计算物理作业 3

谢昀城 22307110070

2024 年 10 月 18 日

1 题目 1：计算 LU 分解的复杂度

1.1 题目描述

Prove that the time complexity of the LU compositionde algorithm is $O(N^3)$

1.2 证明

对于将矩阵 A 分解为上三角矩阵 U 和下三角矩阵 L, 即:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}$$

同时有 $A^T = U^T L^T$, 因此我们可以交替计算 U 的第 i 行和 L 的第 i 列, 计算过程如下：

Algorithm 1 LU Decomposition

```
for i ← 1 To N do
    for j ← i To N do
        u_ij ← a_ij − Σ_{k=1}^{i−1} l_{i,k} u_{k,j}          ▷ total of i computations
    end for                                                  ▷ total of i × (N − i + 1) computations
    for j ← i + 1 To N do
        l_ji ← a_ji − (1/u_kk) Σ_{k=1}^{i−1} l_{i,k} u_{k,j}  ▷ total of i + 1 computations
    end for                                                  ▷ total of (i + 1) × (N − i) computations
end for
```

因此,LU 分解复杂度为:

$$\sum_{i=1}^{N} i(N-i+1) + (i+1)(N-i) = N + \sum_{i=1}^{N} 2Ni - 4\sum_{i=1}^{N} i^2 = N^2(N+1) - \frac{2}{3}N(N+1)(N+2) + N \approx O(N^3)$$

2 题目 2：

2.1 题目描述

. 用 Gaussian elimination algorithm 和 partial-pivoting scheme 求解方程组：

$$2x_1 + 3x_2 + 5x_3 = 5$$

$$3x_1 + 4x_2 + 8x_3 = 6$$

$$x_1 + 3x_2 + 3x_3 = 5$$

2.2 程序描述

在本程序中，我们使用最大主元的高斯消元法来求解方程组。其中 partial_pivoting_scheme 函数用于将大于最大主元行交换到 i 位置。guaussian_elimination 函数用于进行高斯消元法求解方程组，返回增广矩阵的消元结果，解和求解情况 flag。正常求解，flag=0; 无解:x=nan,flag=2; 无穷解,flag=1, 程序将返回其中的一组解。

程序源文件为 gaussian_elimination.py，将输入方程系数以逗号分隔写于 MatrixIn.txt 文件中并置于统一文件夹下，在终端进入当前目录，使用命令 python -u gaussian_elimination.py 运行本程序。运行时请保证 Python 第三方库 Numpy 已安装。程序开发环境为 Python3.12.3，可在 Python3.8 以上版本中运行。

2.3 伪代码

Algorithm 2 PartialPivotingScheme

function PartialPivotingScheme(M, i)
    INPUT: M (2D array), i (integer)
    OUTPUT: reM (2D array)
    $reM \leftarrow M$
    $maxIndex \leftarrow i + ArgMax(|(M[i:, i]|)$
    if $i \neq maxIndex$ **then**
        $reM[maxIndex, :] \leftarrow M[i, :]$ ▷ Swap the rows
        $reM[i, :] \leftarrow M[maxIndex, :]$
    end if
    return reM
end function

Algorithm 3 GaussianElimination

function GaussianElimination(M)
    INPUT: M (2D array)
    OUTPUT: Me (2D array), x (1D array), $flag$ (integer)
    $Me \leftarrow M$
    $flag \leftarrow 0$
    $row \leftarrow$ size(Me, 0) ▷ Forward elimination
    for $i \leftarrow 0$ **To** $row - 1$ **do**
        $Me \leftarrow$ PartialPivotingScheme(Me, i)
        for $j \leftarrow i + 1$ **To** $row - 1$ **do**
            $Me[j, :] \leftarrow Me[j, :] - Me[i, :] * Me[j, i]/Me[i, i]$
        end for
    end for ▷ Backward substitution

$x \leftarrow \text{zeros}(row)$

for $i \leftarrow row - 1$ **To** 0 **step** -1 **do**

  for $j \leftarrow row - 1$ **To** $i + 1$ **step** -1 **do**

    $x[i] \leftarrow x[i] - Me[i, j] * x[j]$

  end for

  $x[i] \leftarrow x[i] + Me[i, -1]$

  if $Me[i, i] = 0$ and $x[i] \neq 0$ **then**

    $flag \leftarrow 2$

    return Me, nan, $flag$

  end if

  if $Me[i, i] = 0$ and $x[i] = 0$ **then**

    $x[i] \leftarrow 1$

    $flag \leftarrow 1$

    continue

  end if

  $x[i] \leftarrow x[i]/Me[i, i]$

end for

return Me, x, $flag$

end function

2.4 输入输出实例

对于本程序，运行后读取 MatrixIn.txt 文件中的系数矩阵，输出增广矩阵的消元结果，解和求解情况。程序运行截图如图1所示。程序得到方程组的解为 $x_1 = 2, x_2 = 2, x_3 = -1$ 此外, 图23展示了无解和无穷解的程序输出情况。

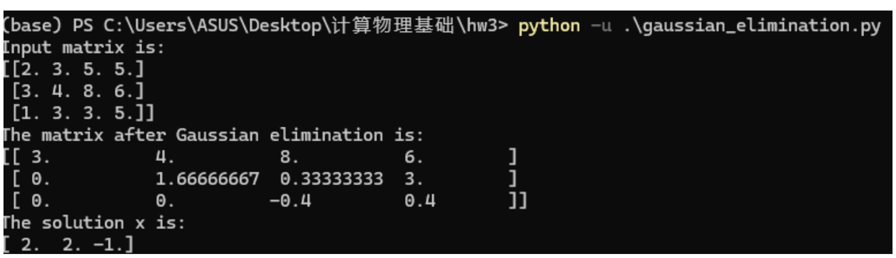


图 1: 题目 2 程序运行截图

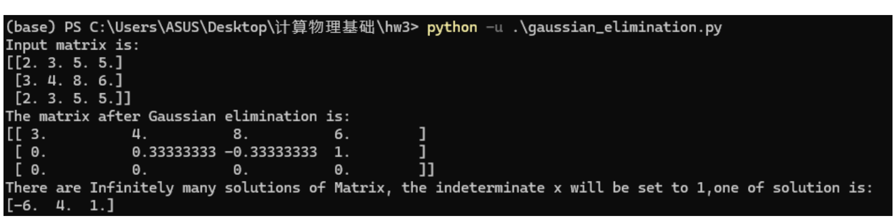


图 2: 题目 2 程序运行截图: 无穷解情况

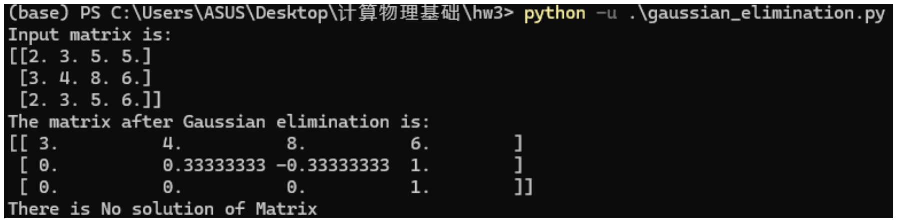


图 3: 题目 2 程序运行截图: 无解情况

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. The Gaussian basis functions are defined as: $\phi_i(x) = (\frac{v_i}{\pi}^(1/2)e^(- v_i(x - s_i)^2))$his function has two variational parameters: v_i, the width of the Gaussian, and s_i , the center of the Gaussian. For simplicity, we only vary one of these parameters at a time and do calculations with either fixed widths or fixed centers.

3.2 程序描述

在本程序中，我们通过将波函数在高斯基上展开, 并求解关于系数的广义本征值问题 $HC = ESC$，以得到 $V(x) = x^2$ 和 $V(x) = x^4 - x^2$ 的 $n = 1, 2, 3$ 时的能量和波函数。

其中:

$$H_{ij} = \langle\phi_i|\hat{H}|\phi_j\rangle, \quad S_{ij} = \langle\phi_i|\phi_j\rangle, \Psi(x) = \sum C_i\phi_i(x)$$

由于 $V(x)$ 关于 $x = 0$ 对称，波函数将存在奇宇称和偶宇称两种情况, 而高斯基关于 s_i 对称，因此我们固定 v_i，变化 s_i 将会得到更好的效果。

因此，本程序中我们固定 $v_i = 1$，在 $(-20, 20)$ 范围内，均匀选取 100 组 s_i 展开波函数 Ψ，并且使用 scipy 库中的 eigh 函数求解广义本征值问题，得到特征值值和归一化特征向量，最终得到最低的三个能级能量和波函数。

为方便我们选取单位为 $\bar{h} = 1, m = 1$, 因此 $\hat{H} = -\frac{1}{2}\frac{d^2}{dx^2} + V(x)$ 可以得到:

$$S_{Si,Sj} = \frac{e^{-\frac{1}{2}(S_i - S_j)^2}}{\sqrt{2\pi}}$$

对 $V(x) = x^2$:

$$H_{S_i,S_j} = \frac{e^{-\frac{1}{2}(S_i - S_j)^2}\left(-3 + S_i^2 - 6S_iS_j + S_j^2\right)}{4\sqrt{2\pi}}$$

对 $V(x) = x^4 - x^2$:

$$H_{S_i,S_j} = \frac{e^{-\frac{1}{2}(S_i - S_j)^2}\left(7 + S_i^4 + 4S_i^3S_j - 6S_j^2 + S_j^4 + 6S_i^2(-1 + S_j^2) + 4S_iS_j(5 + S_j^2)\right)}{16\sqrt{2\pi}}$$

本程序源文件为 SchrodingerEq.py，在终端进入当前目录，使用命令 python -u SchrodingerEq.py 运行本程序。运行时请保证 Python 第三方库 Numpy,Matplotlib,scipy 已安装。程序开发环境为 Python3.12.3，可在 Python3.8 以上版本中运行。

3.3 伪代码

3.3.1 Trapezoidal integrate 伪代码:

Algorithm 4 SolveSchrodingerEq

for $i \leftarrow 0$ **To** N **do**
  for $j \leftarrow 0$ **To** N **do**
    $H_{ij} \leftarrow \langle\phi_i|\hat{H}|\phi_j\rangle$
    $S_{ij} \leftarrow \langle\phi_i|\phi_j\rangle$
  end for
end for
$E, C \leftarrow eigh(H, S)$
for $i \leftarrow 0$ **To** 3 **do**
  $Energy_i \leftarrow E[i]$
  $\Psi_i(x) = \leftarrow \sum C_i\phi_i(x)$
end for

3.4 输入输出实例

对于本程序,运行后会生成两种势能下的归一化波函数和势能图4和5,分别为"wavefunction of V=x2.png" 和"wavefunction of V=x4-x2.png",于当前目录下,并输出两种势能下的最低三个能级能量。程序运行截图如图6所示。得到能量为($\bar{h} = 1, m = 1$):

$$V(x) = x^2 : E_0 = 0.70710678, E_1 = 2.12132034, E_2 = 3.53553391$$

$$V(x) = x^4 - x^2 : E_0 = 0.33796154, E_1 = 1.61273703, E_2 = 3.66191064$$

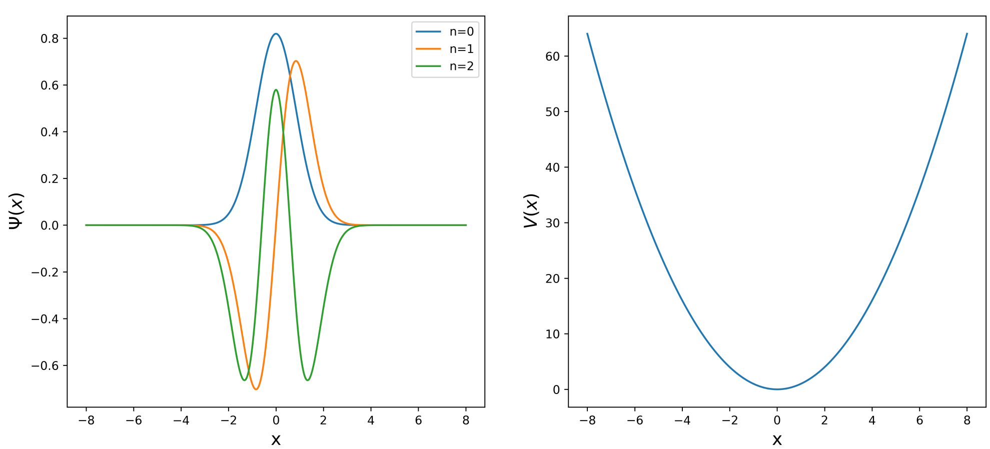


图 4: $V(x) = x^2$ 时的归一化波函数和势能

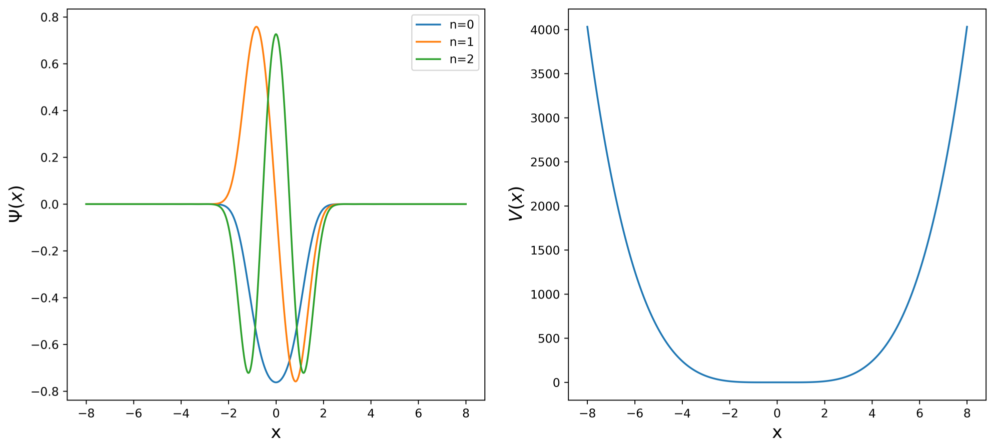
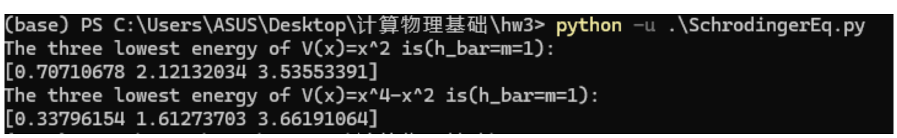


图 5: $V(x) = x^4 - x^2$ 时的归一化波函数和势能



```
(base) PS C:\Users\ASUS\Desktop\计算物理基础\hw3> python -u .\SchrodingerEq.py
The three lowest energy of V(x)=x^2 is(h_bar=m=1):
[0.70710678 2.12132034 3.53553391]
The three lowest energy of V(x)=x^4-x^2 is(h_bar=m=1):
[0.33796154 1.61273703 3.66191064]
```

图 6: 题目 3 程序运行截图