

第三次作业

尹朝阳 物理学系

1 题目 1

1.1 题目描述

Prove that the time complexity of the Gaussian elimination algorithm is $O(N^3)$.

1.2 证明过程

高斯消去法的步骤如下（假设是一个 $n \times n$ 的矩阵）：

(0. 通过换行操作，使得第一行主元不为 0。该步骤需要 1 步操作)

1. 将第 1 行的所有 n 个元素乘上系数 $C_{1i}, 2 \leq i \leq n$ (其中系数满足 $a_{i1} - C_{1i} * a_{11} = 0$) 后，用第 2 行到第 n 行减去第 1 行操作过的所有 n 个元素。此时第 1 个元素 $a_{i1}, 2 \leq i \leq n$ 均变为 0。

第 1 行的每个元素与其同列的 $n - 1$ 的元素进行运算，需要 $n - 1$ 步。第 1 行有 n 个元素。则整个步骤 1 的操作数为 $n(n - 1)$ 。

2. 对于第 2 行，除第一个元素（已经为 0）外的其他元素乘上系数 $C_{2i}, 3 \leq i \leq n$ (其中系数满足 $a_{i2} - C_{2i} * a_{22} = 0$) 后，用第 3 行到第 n 行减去第 2 行操作过的所有 $n - 1$ 个元素。此时第 2 个元素 $a_{i2}, 3 \leq i \leq n$ 均变为 0。

第 2 行的每个元素与其同列的 $n - 2$ 的元素进行运算，需要 $n - 2$ 步。第 2 行有 $n - 1$ 个元素。则整个步骤 2 的操作数为 $(n - 1)(n - 2)$ 。

3. 仿照步骤 1、2，对第 3 到 n 行进行相同操作。其中，对于第 k ($3 \leq k \leq n$) 行，将第 k 列 a_{kk} 以下的元素全部变为 0 的操作数为 $(n - k + 1)(n - k)$ 。

总共对 n 行进行操作，总操作数为

$$\begin{aligned} \text{sum} &= \sum_{k=1}^n (n - k + 1)(n - k) \\ &= \sum_{k=1}^n (n - k + 1)^2 - \sum_{k=1}^n (n - k + 1) \\ &= \frac{n(n + \frac{1}{2})(n + 1)}{3} - \frac{n(n + 1)}{2} \\ &\sim n^3 \end{aligned}$$

在 n 很大时，第 0 步骤的 1 步操作对时间复杂度没有影响。

所以高斯消去法的整体时间复杂度为 $O(n^3)$ 。

注：有关高斯消去法的内容，可以参看 `/RREF/transform_matrix_into_RREF.py` 中的 `Gauss_Elimination` 函数，可以发现，最影响时间复杂度的为 3 个嵌套的 `for` 循环，这也可以说明高斯消去法的时间复杂度为 $O(n^3)$ 。

2 题目 2

2.1 题目描述

Write a general code to transform a $n*m$ matrix into the REDUCED ROW ECHELON FORM, and use the code to obtain the RREF of the following matrix.

$$\begin{bmatrix} 2 & 8 & 4 & 2 \\ 2 & 5 & 1 & 5 \\ 4 & 10 & -1 & 1 \end{bmatrix}$$

2.2 程序描述

题目要求将任意矩阵化为简化行阶梯形 (RREF)，即最终的矩阵形式要求：

- 非零行处于全零行的上方
- 非零行的主元为 1
- 非零行主元列位置先于该行下方的非零行的主元
- 任意一行主元所在的列，除主元外其他元素为 0

本程序的实现思路分为两部分：首先进行高斯消去法，将任意给定的矩阵化简为行阶梯形，再从行阶梯形出发，化简为唯一确定的简化行阶梯形。

其中，高斯消去法时，每一行操作的主元需要为非零数。若为 0，需要和下方的同列有非零数的行进行行交换。为减小浮点数导致的精度误差，行交换时，最好选取较大的数作为主元。

本程序的源代码为 `/RREF/transform_matrix_into_RREF.py`，其中 `Gauss_Elimination()` 函数实现高斯消去法，`RREF()` 函数实现将高斯消去法得到的行阶梯形变为简化行阶梯形。导入的第三方库有 `numpy`。

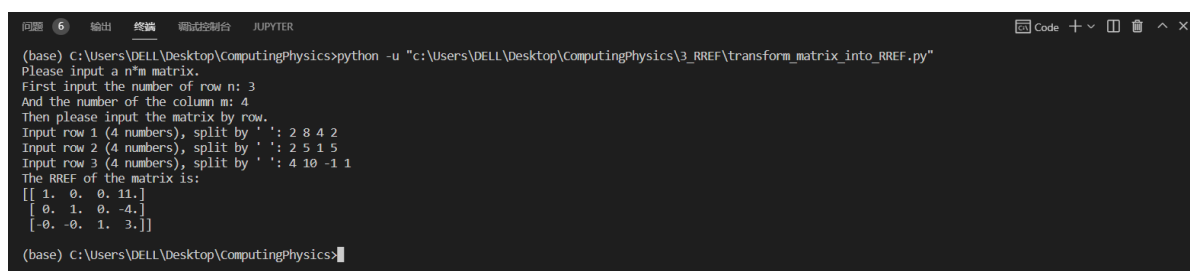
2.3 伪代码

Algorithm 2 transform matrix into RREF

Input : a $n * m$ matrix**Output :** the RREF of the given matrix: matrix_rref

```
1: function GAUSS_ELIMINATION(matrix)
2:   for  $i \leftarrow 1$  to  $n$  do ▷ prevent the pivot from being 0
3:      $m \leftarrow$  index of max(from  $matrix[i][i]$  to  $matrix[n][i]$ )
4:     exchange row  $m$  and row  $i$ 
5:   end for
6:   for  $i \leftarrow 1$  to  $n$  do
7:     for  $k \leftarrow i + 1$  to  $n$  do
8:        $R_k \leftarrow R_k - R_i * \frac{R_{ki}}{R_{ii}}$  ▷ turn all the lower entries in that column into 0
9:     end for
10:  end for
11: end function
12: function RREF(matrix)
13:  matrix  $\leftarrow$  GAUSS_ELIMINATION(matrix) ▷ transform the matrix into a row echelon form
14:  for  $i \leftarrow 1$  to  $n$  do
15:     $R_i \leftarrow R_i / R_{ii}$  ▷ turn the pivots into 1
16:    for  $j \leftarrow 1$  to  $i - 1$  do
17:       $R_j \leftarrow R_j - R_j * \frac{R_{ji}}{R_{ii}}$  ▷ turn all the other entries in that column into 0
18:    end for
19:  end for
20: end function
21: matrix_rref  $\leftarrow$  RREF(matrix)
```

2.4 测试用例



```
(base) C:\Users\DELL\Desktop\ComputingPhysics>python -u "c:\Users\DELL\Desktop\ComputingPhysics\3_RREF\transform_matrix_into_RREF.py"
Please input a n*m matrix.
First input the number of row n: 3
And the number of the column m: 4
Then please input the matrix by row.
Input row 1 (4 numbers), split by ' ': 2 8 4 2
Input row 2 (4 numbers), split by ' ': 2 5 1 5
Input row 3 (4 numbers), split by ' ': 4 10 -1 1
The RREF of the matrix is:
[[ 1.  0.  0. 11.]
 [ 0.  1.  0. -4.]
 [-0. -0.  1.  3.]]
(base) C:\Users\DELL\Desktop\ComputingPhysics>
```

图 1: RREF

3 题目 3

3.1 题目描述

Solve the 1D Schrodinger equation with the potential (i) $V(x) = x^2$; (ii) $V(x) = x^4 - x^2$ with the variational approach using a Gaussian basis (either fixed widths or fixed centers).

Consider the three lowest energy eigenstates.

3.2 程序描述

题目要求以高斯波包 $\psi(x) = \sqrt{\frac{v}{\pi}}e^{-v(x-s)^2}$ 为一组基，通过矩阵展开的形式变分法求解 1 维薛定谔方程，即得到其能量本征态以及相应的波函数解，这里我们只考虑画出三个能量最低的本征态的波函数解。

高斯波包有两个超参数：波包中心位置 s 和波包展宽 v ，为求解方便我们在构造基底的时候，只选择一个超参数进行变化，形成一组基，而事先固定另一个超参数的值。

求解思路为：对于一组基 $\psi(x)$ ，哈密顿矩阵满足 $H_{ij} = \int_{-\infty}^{\infty} \psi_i^*(x) \hat{H} \psi_j(x) dx$ ，其中 \hat{H} 为哈密顿算符，在一维定态薛定谔方程中 $\hat{H} = -\frac{\hbar^2}{2m} \nabla^2 + V(x)$ 。则能量满足矩阵方程 $HC = EC$ ，其中 E 为能量本征值， C 为求解的本征向量，则波函数解 $\Psi(x) = \sum_{k=1}^n c_k \psi_k(x)$ ，其中 c_k 为本征列向量 C 的第 k 个元素。

但是对于高斯波包，其张成的基中各基底显然是非正交的，对于求特征值、特征向量，需要先引入修正矩阵 S ，满足 $S_{ij} = \int_{-\infty}^{\infty} \psi_i^*(x) \psi_j(x) dx$ ，将矩阵 H 修正为 $H' = S^{-\frac{1}{2}} H S^{-\frac{1}{2}}$ ，计算 H' 的本征值 E 、本征向量 C' ，再通过 $C = S^{-\frac{1}{2}} C'$ 得到波函数解对应的本征向量 C 。

对求解程序来说，主要可以分为三个部分，分别为计算 H 矩阵、计算 S 矩阵、以及解出修正后的 H' 的本征值、本征向量。选择先用 Mathematica 计算出两个矩阵相应元素的表达式，通过 Python 程序完成第三个部分。

对第 1 问势能为 $V(x) = x^2$ 的情况，分别变化 s 和 v 进行求解。

首先变化 s 。取一组 s_i ，则 $\psi_i(x) = \sqrt{\frac{v}{\pi}}e^{-v(x-s_i)^2}$ ， $\hat{H} = -\frac{\hbar^2}{2m} \nabla^2 + x^2$ 。计算得

$$S_{ij} = \int_{-\infty}^{\infty} \psi_i^*(x) \psi_j(x) dx = \frac{\sqrt{v} e^{-\frac{1}{2}v(s_i-s_j)^2}}{\sqrt{2\pi}} \quad (1)$$

$$\begin{aligned} H_{ij} &= \int_{-\infty}^{\infty} \psi_i^*(x) \hat{H} \psi_j(x) dx \\ &= \frac{e^{-\frac{1}{2}v(s_i-s_j)^2} (2\hbar^2 v^2 (1 - v(s_i - s_j)^2) + mv(s_i + s_j)^2 + m)}{4\sqrt{2\pi} m \sqrt{v}} \end{aligned} \quad (2)$$

然后变化 v 。取一组 v_i ，则 $\psi_i(x) = \sqrt{\frac{v_i}{\pi}}e^{-v_i(x-s)^2}$ 。计算得

$$S_{ij} = \int_{-\infty}^{\infty} \psi_i^*(x) \psi_j(x) dx = \frac{\sqrt{v_i v_j}}{\sqrt{\pi} \sqrt{v_i + v_j}} \quad (3)$$

$$\begin{aligned} H_{ij} &= \int_{-\infty}^{\infty} \psi_i^*(x) \hat{H} \psi_j(x) dx \\ &= \frac{\sqrt{v_i v_j} (2\hbar^2 v_i v_j + 2ms^2(v_i + v_j) + m)}{2\sqrt{\pi} m (v_i + v_j)^{3/2}} \end{aligned} \quad (4)$$

同理，对第 2 问势能为 $V(x) = x^4 - x^2$ 的情况，也分别变化 s 和 v 进行求解。

首先变化 s , 计算得 S_{ij} 同式 (1),

$$H_{ij} = \int_{-\infty}^{\infty} \psi_i^*(x) \hat{H} \psi_j(x) dx$$

$$= \frac{(8h^2 v^3 (1 - v(s_i - s_j)^2) + m(v((s_i + s_j)^2(v(s_i + s_j)^2 - 4) + 6) - 4) + 3))}{16\sqrt{2\pi} m v^{3/2} e^{\frac{1}{2}v(s_i - s_j)^2}} \quad (5)$$

然后变化 v , 计算得 S_{ij} 同式 (3),

$$H_{ij} = \int_{-\infty}^{\infty} \psi_i^*(x) \hat{H} \psi_j(x) dx$$

$$= \frac{\sqrt{v_i v_j} (4h^2 v_i v_j (v_i + v_j))}{4\sqrt{\pi} m (v_i + v_j)^{5/2}}$$

$$+ \frac{m(4(s^2 - 1)s^2 v_i^2 + 2v_i(2s^2(2(s^2 - 1)v_j + 3) - 1))}{4\sqrt{\pi} m (v_i + v_j)^{5/2}} \quad (6)$$

$$+ \frac{m(4s^2 v_j((s^2 - 1)v_j + 3) - 2v_j + 3)}{4\sqrt{\pi} m (v_i + v_j)^{5/2}}$$

最后在程序中执行第三部分, 即解出本征值, 以及画出相应的波函数解。

程序源代码位于 SchrodingerEquation 文件夹中, 其中

- potential_V1_and_S_as_parameter.py 将 s 作为超参数解第 (1) 问
- potential_V1_and_v_as_parameter.py 将 v 作为超参数解第 (1) 问
- potential_V2_and_S_as_parameter.py 将 s 作为超参数解第 (2) 问
- potential_V2_and_v_as_parameter.py 将 v 作为超参数解第 (2) 问

程序导入的第三方库有 numpy, scipy 和 matplotlib。

3.3 伪代码

由于代码逻辑基本一致, 这里只给出一个伪代码。

Algorithm 3 Solve 1D Schrodinger equation

Output : the 3 lowest energy eigenstates and the corresponding wave functions

- 1: calculate the matrix H and matrix S
 - 2: $H' \leftarrow S^{-\frac{1}{2}} H S^{-\frac{1}{2}}$
 - 3: Solve $H' C' = E C'$ ▷ calculate eigenvalues and corresponding eigenvectors
 - 4: $indexes \leftarrow$ the 3 indexes of 3 lowest eigenvalues of E
 - 5: $C \leftarrow S^{-\frac{1}{2}} C'$
 - 6: $eigenstates \leftarrow E[indexes]$ ▷ the 3 lowest energy eigenstates
 - 7: $\Psi(x) \leftarrow \sum_{i=0}^{n-1} c_i * \psi_i(x)$ when C is eigenvector corresponding with $E[indexes]$ ▷ wave functions
-

3.4 测试用例

第 (1) 问:

1. 以 s 为变量:

```
(base) C:\Users\DELL\Desktop\ComputingPhysics>python -u "c:\Users\DELL\Desktop\ComputingPhysics\3_SchrodingerEquation\potential_V1_and_s_as_parameter.py"
The 3 lowest energy eigenstates with potential  $V(x) = x^2$  are:
1.0000
3.0000
5.0000
(base) C:\Users\DELL\Desktop\ComputingPhysics>
```

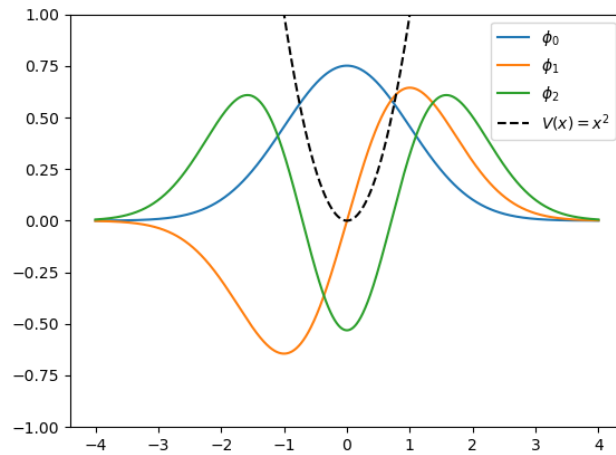


图 2: 3 lowest energy eigenstates and corresponding wave functions

2. 以 v 为变量:

```
(base) C:\Users\DELL\Desktop\ComputingPhysics>python -u "c:\Users\DELL\Desktop\ComputingPhysics\3_SchrodingerEquation\potential_V1_and_v_as_parameter.py"
The 3 lowest energy eigenstates with potential  $V(x) = x^2$  are:
1.0124
5.6380
13.1312
(base) C:\Users\DELL\Desktop\ComputingPhysics>
```

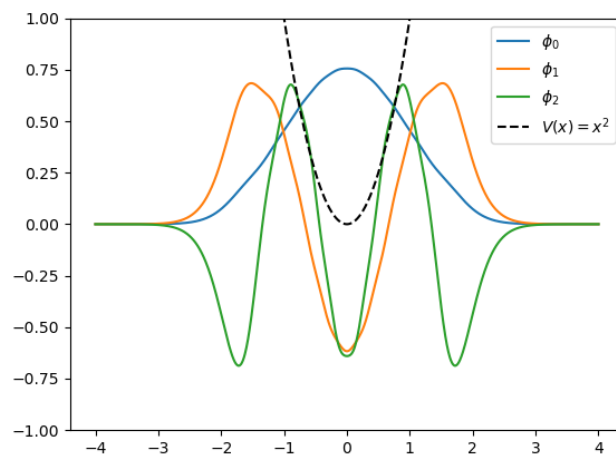


图 3: 3 lowest energy eigenstates and corresponding wave functions

第 (2) 问:

1. 以 s 为变量:

```
(base) C:\Users\DELL\Desktop\ComputingPhysics>python -u "c:\Users\DELL\Desktop\ComputingPhysics\3_SchrodingerEquation\potential_V2_and_s_as_parameter.py"
The 3 lowest energy eigenstates with potential  $V(x) = x^4 - x^2$  are:
0.6577
2.8345
6.1639
(base) C:\Users\DELL\Desktop\ComputingPhysics>
```

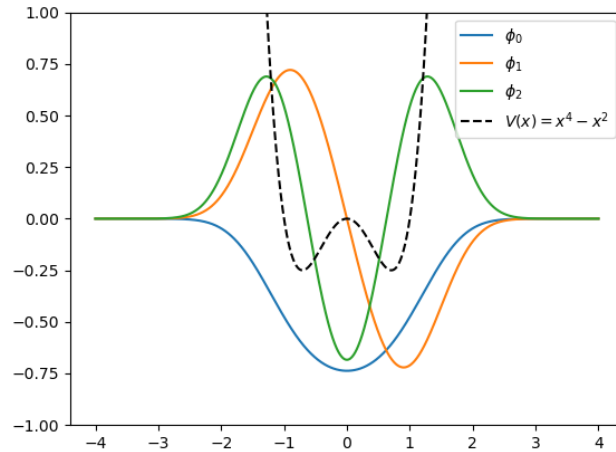


图 4: 3 lowest energy eigenstates and corresponding wave functions

2. 以 v 为变量:

```
(base) C:\Users\DELL\Desktop\ComputingPhysics>python -u "c:\Users\DELL\Desktop\ComputingPhysics\3_SchrodingerEquation\potential_V2_and_v_as_parameter.py"
The 3 lowest energy eigenstates with potential  $V(x) = x^4 - x^2$  are:
0.6577
6.1702
14.6613
(base) C:\Users\DELL\Desktop\ComputingPhysics>
```

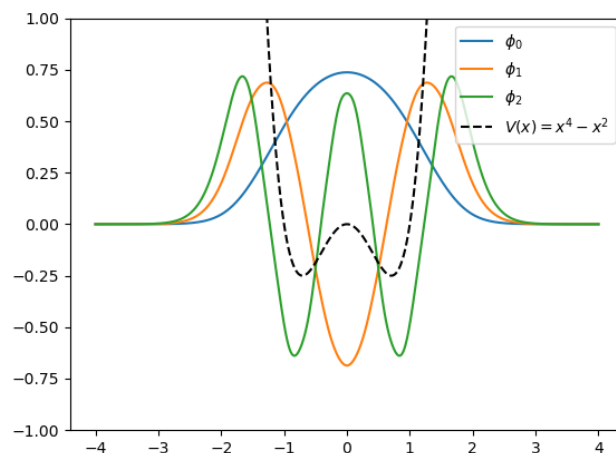


图 5: 3 lowest energy eigenstates and corresponding wave functions

从波函数图像可以发现，以 s 为变量和以 v 为变量两个波函数解似乎不一致。然而，因为只改变 v 即波包展宽时，叠加的基均为偶函数，则最后解出的波函数解也一定是偶宇称的。而改变 s 即用高斯波包的基底在一维空间中扫一遍，因为势函数是偶函数，所以最后的解为奇宇称或偶宇称。

根据谐振子势的波函数解，容易得到 s 为变量的解三个最低能量本征态分别对应基态 ($n = 1$)、第一激发态 ($n = 2$) 和第二激发态 ($n = 3$)，而以 v 为变量的解中三个最低能量本征态分别对应基态 ($n = 1$)、第二激发态 ($n = 3$) 和第四激发态 ($n = 5$)，这从波函数图像中也可以看出，实际上图 2 的 ϕ_3 即 $n = 3$ 对应图 3 的 ϕ_2 即 $n = 3$ ，图 4 的 ϕ_3 即 $n = 3$ 对应图 5 的 ϕ_2 即 $n = 3$ 。