle j|) + F\sum_{j=1}^{N} j\,|j\rangle\langle j|\,. \tag{1}$$

设时间演化的初态为

$$|\psi(t=0)\rangle = \frac{1}{\sqrt{\Omega}}\sum_j e^{-(\alpha^2/2)(j-N_0)^2} e^{ik_0 j}\,|j\rangle\,. \tag{2}$$

为了计算时间演化，我们需要首先对哈密顿量进行精确对角化。该哈密顿量的形式比较简单，可以看出，若以 $\{j\}_{j=1}^N$ 为基，则 $|j\rangle\langle j+1|$ 和 $|j+1\rangle\langle j|$ 提供次对角线元素；而 $|j\rangle\langle j|$ 提供对角线元素，最终该哈密顿量在 $\{j\}_{j=1}^N$ 下的表示为一个 $N\times N$ 三对角稀疏矩阵。

一旦求出了对角化，我们得到哈密顿量的谱分解：

$$H = \sum_n E_n\,|\psi_n\rangle\langle\psi_n| \tag{3}$$

其中，E_n 表示第 n 个特征值，$|\psi_n\rangle$ 表示 E_n 对应的特征向量。此时，时间演化可以按下面的公式计算：

$$|\psi(t_0)\rangle = e^{-iHt}|\psi(0)\rangle = \sum_n e^{-iE_nt}|\psi_n\rangle\langle\psi_n|\psi(0)\rangle = \sum_n C_n e^{-iE_nt}|\psi_n\rangle = \sum_j\sum_n C_n e^{-iE_nt}\langle j|\psi_n\rangle|j\rangle. \tag{4}$$

式中，C_n 表示初态和第 n 个特征向量的重叠：

$$C_n = \langle \psi_n | \psi(0) \rangle. \tag{5}$$

2 代码实现

2.1 精确对角化

根据哈密顿量的形式，次对角线上应为 -1，而对角线上为 Fj，因此可以用如下代码构造矩阵，并进行精确对角化，找出最低的 10 个特征值：

```
# Define the Hamiltonian
H = np.zeros((N, N))

#Define the off-diagonal terms
for i in range(N-1):
    H[i, i+1] = -1
    H[i+1, i] = -1

#Define the diagonal terms
for i in range(N):
    H[i, i] = F * (i+1)

# Diagonalize the Hamiltonian
evals, evecs = np.linalg.eig(H)

# Find the 10 lowest eigenvalues
lowest_evals = np.sort(evals)[:10]

print("(1)_10_eigenvalues:{}".format(lowest_evals))
```

如果取 $(N, F) = (101, 0.4)$，则得到的输出为

```
(1)_10_eigenvalues:[-0.75734575  0.13035468  0.82547729  1.41163676  1.92122775
   2.37508492  2.79449719  3.1991657   3.59990942  3.99999256]
```

与讲义中的样例输出相同。

2.2 计算时间演化

对于时间演化，我们可以分步计算。先计算初态中的系数因子：

$$A_j = \frac{1}{\sqrt{\Omega}} e^{(-\alpha^2/2)(j-N_0)^2} e^{ik_0 j}. \tag{6}$$

我们在精确对角化时，得到的其实是 $|\psi_n\rangle$ 关于 $|j\rangle$ 的展开系数，记为：

$$\mathbf{\Psi}_n = ((\psi_n)_0, \cdots, (\psi_n)_{N-1}), \tag{7}$$

因此，C_n 可以按下式计算：

$$C_n = \langle\psi_n|\psi(0)\rangle = \mathbf{\Psi}_n^* \cdot \mathbf{A}, \quad 其中 \quad \mathbf{A} := (A_0, \cdots, A_{N-1}). \tag{8}$$

有了 C_n，我们就按照式 (4) 可以计算时间演化，求出 $|\psi(j,t)|^2$，具体代码实现为：

```python
#calculate the factors: (Alist = \exp^{(\alpha^2/2)(j-N_0)^2}e^{i_0kj},   Clist = <
    psi(0)|psi(j)>,)
Alist = [np.exp(-(alpha**2/2)* (j+1-N0)**2)*np.exp(1j*k0*(j+1)) for j in range(N)]
Alist = Alist/np.sqrt(np.sum(np.abs(Alist)**2))
Clist = [np.dot(np.conj(evecs[j]),Alist) for j in range(N)]

def time_evol(time):
    Elist = []
    for j in range(N):
        #Elist = e^{-iE_nt}<j|psi_n>
        Elist.append([np.exp(-1j*evals[n]*time)*evecs[n][j] for n in range(N)])
    prob_list = [np.abs(np.dot(Clist,Elist[j]))**2 for j in range(N)]
    return(prob_list)
prob_list_jlist = [time_evol(42)[j-1] for j in jlist]

print("(2)_Probability:{}".format(prob_list_jlist))
```

当输入为 $(N, F, k_0, \alpha, N_0, t_{\max}, t) = (101, 0.4, 0.618, 0.314, 33, 100, 42)$ 时，输出结果为：

```
(2)_Probability:[1.23026543116982e-08, 0.02239122543908873, 0.009280806433488786,
    2.1221076833967742e-12, 1.1244201998055816e-23]
```

与讲义的输出结果一致。

最后，对于密度图的绘制，我们只需要对 $j \in \{1, 2, \cdots, N\}$ 以及 $t \in \{1, 2, \cdots, t_{\max}\}$ 扫描，分别计算出 $|\psi(j,t)|^2$，再使用 matplotlib.pyplot 提供的 imshow 方法绘制即可，具体实现如下：

```python
#plot the density diagram
j_values = np.arange(0, N)
t_values = np.arange(0, tmax)
density = np.zeros((len(j_values), len(t_values)))
for j in tqdm(range(len(j_values))):
    for t in range(len(t_values)):
        prob_density = time_evol(t)[j-1]
        density[j, t] = prob_density
```

```
plt.imshow(density, cmap='hot', interpolation='nearest', origin='lower')
plt.xlabel('Time (t)')
plt.ylabel('Position (j)')
plt.title('Density plot of psi(j,t)')
plt.colorbar()
plt.show()
print("(3)_Plot: figure saved as 2021010312_0925_task2.png")
```

对于上述输入，输出结果如图1。

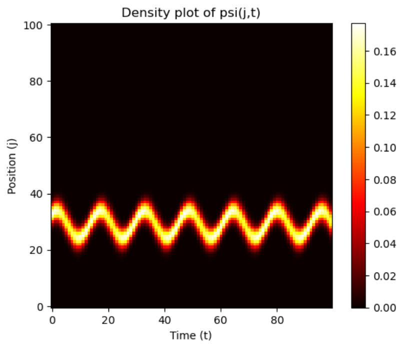

图 1: 样例输出

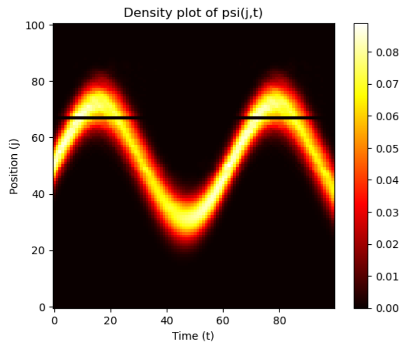

图 2: 输出

与讲义中的样例输出一致。

3 结果

设置参数为 $(N, F, k_0, \alpha, N_0, t_{\max}, t) = (101, 0.1, \frac{\pi}{2}, 0.15, 51, 100, 42)$，输出结果如下

```
=========Task 2:output=========
(1)_10_eigenvalues:[-1.50050978 -1.13263731 -0.83544214 -0.57592755 -0.34141332
 -0.12533996 0.07623401  0.26590362  0.44548515  0.61630546]
(2)_Probability:[9.253801835103358e-07, 0.003913502822921027, 0.05771569802792494,
 0.037273651473288025, 0.00029070016793455621]
(3)_Plot: figure saved as 2021010312_0925_task2.png
```

密度图见图2。