计算物理作业 6

杨远青　22300190015　CompPhys 24

2024 年 11 月 6 日

起视四境，而作业又至矣。

1　题目 1：一维 Kronig-Penney 模型的本征值求解

1.1　题目描述

One-dimensional Kronig-Penney problem. Considering the Hamiltonian of the system as

$$\hat{H} = -\frac{\hbar^2}{2m_e}\frac{\partial^2}{\partial x^2} + V(x)$$

with a one-dimensional periodic potential $V(x) = V(x+a)$. The potential can be expressed as

$$V(x) = \begin{cases} 0, & \text{if } 0 \le x < L_W, \\ U_0, & \text{if } L_W \le x < a \end{cases}$$

and the period of the potential is $a = L_W + L_B$, which is also shown in the Figure below.

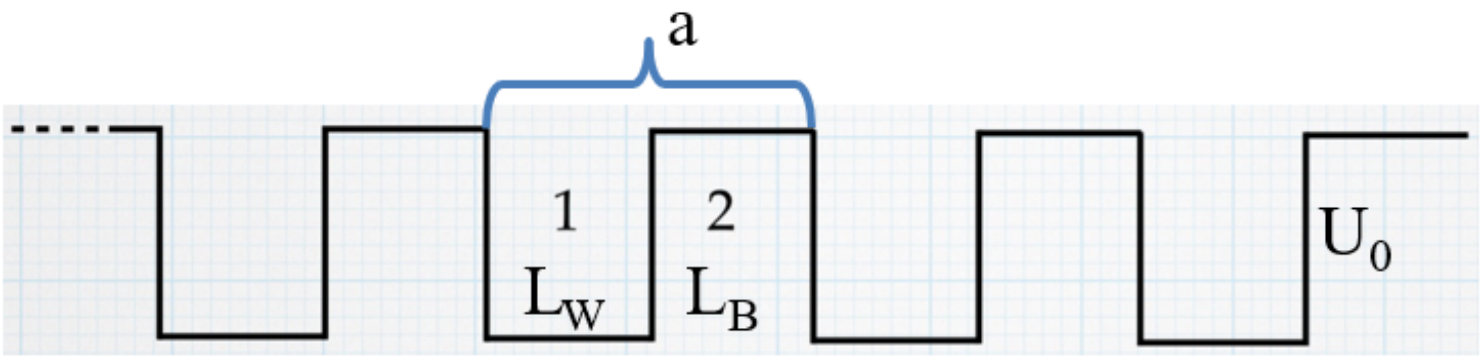


图 1: Kronig-Penney potential well

With parameters:

$$U_0 = 2\,\text{eV}, \quad L_W = 0.9\,\text{nm}, \quad L_B = 0.1\,\text{nm} \quad (a = 1\,\text{nm})$$

Using FFT, find the lowest three eigenvalues of the electric eigenstates that satisfy

$$\hat{H}\psi_i = E\psi_i \quad \text{and} \quad \psi_i(x) = \psi_i(x+a).$$

Explanation: Since the system is translation-invariant, i.e., $\psi_i(x) = \psi_i(x+a)$, we can use the plane wave basis expansion

$$\psi(x) = \frac{1}{\sqrt{a}} \sum_q C_q e^{iq\frac{2\pi}{a}x}, \quad q = -N, -N+1, \ldots, -1, 0, 1, \ldots, N-1, N.$$

In this basis set, the Hamiltonian can be represented in matrix form as

$$H_{pq} = \frac{1}{a}\langle e^{ip\frac{2\pi}{a}x}|\hat{H}|e^{iq\frac{2\pi}{a}x}\rangle_{\text{cell}} = \frac{1}{a}\int_0^a dx e^{-ip\frac{2\pi}{a}x}\hat{H}e^{iq\frac{2\pi}{a}x}.$$

To calculate $\hat{H}e^{iq\frac{2\pi}{a}x}$, the periodic potential $V(x)$ can be expanded in Fourier series as $V(x) \to V_q$, where

$$V(x) = \sum_{q'=-N}^{N} V_{q'} e^{iq'\frac{2\pi}{a}x}.$$

The basis wave function can then be written as:

$$\hat{H}e^{iq\frac{2\pi}{a}x} = (\hat{T}+\hat{V})e^{iq\frac{2\pi}{a}x} = \frac{2\hbar^2 q^2\pi^2}{m_e a^2}e^{iq\frac{2\pi}{a}x} + \sum_{q'=-N}^{N} V_{q'}e^{i(q'+q)\frac{2\pi}{a}x}$$

Try constructing the Hamiltonian matrix H_{pq} and solve the eigenvalue equation $\hat{H}\psi_i = E\psi_i$ to obtain the three lowest energy eigenvalues.

Special note: You can use built-in functions to simplify the eigenvalue calculations and FFT transformations.

1.2 程序描述

这道题的过程和结果都太神奇了。

我们先顺着题目未完成的思路进行下来，计算哈密顿矩阵元

$$H_{pq} = \frac{1}{a}\left\langle e^{ip\frac{2\pi}{a}x}\left|\hat{T}\right|e^{iq\frac{2\pi}{a}x}\right\rangle + \frac{1}{a}\left\langle e^{ip\frac{2\pi}{a}x}\left|\hat{V}\right|e^{iq\frac{2\pi}{a}x}\right\rangle$$

其中动能项是对角阵

$$\frac{1}{a}\left\langle e^{ip\frac{2\pi}{a}x}\left|\hat{T}\right|e^{iq\frac{2\pi}{a}x}\right\rangle = \frac{\hbar^2}{2m_e}\left(\frac{2\pi q}{a}\right)^2 \cdot \frac{1}{a}\int_0^a e^{-ip\frac{2\pi}{a}x}e^{iq\frac{2\pi}{a}x}dx = \frac{\hbar^2}{2m_e}\left(\frac{2\pi q}{a}\right)^2\delta_{pq} = \frac{2\hbar^2\pi^2q^2}{m_e a^2}\delta_{pq}$$

利用

$$\left\langle e^{ip\frac{2\pi}{a}x}\left|e^{iq'\frac{2\pi}{a}x}\right|e^{iq\frac{2\pi}{a}x}\right\rangle = \int_0^a e^{-ip\frac{2\pi}{a}x}e^{iq'\frac{2\pi}{a}x}e^{iq\frac{2\pi}{a}x}dx = \int_0^a e^{i(q'+q-p)\frac{2\pi}{a}x}dx = a\delta_{p-q,q'}$$

可以发现，势能矩阵元很自然地承载势能的傅里叶系数

$$\frac{1}{a}\left\langle e^{ip\frac{2\pi}{a}x}\left|V(x)\right|e^{iq\frac{2\pi}{a}x}\right\rangle = \sum_{q'=-N}^{N} V_{q'} \cdot \frac{1}{a}\left\langle e^{ip\frac{2\pi}{a}x}\left|e^{iq'\frac{2\pi}{a}x}\right|e^{iq\frac{2\pi}{a}x}\right\rangle = \sum_{q'=-N}^{N} V_{q'}\delta_{p-q,q'} = V_{p-q}$$

注意到我们对实数域的势能展开时 $V_{-q'} = V_{-q'}^*$，所以最后的哈密顿矩阵是厄米阵。

不过这里有个小漏洞，原本 $p-q$ 的取值应该在 $[-2N, 2N]$，但我们在傅里叶展开势能为 $V_{q'}$ 的时候只展开到了 $q' \in [-N, N]$，所以按题意需要将 $p-q$ 截断在 $p-q \in [-N, N]$ 之间（在赋值的时候加一组布尔掩码），这样才能保证 $\sum_{q'=-N}^{N} V_{q'}\delta_{p-q,q'} = V_{p-q}$.

我原本认为这样是不合理的，应该将势能展开到 $q' \in [-2N, 2N]$，而不是选择牺牲势能项的精度。但在助教提示下发现，本题的势能项本身影响就不大，所以这样的截断是可以接受的。在主程序中我也对这一点进行了对比论证，详见结果示例。

在最初阅读题干时，我最不明白的是，如此简单的阶梯函数已经可以解析计算出势能的傅里叶展开

$$V_{q'} = \begin{cases} \frac{U_0 L_B}{a}, & q = 0 \\ \frac{U_0}{iq'2\pi}\left(1 - e^{-iq'2\pi\frac{L_W}{a}}\right), & q \neq 0 \end{cases}$$

为何还需要进行 FFT 变换？随后想到这节课毕竟是在学习 FFT，故在主程序中加入了 FFT 变换（先将实空间离散化再传递给 `scipy.fft` 进行变换）的系数与解析计算的系数进行对比，发现两者差异在容许范围内，后面的计算则是基于解析计算的系数进行的。

源代码在 `src/KP_well.py` 中，请在该目录下运行`python -u KP_well.py` 即可得到结果，需要安装 `numpy` 与 `scipy` 库辅助运算，`matplotlib` 库绘图。结果非常神奇，在大规模的绘制与计算中，我们发现了疑似二重简并的现象。在与 QM1 荣誉课的助教老师讨论后，合理怀疑是有限 N 的截断带来的边界效应，且在高能级时非局域化更明显，两侧的对称传播似乎是合理的。可惜在 `debug` 时我尝试过绘制其它相邻能级的波函数概率分布，未发现明显对称性。二重简并的原因有待进一步探讨，但能级的 n^2 增长是可以确认的。

主程序流程描述

1. **初始化物理常量与参数**：从 `const` 获取需要使用的物理常数（如 $\hbar$、m_e 等），定义参数，如 U_0、L_w 和 L_b。
2. **傅里叶系数计算对比**：均采用自然顺序 $[0, 1, 2, \ldots, N, -N, \ldots, -1]$，便于切片负数索引

- 使用 `compute_Vq_analytical` 方法计算势能傅里叶系数，基于解析表达式，
- 使用 `compute_Vq_fft` 方法，通过 FFT 计算傅里叶系数，验证解析方法的精确性。
- 比较两种方法的结果，输出最大和平均差异。

3. **小规模收敛性检查**：借助 `check_convergence` 函数，在较小 N 下进行能级收敛性测试。
4. **AB 方法对比**：

- **方法 A**：在傅里叶系数范围 $Vq' \in [-N, N]$ 下进行求解，得到小规模能级，
- **方法 B**：在扩展傅里叶系数范围 $Vq' \in [-2N, 2N]$ 下进行求解，
- 比较两种方法的能级差异，在之后计算中遵循题意采用方法 A.

5. **中等规模 $N = 100$ 计算与绘制**：

- 使用解析方法计算的傅里叶系数，构建哈密顿量矩阵，通过 `solve_eigenvalues` 方法得到能级和波函数，
- 使用 `reconstruct_wavefunction_vectorized` 重构波函数，充分考虑基底正交性，归一化操作不需要积分，
- 通过 `plot_wavefunctions_and_potential` 绘制势阱和前三个波函数，
- 绘制能级图，发现在小规模 N 中未出现的二级简并现象。

6. **大规模计算与拟合**（可选）：在用户确认的情况下

- 使用更大 $N = 500$ 计算较多能级，
- 通过 `curve_fit` 函数拟合能级间距与量子数的平方增长关系，
- 输出拟合系数并绘制拟合结果。

1.3 伪代码

Powered by LaTeX pseudocode generator

Algorithm 1: Main Process Flow

Input: U_0, L_w, L_b, $N_{initial}$, $num\_levels_{initial}$, $medium\_N$, $medium\_num\_levels$, $large\_N$, $large\_num\_levels$

Output: Energy levels and wave functions, potential well plots, convergence analysis

Step 1: Fourier Coefficients Comparison;
$Vq\_analytical, Vq\_fft \leftarrow$ `compare_fourier_methods`$(N_{initial}, U_0, L_w, L_b, a)$;
if `compare_fourier_methods` *accuracy is high* **then**
    Proceed to next step;
end
Step 2: Convergence Check;
`check_convergence`$(U_0, L_w, L_b, a, hbar, m\_e)$;
Step 3: Energy Levels Calculation with Method A and B;
for *Method in [A, B]* **do**
    $eigenvalues, eigenvectors \leftarrow$ `build_and_solve`(*Method parameters*);
    if *degeneracies found* **then**
        Print "Degeneracies exist";
    end
    else
        Print "No degeneracies found";
    end
end
Step 4: Plotting for Medium N;
$eigenvalues\_medium, eigenvectors\_medium \leftarrow$ `build_and_solve`$(medium\_N, medium\_num\_levels)$;
$x\_medium, wavefunctions\_medium \leftarrow$ `reconstruct_wavefunction_vectorized`$(eigenvectors\_medium)$;
$V\_x\_medium \leftarrow$ `generate_potential_vectorized`(x_medium);
`plot_wavefunctions_and_potential`$(x\_medium, wavefunctions\_medium, V\_x\_medium)$;
`plot_energy_levels`$(eigenvalues\_medium, medium\_N)$;
if *Proceed to large scale* **then**
    Repeat plotting with $large\_N$ and fit quadratic relation;
end

Algorithm 2: Compute Analytical Fourier Coefficients

Input: N, U_0, L_w, L_b, a

Output: Fourier coefficients Vq

$n\_list \leftarrow$ `get_n_list(`N`)`;

for *Each non-zero* n *in* n_list **do**

  Compute $Vq[n]$ based on analytical formula;

end

$Vq[0] \leftarrow U_0 \cdot L_b/a$;

return Vq

Algorithm 3: Build Kinetic Matrix

Input: q_values, a, $hbar$, m_e

Output: Kinetic matrix T

for *Each* q *in* q_values **do**

  $T[q,q] \leftarrow \frac{\hbar^2 \cdot (2\pi q/a)^2}{2m_e}$;

end

return T *as a sparse diagonal matrix*

Algorithm 4: Solve for Eigenvalues with Error Handling

Input: Hamiltonian matrix H, desired levels num_levels

Output: Eigenvalues and eigenvectors, or an error message if solving fails

if `eigsh` *can solve for eigenvalues* **then**

  (eigenvalues, eigenvectors) $\leftarrow$ `eigsh(`$H, k = num\_levels, which =$ "SA"`)`;

  Sort eigenvalues and eigenvectors by eigenvalue magnitude;

  return eigenvalues, eigenvectors;

end

else

  Print "Error in eigenvalue solver";

  return $None, None$;

end

Algorithm 5: Reconstruct Wavefunction

Input: Eigenvectors, q_values, a

Output: Wavefunctions over extended domain

for *Each eigenvector* v **do**

  for *Each position* x *in domain* **do**

    $psi(x) \leftarrow v \cdot$ `exp`$(2\pi i \cdot q \cdot x/a)$;

    Normalize psi over extended domain;

  end

end

1.4 结果示例

```
第一步：比较解析方法和FFT方法计算傅里叶系数的差异

傅里叶系数计算方法比较：
傅里叶级数范围：[-50, 50]
解析方法用时：0.00e+00秒
FFT方法用时：5.12e-04秒
最大绝对差异：2.058607e-02 eV
平均绝对差异：1.223039e-02 eV
解析方法和FFT方法计算的傅里叶系数差异非常小。
```

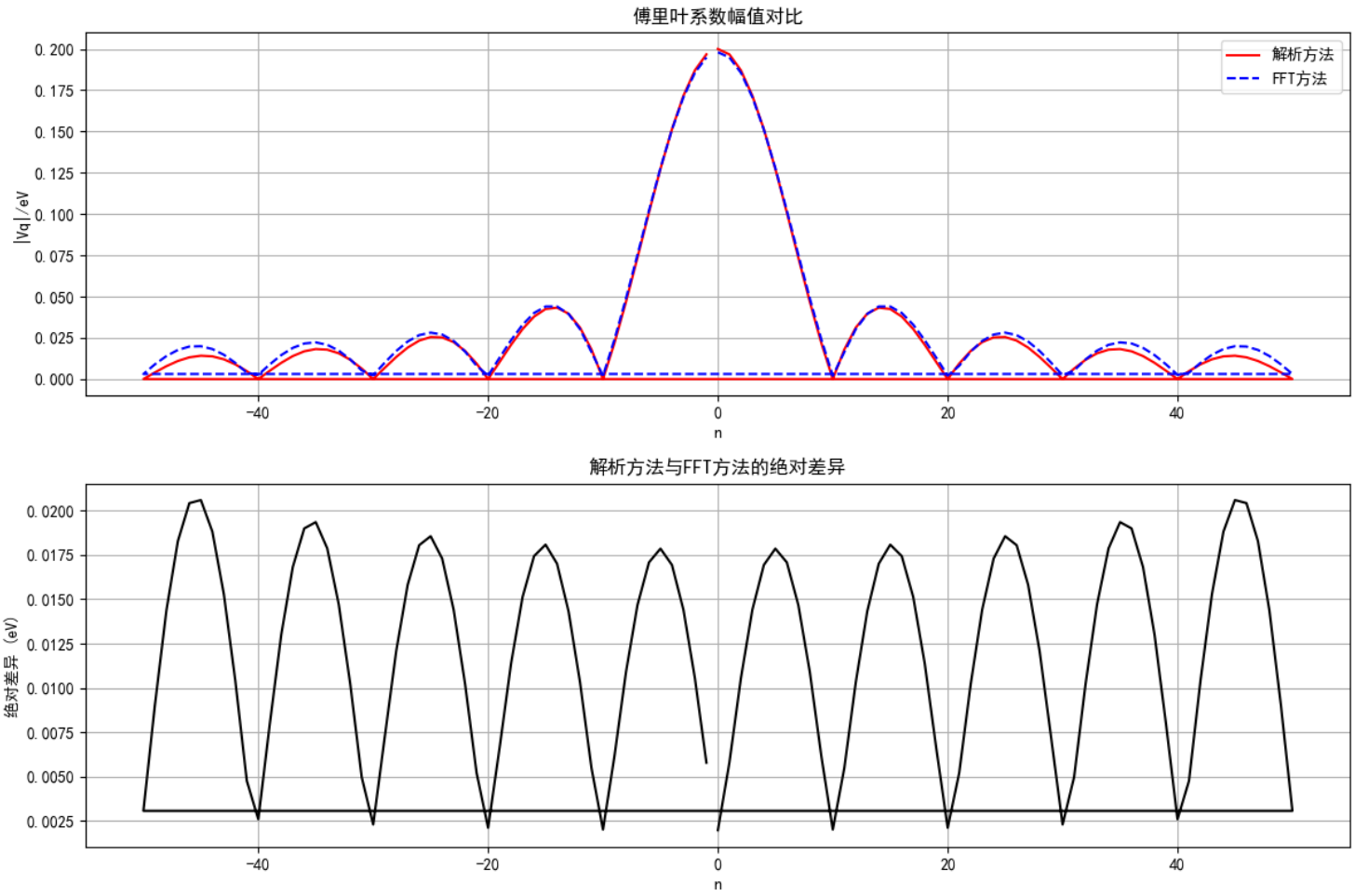


图 2: 第一步：比较解析计算与 FFT 得到的势能傅里叶系数

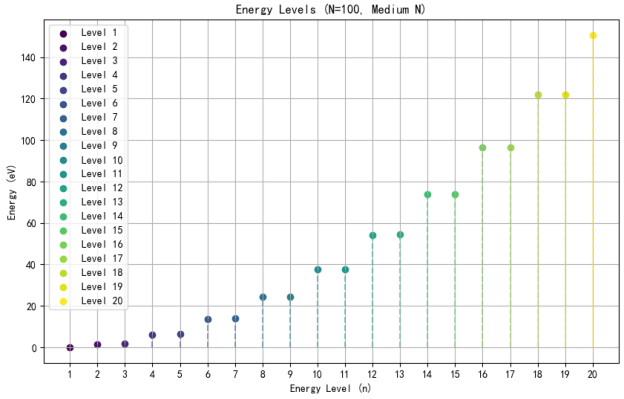
```
第二步：进行收敛性检查（仅对小规模N进行）
N=10, Lowest 3 eigenvalues (eV): [0.14464301 1.51649618 1.88295936]
No degeneracies found.
N=20, Lowest 3 eigenvalues (eV): [0.1446063  1.51647998 1.88283855]
No degeneracies found.
N=30, Lowest 3 eigenvalues (eV): [0.14460216 1.51647846 1.88282499]
No degeneracies found.
Converged at N=30
```

图 3: 第二步：收敛性检查

```
第三步：使用解析方法计算的傅里叶系数进行AB方法对比

方法A（Vq' ∈[-50, 50]）
Lowest 3 eigenvalues (eV):
  Level 1: 1.4460e-01 eV
  Level 2: 1.5165e+00 eV
  Level 3: 1.8828e+00 eV
No degeneracies found.

方法B（Vq' ∈[-100, 100]）
Lowest 3 eigenvalues (eV):
  Level 1: 1.4460e-01 eV
  Level 2: 1.5165e+00 eV
  Level 3: 1.8828e+00 eV
No degeneracies found.

方法A和方法B的前3个能级相对差异：[2.39109506e-06 7.59823372e-08 6.01067236e-07]

方法A和方法B的能级差异非常小。
```

图 4: 第三步：不同势能展开范围的 AB 方法对比

```
第四步：使用中等规模N=100进行能级绘制
N=100, Lowest 20 eigenvalues (eV):
  Level 1: 1.4460e-01 eV
  Level 2: 1.5165e+00 eV
  Level 3: 1.8828e+00 eV
  Level 4: 6.0633e+00 eV
  Level 5: 6.3848e+00 eV
  Level 6: 1.3633e+01 eV
  Level 7: 1.3855e+01 eV
  Level 8: 2.4216e+01 eV
  Level 9: 2.4325e+01 eV
  Level 10: 3.7801e+01 eV
  Level 11: 3.7810e+01 eV
  Level 12: 5.4321e+01 eV
  Level 13: 5.4379e+01 eV
  Level 14: 7.3860e+01 eV
  Level 15: 7.3946e+01 eV
  Level 16: 9.6426e+01 eV
  Level 17: 9.6503e+01 eV
  Level 18: 1.2201e+02 eV
  Level 19: 1.2206e+02 eV
  Level 20: 1.5061e+02 eV
Degeneracies found.
```



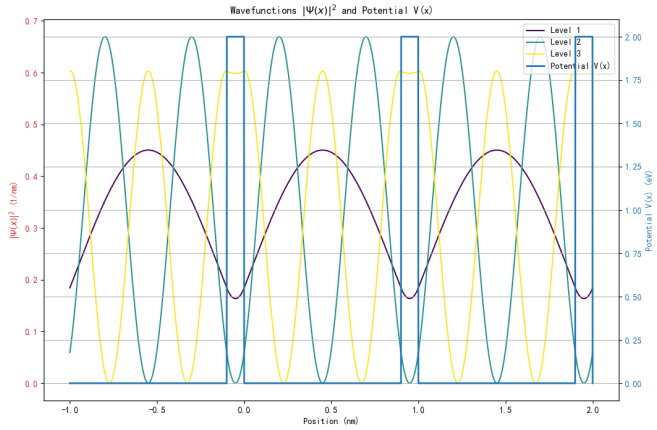


图 5: 第四步：进行中等规模 $N = 100, num\_levels = 20$ 的计算与绘制，并绘制势阱与前三个能级的波函数

```
是否进行大规模N=500的能级绘制与拟合？（y/n）：y

继续进行大规模N的能级绘制与拟合
N=500, Lowest 100 eigenvalues (eV):
  Level 1: 1.4460e-01 eV
  Level 2: 1.5165e+00 eV
  Level 3: 1.8828e+00 eV
  Level 4: 6.0633e+00 eV
  Level 5: 6.3848e+00 eV
  Level 6: 1.3633e+01 eV
  Level 7: 1.3855e+01 eV
  Level 8: 2.4216e+01 eV
  Level 9: 2.4325e+01 eV
  Level 10: 3.7801e+01 eV
  Level 11: 3.7810e+01 eV
  Level 12: 5.4321e+01 eV
  Level 13: 5.4379e+01 eV
  Level 14: 7.3860e+01 eV
  Level 15: 7.3946e+01 eV
  Level 16: 9.6426e+01 eV
  Level 17: 9.6503e+01 eV
  Level 18: 1.2201e+02 eV
  Level 19: 1.2206e+02 eV
  Level 20: 1.5061e+02 eV
  Level 21: 1.5061e+02 eV
  Level 22: 1.8218e+02 eV
  Level 23: 1.8222e+02 eV
  Level 24: 2.1677e+02 eV
  Level 25: 2.1682e+02 eV
  Level 26: 2.5437e+02 eV
  Level 27: 2.5442e+02 eV
  Level 28: 2.9499e+02 eV
  Level 29: 2.9502e+02 eV
  Level 30: 3.3863e+02 eV
  Level 31: 3.3863e+02 eV
  Level 32: 3.8524e+02 eV
  Level 33: 3.8527e+02 eV
  Level 34: 4.3487e+02 eV
  Level 34: 4.3487e+02 eV
  Level 35: 4.3491e+02 eV
  Level 36: 4.8752e+02 eV
  Level 37: 4.8755e+02 eV
  Level 38: 5.4318e+02 eV
  Level 39: 5.4320e+02 eV
  Level 40: 6.0185e+02 eV
  Level 41: 6.0185e+02 eV
  Level 42: 6.6351e+02 eV
  Level 43: 6.6353e+02 eV
  Level 44: 7.2818e+02 eV
  Level 45: 7.2821e+02 eV
  Level 46: 7.9587e+02 eV
  Level 47: 7.9589e+02 eV
  Level 48: 8.6657e+02 eV
  Level 49: 8.6658e+02 eV
  Level 50: 9.4028e+02 eV
  Level 51: 9.4028e+02 eV
  Level 52: 1.0170e+03 eV
  Level 53: 1.0170e+03 eV
  Level 54: 1.0967e+03 eV
  Level 55: 1.0967e+03 eV
  Level 56: 1.1794e+03 eV
  Level 57: 1.1794e+03 eV
  Level 58: 1.2652e+03 eV
  Level 59: 1.2652e+03 eV
  Level 60: 1.3539e+03 eV
  Level 61: 1.3539e+03 eV
  Level 62: 1.4457e+03 eV
  Level 63: 1.4457e+03 eV
  Level 64: 1.5404e+03 eV
  Level 65: 1.5404e+03 eV
  Level 66: 1.6382e+03 eV
  Level 69: 1.7390e+03 eV
  Level 70: 1.8427e+03 eV
  Level 71: 1.8427e+03 eV
  Level 72: 1.9495e+03 eV
  Level 73: 1.9495e+03 eV
  Level 74: 2.0593e+03 eV
  Level 65: 1.5404e+03 eV
  Level 66: 1.6382e+03 eV
  Level 67: 1.6382e+03 eV
  Level 68: 1.7390e+03 eV
  Level 69: 1.7390e+03 eV
  Level 70: 1.8427e+03 eV
  Level 71: 1.8427e+03 eV
  Level 72: 1.9495e+03 eV
  Level 73: 1.9495e+03 eV
  Level 74: 2.0593e+03 eV
  Level 75: 2.0593e+03 eV
  Level 76: 2.1721e+03 eV
  Level 77: 2.1722e+03 eV
  Level 78: 2.2880e+03 eV
  Level 79: 2.2880e+03 eV
  Level 80: 2.4068e+03 eV
  Level 81: 2.4068e+03 eV
  Level 82: 2.5286e+03 eV
  Level 83: 2.5286e+03 eV
  Level 84: 2.6535e+03 eV
  Level 85: 2.6535e+03 eV
  Level 86: 2.7813e+03 eV
  Level 87: 2.7813e+03 eV
  Level 88: 2.9122e+03 eV
  Level 89: 2.9122e+03 eV
  Level 90: 3.0460e+03 eV
  Level 91: 3.0460e+03 eV
  Level 92: 3.1829e+03 eV
  Level 93: 3.1829e+03 eV
  Level 94: 3.3228e+03 eV
  Level 95: 3.3228e+03 eV
  Level 96: 3.4657e+03 eV
  Level 97: 3.4657e+03 eV
  Level 98: 3.6116e+03 eV
  Level 99: 3.6116e+03 eV
  Level 100: 3.7605e+03 eV
Degeneracies found.

拟合结果：k = 0.371510 eV
```

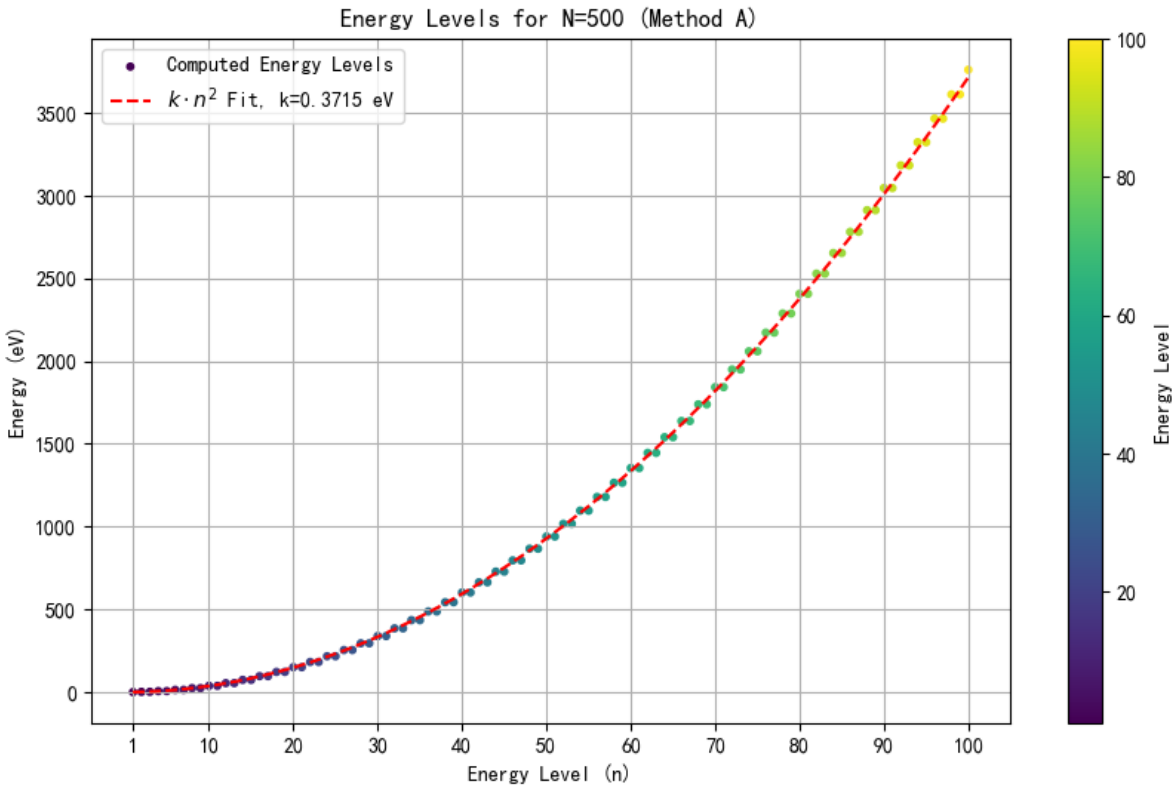


图 6: 第五步：进行大规模 $N = 500, num\_levels = 100$ 的计算与绘制，并进行拟合

2 题目 2：太阳黑子周期性检测

2.1 题目描述

Use the file called `sunspots.txt`, which contains the observed number of sunspots on the Sun for each month since January 1749. Write a program to calculate the Fourier transform of the sunspot data and then make a graph of the magnitude squared $|c_k|^2$ of the Fourier coefficients as a function of k—also called the power spectrum of the sunspot signal. You should see that there is a noticeable peak in the power spectrum at a nonzero value of k. Find the approximate value of k to which the peak corresponds. What is the period of the sine wave with this value of k?
Special note: You may use any built-in functions for the Fourier transform.

2.2 程序描述

使用 `scipy` 库的 `detrend` 函数去除线性趋势，也进行了使用 Hanning 窗平滑过渡原信号

$$w(n) = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right)$$

的效果对比，该操作有效避免边缘频率泄露，最后使用 `np.fft` 进行快速傅里叶变换，得到频谱图并使用 `np.fft.ifft` 进行信号重构的对比。最终峰值频率对应周期为 $130.6m \approx 10.9a$，与太阳黑子爆发周期实验观测值相符。

源代码在 `src/analysis.py` 中，请在该目录下运行`python -u analysis.py` 即可得到结果，需要安装 `numpy` 与 `scipy` 库辅助运算，`matplotlib` 库绘图，其他平台运行可能需要调整字体管理代码。

2.3 结果示例

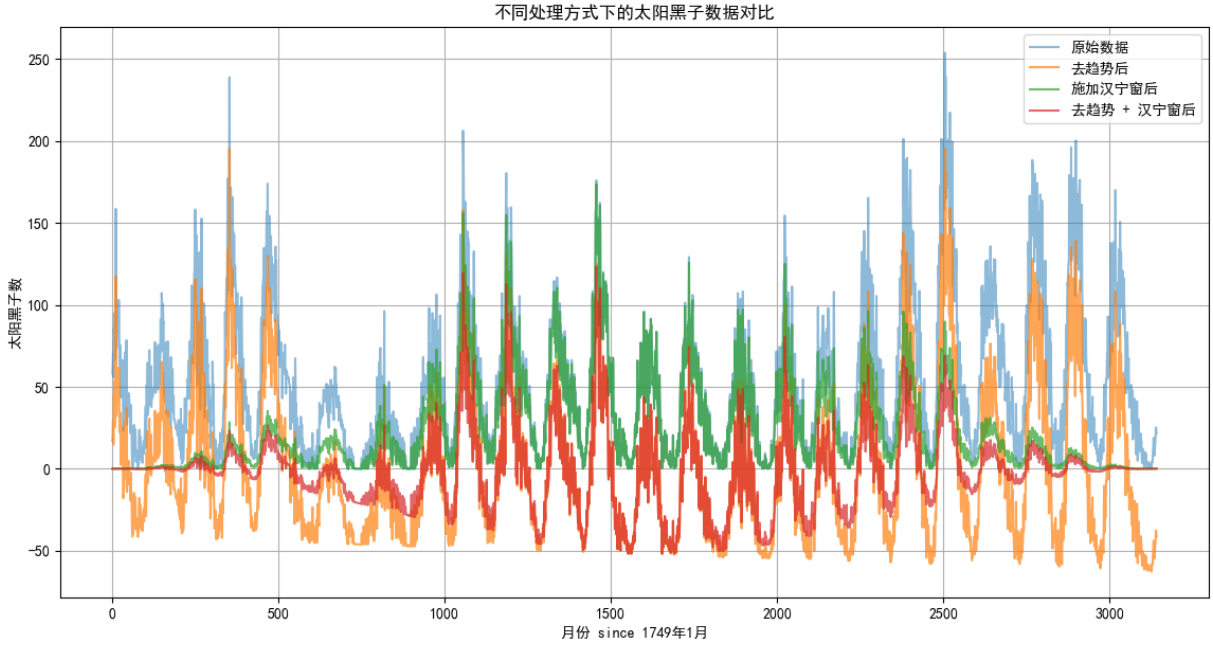


图 7: 不同方式处理的数据对比

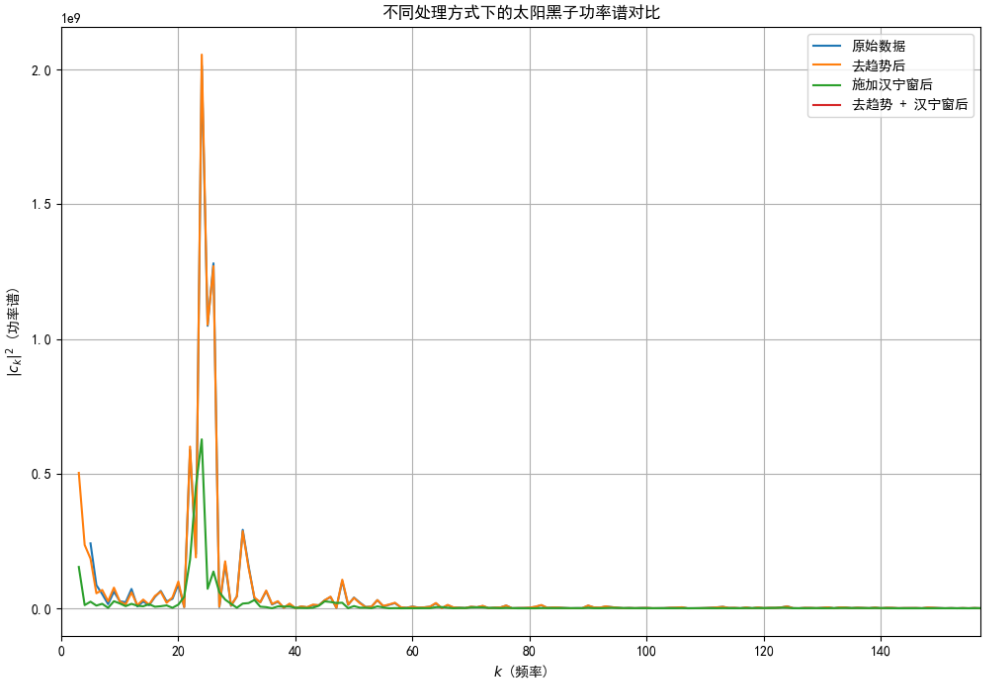


图 8: 去趋势 + 汉宁窗处理后的功率谱

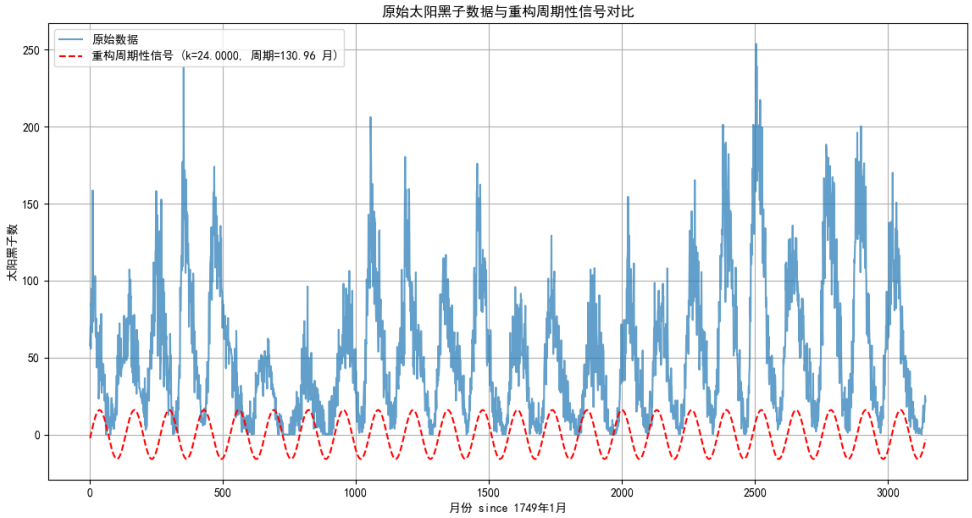


图 9: 使用峰值频率重构与原信号对比