数值分析第 4 次上机作业

学号：221840189，姓名：王晨光

§1 问题一

1.1 问题

实现高斯消去法，了解高斯列选主元消去法（课本第三章）.

1.2 算法思路

对于三角方程组 $Ax=b$，其中 $A\in\mathbb{R}^{n\times n}$ 为上三角矩阵, $x,b\in\mathbb{R}^n$. 若 $|A|\neq 0$, 即 $a_{ii}\neq 0, 1\leqslant i\leqslant n$, 通过回代的方式，容易得到方程组有唯一解

$$\begin{cases} x_n=\dfrac{b_n}{a_{nn}} \\ x_i=\left(b_i-\displaystyle\sum_{j=i+1}^{n}a_{ij}x_j\right)/a_{ii} \quad , i=n-1,n-2,\dots,1 \end{cases}$$

以下是实现列主元高斯消去法的基本步骤：
1. 构建增广矩阵：将线性方程组的系数矩阵和常数向量合并成一个增广矩阵.
2. 选取列主元：在每一次消元操作中，选择当前列中绝对值最大的元素作为主元.
3. 行交换：将含有主元的行与当前操作行交换，确保主元位于当前操作行的第一个位置.
4. 消元计算：通过对当前操作行进行线性组合，将当前列下方的所有元素消为零.
5. 重复步骤 2-4，直到得到上三角矩阵形式.
6. 回代求解：从最后一行开始，通过回代计算出未知数的值.

Algorithm 1 高斯消去法

Require: 系数矩阵 $A=(a_{ij}), i,j=1,2,\ldots,n$, 值向量 $b=(b_i), i=1,2,\ldots,n$

Ensure: 方程组无法求解或结果向量 $x=(x_i)(i=1,2,\ldots,n)$

function GAUSSIAN ELIMINATION(A,b)
  for k from 1 to $n-1$ **do**
    if $a_{kk}=0$ **then**
      输出“方程组无法求解：算法终止”
    end if
    for i from $k+1$ to n **do**
      $l_{ik} \leftarrow \dfrac{a_{ik}}{a_{kk}}$
      $a_{ik} \leftarrow l_{ik}$
      $b_i \leftarrow b_i - l_{ik} b_k$
      for j from $k+1$ to n **do**
        $a_{ij} \leftarrow a_{ij} - l_{ik} a_{kj}$
      end for
    end for
  end for
  if $a_{nn}=0$ **then**
    输出“方程组无法求解：算法终止”
  else
    $x_n \leftarrow \dfrac{b_n}{a_{nn}}$
    $b_n \leftarrow x_n$
    for i from $n-1$ to 1 **do**
      $x_i \leftarrow \dfrac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}$
      $b_i \leftarrow x_i$
    end for
  end if
  return $x=(x_i), i=1,2,\ldots,n$
end function

1.3 结果分析

通过 Python 编程实现高斯消去法的完整过程，完整代码可见附录.

考虑求解实际问题：求解线性方程组 $Ax=b$, 其中 $A=\begin{pmatrix} 2 & 1 & -1 \\ -3 & -1 & 2 \\ -2 & 1 & 2 \end{pmatrix}$, $b=\begin{pmatrix} 8 \\ -11 \\ -3 \end{pmatrix}$. 通过该程序解得 $x=\begin{pmatrix} 2 \\ 3 \\ -1 \end{pmatrix}$，经检验，结果正确无误.

§ 2 问题二

2.1 问题

了解矩阵的 LU 分解.

2.2 算法思路

矩阵的 LU 分解是指矩阵 $A = LU$ 的分解形式，其中 L 是单位下三角矩阵，U 是上三角矩阵. LU 分解可以通过 Gauss 消去法实现. 思路如下：

若想将给定矩阵 A 分解为下三角矩阵 L 和上三角矩阵 U，一个思路就是通过一系列的初等变换将 A 化为上三角矩阵，且保证这些变换的乘积是一个下三角，比如通过初等变换 $L_nL_{n-1}\dots L_1A = U$，则 $A = L_1^{-1}L_2^{-1}\dots L_n^{-1}U = (L_n \dots L_2L_1)^{-1}U$，其中 U 是一个上三角矩阵，$(L_nL_{n-1}\dots L_1)^{-1}$ 是一个下三角矩阵。所以问题就转化为找满足条件的下三角矩阵，对于任意给定的向量 $x \in \mathbb{R}^n$，找一个简单的下三角矩阵 L_k 使 x 经过这一矩阵的作用之后的第 k+1 至第 n 个分量均为 0。能够完成这一条件的最简单的下三角矩阵如下：

$$L_k = I_n - l_k e_k^T$$

其中

$$l_k = (0, \dots, 0, l_{k+1,k}, \dots, l_{n,k})^T$$

即

$$L_n = \begin{pmatrix} 1 & & & & & 0 \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & \ddots & & \\ & & \vdots & & \ddots & \\ 0 & & -l_{n,k} & & & 1 \end{pmatrix}$$

对于任意给定的向量

$$x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$$

则

$$L_kx = (x_1, \dots, x_k, x_{k+1} - x_k l_{k+1,k}, \dots, x_n - x_k l_{n,k})$$

若要使得第 k+1 至第 n 个分量均为 0，则

$$l_{i,k} = \frac{x_i}{x_k}, i = k+1, \dots, n$$

此时

$$L_kx = (x_1, \dots, x_k, 0, \dots, 0)$$

且 Gauss 变换 L_k 的性质非常好，Gauss 变换的逆特别容易求

$$(I_n - l_k e_k^T)(I_n + l_k e_k^T) = I_n - l_k e_k^T l_k e_k^T = I_n$$

即

$$L_k^{-1} = I_n + l_k e_k^T$$

此时

$$A = L_1^{-1}L_2^{-1}\cdots L_n^{-1}U = LU$$

其中

$$L = L_1^{-1}L_2^{-1}\cdots L_n^{-1} = \left(I_n + l_1e_1^T\right)\left(I_n + l_2e_2^T\right)\cdots\left(l_{n-1}e_{n-1}^T\right) = I_n + l_1e_1^T + \cdots + l_{n-1}e_{n-1}^T$$

则 L 具有如下形状

$$L = I_n + (l_1, l_2, \ldots, l_{n-1}, 0) = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix}$$

所以 L 是一个单位下三角矩阵，矩阵 A 可以分解为单位下三角 L 和上三角矩阵 U 的形式.

2.3 结果分析

举例说明利用 Gauss 消去法实现 A 的 LU 分解：

$$A = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 10 \end{pmatrix} = (a_1, a_2, a_3)$$

计算 L_1，设

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -l_{2,1} & 1 & 0 \\ -l_{3,1} & 0 & 1 \end{pmatrix}$$

则 $L_1 a_1$ 的第二个分量和第三个分量为 0，即

$$\begin{cases} -l_{2,1} + 2 = 0 \Rightarrow l_{2,1} = 2 \\ -l_{3,1} + 3 = 0 \Rightarrow l_{3,1} = 3 \end{cases}$$

则 $L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}$. 接下来计算 L_2，设

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -l_{3,2} & 1 \end{pmatrix}$$

则 $L_2 b_2$ 的第三个分量为 0，即

$$3l_{3,2} - 6 = 0 \Rightarrow l_{3,2} = 2$$

则

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}$$

此时

$$L_2 (L_1 A) = \begin{pmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{pmatrix} = U$$

最后令 $L = (L_2 L_1)^{-1}$ 即得 A 的 LU 分解.

§ 3 问题三

3.1 问题

求解三对角线性方程组的追赶法或 Thomas 算法.

3.2 算法思路

三对角线性方程组是指形如下的线性方程组：

$$\begin{cases} a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i, \quad i = 1, 2, \ldots, n \\ a_1 = c_n = x_0 = x_{n+1} = 0 \end{cases}$$

其中 a_i, b_i, c_i, d_i 是已知常数. 以下是三对角线性方程组的矩阵形式：

$$\begin{pmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{pmatrix}$$

做三次样条插值时，我们常需要解三对角矩阵.

追赶法（或 Thomas 算法）是一种基于高斯消元法的算法，分为两个阶段：向前消元 (forward elimination) 和回代 (backward substitution).

1. 向前消元：

通过高斯消元法，可以变形得到新的线性方程组：

$$\begin{pmatrix} 1 & \lambda_1 & & & 0 \\ 0 & 1 & \lambda_2 & & \\ & 0 & 1 & \ddots & \\ & & \ddots & \ddots & \lambda_{n-1} \\ 0 & & & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_n \end{pmatrix}$$

其中：

$$\begin{cases} \lambda_i = \dfrac{c_i}{b_i - a_i \lambda_{i-1}}, \quad i = 2, 3, \ldots, n \\ \rho_i = \dfrac{d_i - a_i \rho_{i-1}}{b_i - a_i \lambda_{i-1}}, \quad i = 2, 3, \ldots, n \\ \lambda_1 = \dfrac{c_1}{b_1}, \\ \rho_1 = \dfrac{d_1}{b_1} \end{cases}$$

2. 回代：

得到变形后的线性方程组：

$$\begin{cases} x_i + \lambda_i x_{i+1} = \rho_i, \quad i = 1, 2, \ldots, n-1 \\ x_n = \rho_n \end{cases}$$

将 x_i 逆序求出即可，如下：

$$\begin{cases} x_n = & \rho_n, \\ x_i = & \rho_i - \lambda_i x_{i+1}, \quad i = n-1, n-2, \ldots, 1 \end{cases}$$

整理全部步骤，可以得到一般性公式：$x_i = d_i' - c_i' x_{i+1}$，其中：

$$c_i' = \begin{cases} \dfrac{c_i}{b_i}, & i = 1 \\ \dfrac{c_i}{b_i - c_{i-1}' a_i}, & i = 2, 3, \ldots, n-1 \end{cases}$$

$$d_i' = \begin{cases} \dfrac{d_i}{b_i}, & i = 1 \\ \dfrac{d_i - d_{i-1}' a_i}{b_i - c_{i-1}' a_i}, & i = 2, 3, \ldots, n \end{cases}$$

Algorithm 2 追赶法

Require: 方程组的阶数 n，系数矩阵 A 的元素 $\{a_i\},\{b_i\},\{c_i\}, i=1,2,\dots,n$, 值向量 $d=(d_i), i=1,2,\dots,n$

Ensure: 方程组无法求解或结果向量 $x=(x_i)(i=1,2,\dots,n)$

function THOMAS ALGORITHM(n, A, b)
  if $b_1 = 0$ **then**
    输出“方程组无法求解：算法终止”
  end if
  $p_1 \leftarrow d_1$
  $q_1 \leftarrow \frac{c_1}{d_1}$
  for k from 2 to $n-1$ **do**
    $p_k \leftarrow d_k - a_k q_{k-1}$
    if $p_k = 0$ **then**
      输出“方程组无法求解：算法终止”
    end if
    $q_k \leftarrow \frac{c_k}{p_k}$
  end for
  $p_n \leftarrow d_n - a_n q_{n-1}$
  if $p_n = 0$ **then**
    输出“方程组无法求解：算法终止”
  end if
  $y_1 \leftarrow \frac{b_1}{p_1}$
  for k from 2 to n **do**
    $y_k \leftarrow \frac{b_k - a_k y_{k-1}}{p_k}$
  end for
  $x_n \leftarrow y_n$
  for i from $n-1$ to 1 **do**
    $x_k \leftarrow y_k - q_k x_{k+1}$
  end for
  return $x = (x_i), i = 1,2,\dots,n$
end function

3.3 结果分析

通过 Python 编程实现追赶法的完整过程，完整代码可见附录.

考虑求解实际问题：求解线性方程组 $Ax=b$, 其中 $A=\begin{pmatrix} 2 & 1 & & \\ 1 & 2 & 1 & \\ & 1 & 2 & 1 \\ & & 1 & 2 \end{pmatrix}$, $b=\begin{pmatrix} 5 \\ 6 \\ 7 \\ 8 \end{pmatrix}$. 通过该程序解得 $x=\begin{pmatrix} 1.6 \\ 1.8 \\ 0.8 \\ 3.6 \end{pmatrix}$，经检验，结果正确无误.

§4 附录：高斯消去法（问题一）程序代码

```python
import numpy as np


def gaussian_elimination(A, b):
    n = len(A)

    # 将数组的数据类型转换为float64
    A = A.astype(np.float64)
    b = b.astype(np.float64)

    # 高斯消元
    for i in range(n - 1):
        max_idx = i

        # 选取列主元
        for j in range(i + 1, n):
            if abs(A[j][i]) > abs(A[max_idx][i]):
                max_idx = j

        # 交换行
        A[[i, max_idx]] = A[[max_idx, i]]
        b[[i, max_idx]] = b[[max_idx, i]]

        for j in range(i + 1, n):
            # 计算倍数
            multiplier = A[j][i] / A[i][i]

            # 更新矩阵
            A[j][i:] -= multiplier * A[i][i:]
            b[j] -= multiplier * b[i]

    # 回代求解
    x = np.zeros(n)
    for i in range(n - 1, -1, -1):
        x[i] = (b[i] - np.dot(A[i][i + 1:], x[i + 1:])) / A[i][i]

    return x

# 举例
A=np.array([[2,1,-1],
            [-3,-1,2],
            [-2,1,2]])
b=np.array([8,-11,-3])
```

```
x=gaussian_elimination(A,b)
print("方程组的解为：", x)
```

§5 附录：追赶法（问题二）程序代码

```python
def thomas_algorithm(A, b):
    n = len(b)
    a = [0] * n
    c = [0] * n
    x = [0] * n

    # 从矩阵 A 中提取三对角线系数
    for i in range(n):
        x[i] = A[i][i]
        if i > 0:
            a[i] = A[i][i - 1]
        if i < n - 1:
            c[i] = A[i][i + 1]

    # 第一步：向前消元
    c_prime = [0] * n
    d_prime = [0] * n
    c_prime[0] = c[0] / x[0]
    d_prime[0] = b[0] / x[0]
    for i in range(1, n):
        m = 1.0 / (x[i] - a[i] * c_prime[i - 1])
        c_prime[i] = c[i] * m
        d_prime[i] = (b[i] - a[i] * d_prime[i - 1]) * m

    # 第二步：回代
    x[n - 1] = d_prime[n - 1]
    for i in range(n - 2, -1, -1):
        x[i] = d_prime[i] - c_prime[i] * x[i + 1]

    return x

# 举例：
A = [[2, 1, 0, 0],
     [1, 2, 1, 0],
     [0, 1, 2, 1],
     [0, 0, 1, 2]]
b = [5, 6, 7, 8]
x = thomas_algorithm(A, b)
print("方程组的解为：", x)
```