计算物理作业 n

李明钰 22307110156

2024 年 10 月 17 日

1 题目 1：证明高斯消元法的时间复杂度为 $O(N^3)$

针对一个 $n \times n$ 的矩阵

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \tag{1}$$

根据高斯消元法 (算法1) 将其变为上三角阵，

$$A' = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \tag{2}$$

这对其中要进行消元的 n 列中的第 p 列，需要

- 找到当前列的最大值，并将其所在行放在第 p 行，这需要时间 $O(N)$
- 对第 q 行，$q = p+1, \ldots, n$ 需要进行如下操作

  计算因子 $f = \frac{a_{pq}}{app}$, 这需要时间 $O(1)$

  更新第 q 行所有元素，这需要时间 $O(N)$
- 对所有列重复这两步，这需要时间 $O(N)$

综上，高斯消元法的总时间复杂度为 $O(N^3)$

1.1 伪代码

如算法1所示

Algorithm 1 Gaussian Elimination Algorithm

procedure GAUSSIANELIMINATION(A, b)
  $n \leftarrow$ size of A
  for $p \leftarrow 1$ to $n-1$ **do**
    Find the maximum element $A[i,p]$ in column p below row p
    $maxRow \leftarrow i$
    if $A[maxRow, p] = 0$ **then**
      continue
    end if
    Swap rows p and $maxRow$ in A and b
    for $i \leftarrow p+1$ to n **do**
      $f \leftarrow \frac{A[i,p]}{A[p,p]}$
      for $j \leftarrow p$ to n **do**
        $A[i,j] \leftarrow A[i,j] - f \cdot A[p,j]$
      end for
      $b[i] \leftarrow b[i] - f \cdot b[p]$
    end for
  end for
  return BackwardSubstitution(A, b)
end procedure

2 题目 2：用高斯消元法和 partial-pivoting scheme 求解方程组

2.1 题目描述

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

$$\begin{aligned} 2x_1 + 3x_2 + 5x_3 &= 5 \\ 3x_1 + 4x_2 + 8x_3 &= 6 \\ x_1 + 3x_2 + 3x_3 &= 5 \end{aligned} \tag{3}$$

程序先将其改写为矩阵相乘的形式

$$\mathbf{A}\vec{x} = \vec{b} \tag{4}$$

其中，

$$\mathbf{A} = \begin{bmatrix} 2 & 3 & 5 \\ 3 & 4 & 8 \\ 1 & 3 & 3 \end{bmatrix} \qquad \vec{b} = (5, 6, 6)^{\mathrm{T}} \tag{5}$$

然后使用高斯消元法将扩展矩阵化为上三角阵，之后解出 $\vec{x} = (2, 2, -1)$，与理论解一致。

2.2 伪代码

如算法2所示

Algorithm 2 Gaussian Elimination Algorithm with Partial-pivoting Scheme

procedure GaussianElimination($A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n$)
 $n \leftarrow$ size of A
 for $i \leftarrow 0$ to $n-2$ **do**
  $p \leftarrow i + \text{argmax}(A^T[i, i:])$ ▷ Find the pivot index
  $A_c \leftarrow \text{copy}(A)$
  $b_c \leftarrow \text{copy}(b)$
  $A[i] \leftarrow A_c[p]$ ▷ Swap the pivot row with the i-th row
  $A[p] \leftarrow A_c[i]$
  $b[i] \leftarrow b_c[p]$
  $b[p] \leftarrow b_c[i]$
  for $j \leftarrow i+1$ to $n-1$ **do**
   $f \leftarrow \frac{A[j][i]}{A[i][i]}$
   $A[j] \leftarrow A[j] - f \cdot A[i]$
   $b[j] \leftarrow b[j] - f \cdot b[i]$
  end for
 end for
 $x \leftarrow \text{zeros}(n)$
 for $k \leftarrow n-1$ down to 0 **do**
  $x[k] \leftarrow \frac{b[k] - \text{dot}(A[k], x)}{A[k][k]}$
 end for
 return x
end procedure

2.3 运行实例

```
A_sup,b_sup,x=GaussianElimination(A,b)
print("通过高斯消元将A化为上三角阵为\n{}\n，对应b向量为为{}，\n对应得到方程解为{}".format(A_sup,b_sup,x))
```

```
通过高斯消元将A化为上三角阵为
[[ 3.          4.          8.        ]
 [ 0.          1.66666667  0.33333333]
 [ 0.          0.         -0.4       ]]
，对应b向量为为[6.  3.  0.4]，
对应得到方程解为[ 2.  2. -1.]
```

3 题目 2：变分法解薛定谔方程

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

对高斯基函数：

$$\Phi_i(x) = (\frac{v_i}{\pi})^{1/2} e^{-v_i(x-s_i)^2} \tag{6}$$

我们固定其方差不变，即限定所有 v_i 都等于 1，只改变其 s_i 的值，根据公式

$$S_{pk} = \int_{-\infty}^{\infty} \Phi_p^*(x)\Phi_k(x)dx \tag{7}$$

和公式

$$H_{pk} = \int_{-\infty}^{\infty} \Phi_p^*(x)H\Phi_k(x)dx \tag{8}$$

借助 Mathematica 计算（文件夹中积分运算.nb）出 H_{pk} 与 v_p, v_k 的解析关系，为再在 (−200 200) 之间均匀取 n 个值，作为基函数中心 s_i 的取值，计算出相应具体的矩阵 S 和 H。再求解广义特征值问题

$$HC = ESC \tag{9}$$

即可得到本征值和高斯基下的对应的本征矢。

3.3 输出实例

3.3.1 不同基数量下的本征值

模拟计算出的 $V = x^2$ 下的最低的三个能量本征值与选取基函数数量本征值对应关系如表3.3.1所示

3.4 运行实例

设定 $n = 600$ 计算得到最低三个能级的本征值如图1所示，画出相应的本征态如图23所示

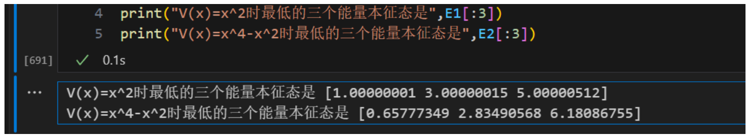

图 1: 输出最低三个能级本征值

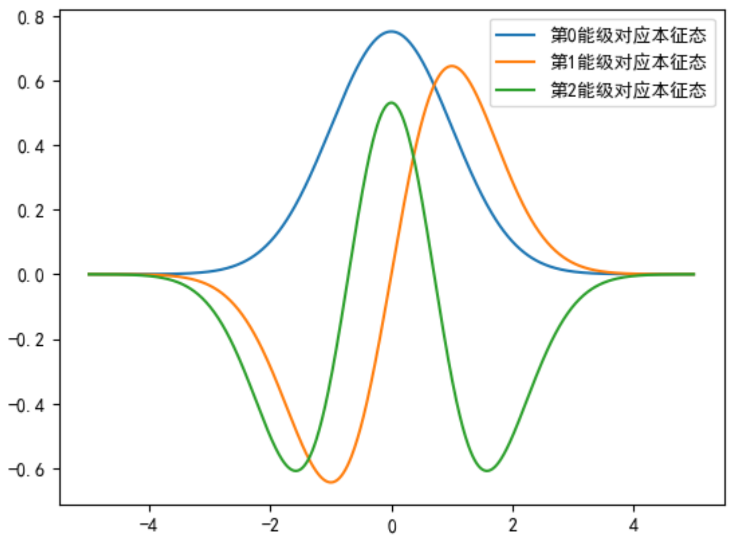

图 2: $V = x^2$ 最低三个能级对应本征态函数

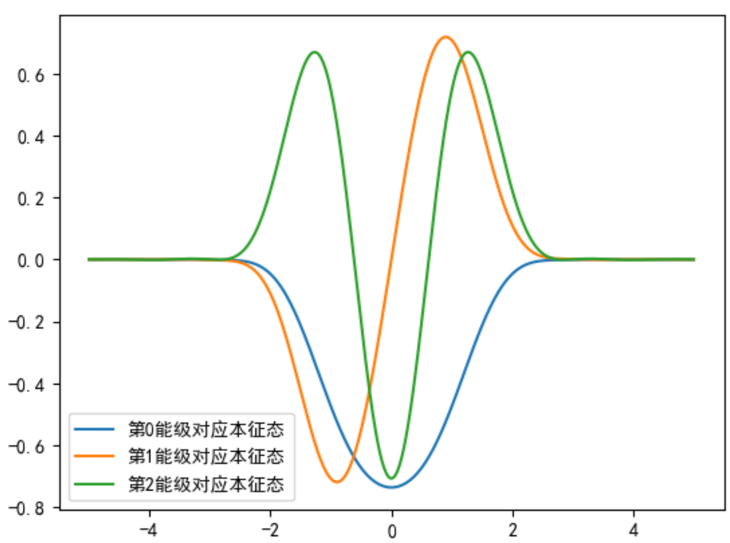

图 3: $V = x^4 - x^2$ 最低三个能级对应本征态函数

基数量 n	E1	E2	E3
50	17.90972511	17.90972511	151.18752603
100	5.32539866	5.33703698	37.98094899
200	1.66828647	3.02394632	10.48343753
300	1.04669018	3.04764203	5.97892026
400	1.00101988	3.0040235	5.07659885
500	1.00000583	3.00005203	5.00137461
600	1.00000001	3.00000015	5.00000512

3.5 伪代码

Algorithm 3 Gaussian Basis Function and Matrix Calculation

x, s, v (for Gaussian basis), v_i, v_j, s_i, s_j (for H and S matrices) S, H_1, H_2, E_1, E_2, C_1, C_2

GaussianBasisFuncGaussianBasisFunc CalculateSCalculate_S CalculateH1Calculate_H1 CalculateH2Calculate_H2

Define GaussianBasisFunc(x, s, v):

Require: return $\sqrt{\frac{v}{\pi}}\exp(-v(x-s)^2)$

Define Calculate_S(v_i, v_j, s_i, s_j):

exponent $\leftarrow -\frac{v_i v_j (s_i - s_j)^2}{v_i + v_j}$

return $\frac{\sqrt{v_i}\sqrt{v_j}\exp(\text{exponent})}{\sqrt{\pi}\sqrt{v_i+v_j}}$

Define Calculate_H1(v_i, v_j, s_i, s_j):

part1 $\leftarrow \frac{1}{2\sqrt{\pi}(v_i+v_j)^{5/2}}$

exponent $\leftarrow -\frac{(s_i-s_j)^2 v_i v_j}{v_i+v_j}$

part2 $\leftarrow \exp(\text{exponent})$

part3 $\leftarrow \sqrt{v_i}\sqrt{v_j}$

part4 $\leftarrow v_i + v_j + 2s_j^2 v_j^2 + 4v_i v_j(s_i s_j + v_j) + 2v_i^2(s_i^2 + 2v_j - 4(s_i - s_j)^2 v_j^2)$

$H_1 \leftarrow$ part1 $\times$ part2 $\times$ part3 $\times$ part4

return H_1

Define Calculate_H2(v_i, v_j, s_i, s_j):

term1 $\leftarrow \frac{1}{4\sqrt{\pi}(v_i+v_j)^{9/2}}\sqrt{v_i}\sqrt{v_j}$

exp_part $\leftarrow \exp(-\frac{(s_i-s_j)^2 v_i v_j}{v_i+v_j})$

term2_vi2 $\leftarrow v_i^2(3 - 2v_i + 4s_i^2 v_i(3 + (-1 + s_i^2)v_i))$

term2_vi_vj $\leftarrow 2v_i(3 + v_i(-3 + s_i^2(6 - 4v_i) - 4s_i s_j(-3 + v_i) + 8s_i^3 s_j v_i))v_j$

term2_vj2 $\leftarrow (3 + 2v_i(-3 + 4s_i s_j(3 - 2v_i) - 2s_j^2(-3 + v_i) + 2s_i^2(-1 + 6s_j^2)v_i))v_j^2$

term2_vj3 $\leftarrow 2(-1 + 2s_j(-2s_i v_i + s_j(3 + (-2 + 4s_i s_j)v_i)))v_j^3$

term2_vj4 $\leftarrow 4s_j^2(-1 + s_j^2)v_j^4$

correction_term $\leftarrow -8v_i v_j(v_i + v_j)^2(-v_j + v_i(-1 + 2(s_i - s_j)^2 v_j))$

term2 $\leftarrow$ term2_vi2 + term2_vi_vj + term2_vj2 + term2_vj3 + term2_vj4 + correction_term

$H_2 \leftarrow$ term1 $\times$ exp_part $\times$ term2

return H_2

Define Constants:

$n\_basis \leftarrow 600$ (number of basis functions)

$v \leftarrow 1$ (constant width of basis functions)

$n\_x \leftarrow 600$ (number of points for plotting)

$s\_max \leftarrow 200$ (maximum value of basis function centers)

$x\_max \leftarrow 5$

$s\_min \leftarrow -s\_max$

$s\_array \leftarrow \text{linspace}(s\_min, s\_max, n\_basis)$

Initialize Matrices:

$H_1 \leftarrow \mathbf{0}_{n\_basis \times n\_basis}$

$H_2 \leftarrow \mathbf{0}_{n\_basis \times n\_basis}$

$S \leftarrow \mathbf{0}_{n\_basis \times n\_basis}$

Calculate H and S Matrices:

for $i \leftarrow 0$ **to** $n\_basis - 1$ **do**

for $j \leftarrow 0$ **to** $n\_basis - 1$ **do** $H_1[i][j] \leftarrow \text{Calculate\_H1}(v, v, s\_array[i], s\_array[j])$

$H_2[i][j] \leftarrow \text{Calculate\_H2}(v, v, s\_array[i], s\_array[j])$

$S[i][j] \leftarrow \text{Calculate\_S}(v, v, s\_array[i], s\_array[j])$

Compute Eigenvalues and Eigenvectors:

$E_1, C_1 \leftarrow \text{eigh}(H_1, S)$

$E_2, C_2 \leftarrow \text{eigh}(H_2, S)$

Print Results:

print "Lowest three energy eigenstates for $V(x) = x^2$: $E_1[:3]$"

print "Lowest three energy eigenstates for $V(x) = x^4 - x^2$: $E_2[:3]$"

=0

[H]