计 算 方 法 实 验 2

System of Linear Equations
Nonlinear Equations

何军辉
hejh@scut.edu.cn

实验说明

- 实验内容：
  - 编写程序调试，用作业题验证算法；
- 考核方式：
  - 实验报告（90%）
  - 实验考勤（10%）
- 提交方式：
  - 文件命名：学号-XXX-LAB2
  - 以班级为单位提交电子版至下面邮箱：hejh@scut.edu.cn

2.1 顺序高斯消去法

① 输入系数矩阵A右端项b及ϵ

② 消元

- 对$k = 1,2,\cdots,n-1$循坏
- 若$|a_{kk}| \le \epsilon$，则打印“求解失败”，停止；否则

$$\text{for } i = k+1 \text{ to } n \text{ do}$$
$$T = \frac{a_{ik}}{a_{kk}}$$
$$b_i = b_i - T \times b_k$$
$$\text{for } j = k+1 \text{ to } n \text{ do}$$
$$a_{ij} = a_{ij} - T \times a_{kj}$$

2.1 顺序高斯消去法

③ 回代:

- 若$|a_{nn}| \le \epsilon$，则打印“求解失败”，停止；否则

$$x_n = \frac{b_n}{a_{nn}}$$

$$\text{for } \ i = n-1 \ \text{ downto } 1 \ \text{ do}$$

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}$$

④ 打印$x_i, i = 1,2,\cdots,n$

2.2 列主元高斯消去法

① 输入$\boldsymbol{A},\boldsymbol{b}$及$\epsilon$

② 选主元及消元：

$$\text{for } k = 1 \text{ to } n-1 \text{ do}$$

选主元：$T = \left|a_{i_k,k}\right| = \max_{k \le i \le n} |a_{ik}|$

若$T < \epsilon$，则打印“求解失败”，停止；否则

若$i_k \neq k$,则交换$\boldsymbol{A}$的第i_k行与第k行，交换b_{i_k}与b_k否则

消元：

$$\text{for } i = k+1 \text{ to } n \text{ do}$$

$$T = \frac{a_{ik}}{a_{kk}};\quad b_i = b_i - T \times b_k$$

$$\text{for } j = k+1 \text{ to } n \text{ do}$$

$$a_{ij} = a_{ij} - T \times a_{kj}$$

2.2 列主元高斯消去法

③ 回代:

- 若$|a_{nn}| \le \epsilon$，则打印“求解失败”，停止；否则

$$x_n = \frac{b_n}{a_{nn}}$$

$$\text{for } i = n-1 \text{ downto } 1 \text{ do}$$

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}$$

④ 打印$x_i, i = 1,2,\cdots,n$

2.3 全主元高斯消去法

① 输入$\boldsymbol{A}, \boldsymbol{b}$及$\epsilon$

② for $i = 1$ to n do $d_i = i$//记录未知量位置变化

③ 选主元消元：

for $k = 1$ to $n - 1$ do

- 求i_k, j_k使得$T = \left|a_{i_k,j_k}^{(k)}\right| = \max\limits_{k \le i, j \le n} \left|a_{ij}^{(k)}\right|$
- 若T $< \epsilon$，则打印“求解失败”，停止；否则
- 若$i \ne k$，则交换$\boldsymbol{A}$的第i_k行与第k行，交换b_{i_k}与b_k
- 若$j_k \ne k$，则交换$\boldsymbol{A}$的第j_k列与第k列，交换d_{j_k}与d_k

$$\text{for } i = k+1 \text{ to } n \text{ do}$$

$$T = \frac{a_{ik}}{a_{kk}};\ b_i = b_i - T \times b_k$$

$$\text{for } j = k+1 \text{ to } n \text{ do}$$

$$a_{ij} = a_{ij} - T \times a_{kj}$$

④ 回代

$$Z_n = \frac{b_n}{a_{nn}}$$

$$\text{for } i = n-1 \text{ downto } 1 \text{ do}$$

$$Z_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} Z_j}{a_{ii}}$$

⑤ $\text{for } j = 1 \text{ to } n \text{ do}$

$$x_{d_j} = Z_j$$

⑥ 打印$x_i, i = 1,2,\cdots,n$

① 输入$\boldsymbol{A}, \boldsymbol{b}$及$\epsilon$

② 对$k = 1,2,\cdots,n$循环

- 对$j = k, k+1,\cdots,n$循环

$$u_{kj} = a_{kj} - \sum_{m=1}^{k-1} l_{km}u_{mj}$$

如果$|u_{kk}| < \epsilon$，则打印“求解失败”，停止；否则

- 对$i = k+1,\cdots,n$计算

$$l_{ik} = \frac{a_{ik} - \sum_{m=1}^{k-1} l_{im}u_{mk}}{u_{kk}}$$

2.4 LU直接分解法

③ 求解$\boldsymbol{LY}=\boldsymbol{b}$

- $y_1=b_1$
- 对$i=2,3,\cdots,n$计算

$$y_i=b_i-\sum_{j=1}^{i-1}l_{ij}y_j$$

④ 求解$\boldsymbol{UX}=\boldsymbol{Y}$

- $x_n=\frac{y_n}{u_{nn}}$
- 对$i=n-1,n-2,\cdots,1$计算

$$x_i=\frac{\left(y_i-\sum_{j=i+1}^{n}u_{ij}x_j\right)}{u_{ii}}$$

⑤ 打印x_i

2.5 Jacobi迭代算法

① 输入$\boldsymbol{A}, \boldsymbol{b}$，初始向量$\boldsymbol{Y}$，容许误差$\epsilon$，容许最大迭代次数$M$

② 令$k = 1$

③ 形成迭代矩阵$\boldsymbol{B}$(存放在$\boldsymbol{A}$中)

- 对$i = 1,2,\cdots,n$循环
  - 若$|a_{ii}| < \epsilon$，则打印“求解失败”，停机；否则
  - $T = a_{ii}$
  - 对$j = 1,2,\cdots,n$计算

$$a_{ii} = -\frac{a_{ij}}{T}, a_{ii} = 0, g_i = \frac{b_i}{T}$$

④ 迭代:

- 对$i = 1,2,\cdots,n$计算

$$x_i = g_i + \sum_{j \neq i} a_{ij} y_j$$

⑤ 若$\|\boldsymbol{X} - \boldsymbol{Y}\| < \epsilon$，输出$\boldsymbol{X}, k$,停机；否则

⑥ 若$k < M$,则$k = k + 1$，将$\boldsymbol{X}$赋值给$\boldsymbol{Y}$,转④；否则，输出求解失败信息，停机

2.6 Seidel迭代算法

① 输入$\boldsymbol{A},\boldsymbol{b}$，初始向量$\boldsymbol{Y}$，容许误差$\epsilon$，容许最大迭代次数$M$

② 令$k=1$，$x_i=y_i$

③ 形成迭代矩阵$\boldsymbol{B}$(存放在$\boldsymbol{A}$中)

- 对$i=1,2,\cdots,n$循环
  - 若$|a_{ii}|<\epsilon$，则打印“求解失败”，停机；否则
  - $T=a_{ii}$
  - 对$j=1,2,\cdots,n$计算

$$a_{ii}=-\frac{a_{ij}}{T},a_{ii}=0,g_i=\frac{b_i}{T}$$

2.6 Seidel迭代算法

④ 迭代:

- 对$i = 1,2,\cdots,n$计算

$$x_i = g_i + \sum_{j \neq i} a_{ij} x_j$$

⑤ 若$\|\boldsymbol{X} - \boldsymbol{Y}\| < \epsilon$，输出$\boldsymbol{X}, k$,停机；否则

⑥ 若$k < M$,则$k = k + 1$，将$\boldsymbol{X}$赋值给$\boldsymbol{Y}$,转④；否则，输出求解失败信息，停机

2.7 松弛迭代算法

① 输入$\boldsymbol{A}, \boldsymbol{b}$，初始向量$\boldsymbol{Y}$，松弛因子$\omega$，容许误差$\epsilon$，容许最大迭代次数$M$

② 令$k = 1$，$x_i = y_i$

③ 形成迭代矩阵$\boldsymbol{B}$(存放在$\boldsymbol{A}$中)

- 对$i = 1,2,\cdots,n$循环
  - 若$|a_{ii}| < \epsilon$，则打印“求解失败”，停机；否则
  - $T = a_{ii}$
  - 对$j = 1,2,\cdots,n$计算

$$a_{ii} = -\omega\times\frac{a_{ij}}{T}, a_{ii} = 1 - \omega, g_i = \omega\times\frac{b_i}{T}$$

2.7 松弛迭代算法

④ 迭代：

- 对$i = 1,2,\cdots,n$计算

$$x_i = g_i + \sum_j a_{ij} x_j$$

⑤ 若$\|\boldsymbol{X} - \boldsymbol{Y}\| < \epsilon$，输出$\boldsymbol{X}, k$,停机；否则

⑥ 若$k < M$,则$k = k + 1$，将$\boldsymbol{X}$赋值给$\boldsymbol{Y}$,转④；否则，输出求解失败信息，停机

2.8 对分法求根

① 给出精度δ, ϵ令$a_0 = a, b_0 = b, k = 0$

② 令$x_k = (a_k + b_k)/2$,计算$f(x_k)$

③ 若$|f(x_k)| < \delta$,则x_k是$f(x) = 0$的根，停止计算，输出结果$\xi = x_k$

若$f(a_k) \cdot f(x_k) < 0$,则令$a_{k+1} = a_k, b_{k+1} = x_k$;

否则令$a_{k+1} = x_k, b_{k+1} = b_k$

④ 若$b_{k+1} - a_{k+1} \le \epsilon$,退出计算，输出结果$\xi = (a_{k+1} + b_{k+1})/2$;反之，令$k = k + 1$，返回②

2.9 松弛法迭代求根

① 给定初值x_0,精度ϵ，$k=0$

② 计算:

$$\omega_k=\frac{1}{1-\varphi'(x_k)}$$

$$x_{k+1}=(1-\omega_k)x_k+\omega_k\varphi(x_k)$$

③ 若$|x_{k+1}-x_k|<\epsilon$输出x_{k+1}；否则取$k=k+1$，返回②

2.10 牛顿法求根

① 给定初值x_0,精度ϵ，$k = 0$

② 计算

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

③ 若$|x_{k+1} - x_k| < \epsilon$输出$x_{k+1}$；否则取$k = k + 1$，返回②

非线性方程求根应用

- 参考文档 Lab2-GGD 完成任选一频率位置 (u, v) GGD 模型参数的估计。
  - 图像：4.2.04.tiff （Lena）
  - 只考虑Y通道（灰度图像）

- 模型和方法文献：

E. Y. Lam and J. W. Goodman, “A mathematical analysis of the DCT coefficient distributions for images,” *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1661–1666, 2000, doi: 10.1109/83.869177.

M. N. Do and M. Vetterli, “Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance,” *IEEE Transactions on Image Processing*, vol. 11, no. 2, pp. 146–158, Feb. 2002, doi: 10.1109/83.982822.


Thank You!