回顾

第二章 非线性方程$f(x)=0$的解法

2.1 引言

2.2 二分法与试值法

2.3 不动点迭代法 (收敛条件、收敛阶)

2.4 牛顿迭代法（迭代格式、收敛阶）

2.5 割线法（迭代格式、收敛阶）

2.6 迭代收敛的加速办法 (选讲)

非线性方程组的解法

$$
\begin{cases}
f_1\left(x_1, x_2, \cdots x_n\right) = 0 \\
f_2\left(x_1, x_2, \cdots x_n\right) = 0 \\
\vdots \\
f_n\left(x_1, x_2, \cdots x_n\right) = 0
\end{cases}
$$

第三章 线性方程组$Ax = b$的数值解法

3.1 引言

3.2 线性代数的基础知识

3.3 直接解法：

- 3.3.1 Gauss消去法
- 3.3.2 三角分解法
- 3.3.3 直接解法的误差分析

3.4 迭代解法

- 3.4.1 迭代法的基本概念
- 3.4.2 Jacobi迭代法
- 3.4.3 Gauss-Seidel迭代法
- 3.4.4 超松弛(SOR)迭代法（选讲）
- 3.4.5 共轭梯度法（选讲）

§ 3.1 引言

大量的科学与工程实际问题常常可以归结为求解含有多个未知量$x_1, x_2, \cdots, x_n$的线性代数方程组求解。

即求：

$$\begin{cases} \alpha_{11}x_1 + \alpha_{12}x_2 + \cdots + \alpha_{1n}x_n = \beta_1 \\ \alpha_{21}x_1 + \alpha_{22}x_2 + \cdots + \alpha_{2n}x_n = \beta_2 \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ \alpha_{n1}x_1 + \alpha_{n2}x_2 + \cdots + \alpha_{nn}x_n = \beta_n \end{cases} \tag{3.1}$$

可以写为矩阵形式

$$\mathbf{Ax} = \mathbf{b}$$

其中

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} .$$

解线性代数方程组的有效方法在计算数学和科学计算中具有特殊的地位和作用，可有效解决如弹性力学、电路分析、热传导和振动、以及社会科学及定量分析商业经济中的各种问题。

精确求解方法

方法1

举例说明

$$Ax = b \;\Rightarrow\; A^{-1}Ax = A^{-1}b \;\Rightarrow\; x = A^{-1}b$$

计算量为矩阵求逆

矩阵求逆的方法:初等行变换法, 伴随矩阵法, 高斯-约当法

方法2 Crammer法则

$$\boxed{\det(A) \neq 0}$$

$$x_i = \frac{|A_i|}{|A|} \quad i = 1, 2, \ldots, n$$

其中 $|A|$是方程组系数矩阵对应的行列式

$|A_i|$是以右端变量向量b替代A的第i列所得矩阵的行列式

$$x_i = \frac{\begin{vmatrix} a_{11} & \cdots & a_{1i-1} & b_1 & a_{1i+1} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2i-1} & b_2 & a_{2i+1} & \cdots & a_{2n} \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & \cdots & a_{ni-1} & b_n & a_{ni+1} & \cdots & a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}}$$

§ 3.1 引言

运用Crammer法则，计算一个n阶行列式需要做$(n-1)(n!)$次个乘法，求解上述方程**所需乘除法的运算量大约为**

$$x_i = \frac{\begin{vmatrix} a_{11} & \cdots & a_{1i-1} & b_1 & a_{1i+1} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2i-1} & b_2 & a_{2i+1} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & \cdots & a_{ni-1} & b_n & a_{ni+1} & \cdots & a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}}$$

$$\mathbf{N=(n+1)\times(n-1)(n!)+n}$$

因此,当线性方程组的阶数n较高时,

计算量太大，现实上不可行，

例如, n=20时，N≈9.7×1020，如果采用每秒十亿次的个人计算机，按每天工作24小时，大约需要3万年。因此，需要采用实用的数值计算方法来求解。

- **快速、高效地数值求解线性方程组是数值线性代数研究中的核心问题，也是目前科学计算中的重大研究课题之一。**

线性方程组的数值解法有：直接法和迭代法。

直接法：只包含有限次四则运算。若在计算过程中都不发生舍入误差的假定下，计算结果就是原方程组的精确解。包括

- ✓ Gauss消元法
- ✓ 三角分解法

迭代法：把方程组的解向量看作是某种极限过程的极限，从一个初始向量出发，按照一定的迭代格式，构造出一个趋向于真解的无穷序列。

- ✓ Jacobi迭代法
- ✓ Gauss-Seidel迭代法
- ✓ 超松弛(SOR)迭代法
- ✓ 共轭梯度法

Remark：由于运算过程中舍入误差的存在，实际上直接方法得到的解也是方程组的近始解。

§3.2 线性代数的基础知识

一个N维实数向量*x*是n个实数的有序集合，通常写成坐标形式

$$x = (x_1, x_2, \cdots, x_n)$$

其中 $x_1, x_2, \cdots, x_n$ 称为*x*的坐标或分量。

向量运算：

相等、和、取负、差、标量乘积cX, 线性组合

➢常用的几种向量范数：

设 $x = (x_1, x_2, \cdots, x_n)^T$

❶ 1-范数：$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

❷ 2-范数：$$\|x\|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2} = \sqrt{(x, x)}$$

❸ ∞-范数：$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|$$

上述3种向量范数统称为P-范数(或者Holder范数)

$$\boxed{\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \quad 1 \le p < \infty}$$

➢三个重要不等式

1 三角不等式 $\forall x, y \in R^n \quad \big|\|x\| - \|y\|\big| \le \|x - y\|$

证明：$\|x\| = \|x - y + y\| \le \|x - y\| + \|y\|$

$$\|x\| - \|y\| \le \|x - y\| \qquad \|x\| - \|y\| \ge -\|y - x\|$$

同理 $\|y\| = \|y - x + x\| \le \|y - x\| + \|x\|$

2 闵可夫斯基(Minkowski)不等式：

$$\left(\sum_{i=1}^{n} |x_i + y_i|^p\right)^{1/p} \le \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p} + \left(\sum_{i=1}^{n} |y_i|^p\right)^{1/p}$$

3 柯西-施瓦茨(Cauchy-Schwartz)不等式:

$$(x,y)^2 \le (x,x)(y,y) \quad \forall x,y \in R^n$$

或者

$$\left|\sum_{i=1}^{n} x_i y_i\right| \le \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2} \left(\sum_{i=1}^{n} |y_i|^2\right)^{1/2}$$

证明

令 $u = y - \dfrac{(x,y)}{\|x\|_2^2} x$ 由于 $\boxed{(u,x) = 0}$

可知 $\|u\|_2^2 = (u,u) = (u, y - \dfrac{(x,y)}{\|x\|_2^2} x) = (u,y)$

≥ 0

$$= (y,y) - \frac{(x,y)^2}{\|x\|_2^2} = \|y\|_2^2 - \frac{(x,y)^2}{\|x\|_2^2}$$

因此,

$$(x,y)^2 \le \|x\|_2^2 \|y\|_2^2 = (x,x)(y,y)$$

§ 3.2 线性代数的基础知识

一个矩阵是数字按行列分布的矩形数组。一个矩阵有M行和N列，称为$M \times N$矩阵。大写字母A表示矩阵，小写带下标字母a_{ij}表示构成矩阵的一个数。矩阵可表示为

$$A = [a_{ij}]_{M\times N}, \quad 1 \le i \le M, 1 \le j \le N$$

这里a_{ij}表示位于（i, j）的数。

矩阵运算：

相等、和、取负、差、标量乘积cX, 线性组合

$$cA = [ca_{ij}]_{M\times N}, \quad 1 \le i \le M, 1 \le j \le N$$

$$pA + qB = [pa_{ij} + qb_{ij}]_{M\times N}, \quad 1 \le i \le M, 1 \le j \le N$$

矩阵乘

设 c 是一个标量，A，B 和 C 是矩阵，而且对应的矩阵加法和乘法有定义，则

$(AB)C = A(BC)$	矩阵乘的结合律	(12)
$IA = AI = A$	单位矩阵	(13)
$A(B+C) = AB + AC$	左分配律	(14)
$(A+B)C = AC + BC$	右分配律	(15)
$c(AB) = (cA)B = A(cB)$	标量结合律	(16)

矩阵满足分配率、结合率，但不满足交换率。

$$A = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$AB = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \qquad BA = \begin{pmatrix} 2 & 2 \\ -2 & -2 \end{pmatrix}$$

行列式

方阵 A 的行列式是一个标量值(实数),表示为 $\det(A)$或$|A|$。如果 A 是 $N\times N$ 矩阵

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}$$

则 A 的行列式表示为

$$\det(A)=\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{vmatrix}$$

尽管行列式的表示看起来像一个矩阵,但它的性质完全不同,行列式是一个标量值(实数)。

如果 $A=[a_{ij}]$是 1×1 矩阵,定义 $\det(A)=a_{11}$。如果 $A=[a_{ij}]_{N\times N}$,其中 $N\geqslant 2$,则让 M_{ij} 为A 的$(N-1)\times(N-1)$子矩阵的行列式,子矩阵是通过去掉矩阵 A 的第 i 行和第 j 列构成的。行列式 M_{ij}称为 a_{ij}的**余子式**。A_{ij}定义为 $A_{ij}=(-1)^{i+j}M_{ij}$,称为 a_{ij}的代数余子式。这样 $N\times N$ 矩阵A 的行列式表示为

$$\det(A)=\sum_{j=1}^{N} a_{ij}A_{ij} \qquad (\text{第 } i \text{ 行扩展}) \tag{19}$$

或

$$\det(A)=\sum_{i=1}^{N} a_{ij}A_{ij} \qquad (\text{第 } j \text{ 列扩展}) \tag{20}$$

Matlab实现

MATLAB 函数 det(A)和 inv(A)分别用来计算方阵 A 的行列式和逆(如果 A 是可逆的)。

例 3.11 使用 MATLAB 和推论(25)中的逆矩阵法,分别求解例 3.6 中的线性方程组。

解:首先通过证明 $\det(A)\neq 0$(参见定理 3.4),验证 A 是非奇异矩阵。

```
>>A=[0.125 0.200 0.400;0.375 0.500 0.600;0.500 0.300 0.000];
>>det(A)
ans=
   -0.0175
```

然后根据推论(25),可得到 $AX=B$ 的解是 $AX=B$, $X=A^{-1}B$。

```
>>X=inv(A)*[2.3 4.8 2.9]'
X=
   4.0000
   3.0000
   3.0000
```

可通过检查 $AX=B$ 来验证此结果。

```
>>B=A*X
B=
   2.3000
   4.8000
   2.9000
```

■

根据向量的1、2和∞范数，可得到如下3种常用的矩阵范数

记 $A=(a_{ij})_{n\times n}$

❶1范数：$\|A\|_1=\max\limits_{1\le j\le n}\sum\limits_{i=1}^{n}|a_{ij}|$ 列和范数

❷ ∞范数：$\|A\|_\infty=\max\limits_{1\le i\le n}\sum\limits_{j=1}^{n}|a_{ij}|$ 行和范数

❸2范数：

$$\|A\|_2=\sqrt{\lambda_1}=\left[\rho(A^TA)\right]^{1/2}$$

其中λ_1是A^TA 的最大特征值

$$\boxed{\rho(A)=\max\limits_{1\le i\le n}|\lambda_i|}$$

谱半径

例3.1： 给定矩阵

$$\boldsymbol{A}=\begin{bmatrix}2 & 0 & 0\\ 1 & 3 & 1\\ 0 & 1 & 2\end{bmatrix}$$

$$\boldsymbol{A}^T=\begin{bmatrix}2 & 1 & 0\\ 0 & 3 & 1\\ 0 & 1 & 2\end{bmatrix}$$

求矩阵$\boldsymbol{A}$的1、2、∞ 范数。

$$\|\boldsymbol{A}\|_1=4 \qquad \|\boldsymbol{A}\|_\infty=5 \qquad \|\boldsymbol{A}\|_2=3.759$$

若$\boldsymbol{A}$是实对称矩阵，则 $\boldsymbol{\rho(A)}=\|\boldsymbol{A}\|_2$

3.3 直接解法--Gauss消去法

研究求解有N个方程和N个未知数的一般方程组Ax=b，目标是运用初等变换构造一个等价的上三角方程组Ux=y.

如果两个$N \times N$线性方程组的解相同，那么二者等价。根据线性代数中的定理可知，对一个给定方程组进行一定的变换，不能改变它的解。

注意：行变换和列变换不能同时执行

初等变换

- Interchanges（对调）交换：对调方程组的两行；
- Scaling（比例）交换：用非零常数乘以方程组的某一行；
- Replacement（置换）交换：将方程组的某一行乘以一个非零常数，再加到另一行上。

3.3.1 Gauss消去法

转化为回代算法

$$\begin{cases} 2x_1 + x_2 = 3 \\ x_1 - x_2 = 0 \end{cases} \Rightarrow \begin{cases} 2x_1 + \quad x_2 = \quad 3 \\ \qquad -\dfrac{3}{2}x_2 = -\dfrac{3}{2} \end{cases} \begin{matrix} \Rightarrow \text{从第二个方程解出} x_2 = 1 \\ \Rightarrow \text{代入第一个方程, 得到} x_1 = 1 \end{matrix}$$

消去法的思想

1.将n元方程组的n-1个方程通过“消元”，形成一个与原方程等价的新方程组

2.继续将n-1个方程通过“消元”，形成一个与之等价的新方程组

3.直到最后一个方程为一元一次方程为止

4.从最后一个方程中解出最后一个未知量，然后回代得到其它的解

消去法的基本步骤：消去、回代

3.3.1 Gauss消去法过程

方程组 $\boldsymbol{Ax=b}$ 的增广矩阵记为:

$$(A^{(0)} \quad b^{(0)}) = \begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} & \beta_1^{(0)} \\ a_{21}^{(0)} & a_{22}^{(0)} & a_{23}^{(0)} & \cdots & a_{2n}^{(0)} & \beta_2^{(0)} \\ a_{31}^{(0)} & a_{32}^{(0)} & a_{33}^{(0)} & \cdots & a_{3n}^{(0)} & \beta_3^{(0)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1}^{(0)} & a_{n2}^{(0)} & a_{n3}^{(0)} & \cdots & a_{nn}^{(0)} & \beta_n^{(0)} \end{pmatrix}$$

将矩阵的第i行分别减去第一行的倍数$l_{i1} = \dfrac{a_{i1}^{(0)}}{a_{11}^{(0)}}, i = 2,3,\ldots,n,$ 得到

$$(A^{(1)} \quad b^{(1)}) = \begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} & \beta_1^{(0)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & \beta_2^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} & \beta_3^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} & \beta_n^{(1)} \end{pmatrix}$$

其中

$$a_{ij}^{(1)} = a_{ij}^{(0)} - l_{i1}a_{1j}^{(0)}$$

$$\beta_i^{(1)} = \beta_i^{(0)} - l_{i1}\beta_1^{(0)}$$

$$j = 2,3,\ldots,n$$

$$i = 2,3,\ldots,n$$

3.3.1 Gauss消去法过程

$$
(A^{(k)} \quad b^{(k)}) = \begin{pmatrix}
a_{11}^{(0)} & \cdots & a_{1k}^{(0)} & a_{1,k+1}^{(0)} & \cdots & a_{1n}^{(0)} & \beta_1^{(0)} \\
 & \ddots & \vdots & \vdots & & \vdots & \vdots \\
 & & a_{kk}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{kn}^{(k-1)} & \beta_k^{(k-1)} \\
 & & & a_{k+1,k+1}^{(0)} & \cdots & a_{k+1,n}^{(k)} & \beta_{k+1}^{(k)} \\
 & & & \vdots & & \vdots & \vdots \\
 & & & a_{n,k+1}^{(k)} & \cdots & a_{nn}^{(k)} & \beta_n^{(k)}
\end{pmatrix}
$$

计算关系式

$$l_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}$$

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - l_{ik} a_{kj}^{(k-1)}, \; j = k+1, \dots, n$$

$$\beta_i^{(k)} = \beta_i^{(k-1)} - l_{ik} \beta_k^{(k-1)}, i = k+1, k+2, \dots, n$$

3.3.1 Gauss消去法过程

$$
\left(A^{(n-1)} \quad b^{(n-1)}\right) = \begin{pmatrix}
a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} & \beta_1^{(0)} \\
 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & \beta_2^{(1)} \\
 & & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & \beta_3^{(2)} \\
 & & & \ddots & \vdots & \vdots \\
 & & & & a_{nn}^{(n-1)} & \beta_n^{(n-1)}
\end{pmatrix}
$$

回代

$$
\begin{cases}
x_n = \dfrac{\beta_n^{(n-1)}}{a_{nn}^{(n-1)}} \\
x_k = \beta_k^{(k-1)} - \displaystyle\sum_{j=k+1}^{n} a_{kj}^{(k-1)} x_j, k = n-1, n-2, \ldots, 1
\end{cases}
$$

算法: Gauss消元法

求方程组$Ax=b$ 的解.

输入:增广矩阵$A_{n\times(n+1)}=(A|b)$.

输出: 近似解 $x_k=a_{k,n+1}(k=1,2,\cdots,n)$ 或失败信息.

消元过程

for k = 1,2,…,n-1 do ***Step 1 - Step 4*** N-1次

Step 1 寻找行号 i_k, 使得 $\left|a_{i_k,k}\right|=\max\limits_{k\le i\le n}\left|a_{i,k}\right|, i=k,k+1,\cdots,n$

Step 2 如果 $a_{i_k,k}\neq 0$, 则交换第k行和i_k行;

否则转*Step 7*

算法: Gauss消元法（续）

Step 3 for i=k+1,…,n 计算 $l_{ik} = a_{ik} / a_{kk}$ N-k次

Step 4 for j=k+1,…,n+1 计算 $a_{ij} = a_{ij} - l_{ik} a_{kj}$ N-k+1次

回代过程

Step 5 $x_n = a_{n,n+1} / a_{n,n}$

n-i+1次

Step 6 for i=n-1,…,1 计算 $x_i = (a_{i,n+1} - \sum_{j=i+1}^{n} a_{i,j} x_j) / a_{i,i}$

Step 7 Output (系数矩阵奇异); /*不成功 */ STOP.

高斯消去法运算量估计

1.消去算法运算量

分为$n-1$步,第k步变换$n-k$行:求倍数,再从$n+1-k$个元素中减去第k行对应列的倍数　**因此，所需要的乘除次数为：**

$$N_1=\sum_{k=1}^{n-1}(n-k)(n-k+1)=\frac{n^3}{3}-\frac{n}{3}$$

已知 $1^2+2^2+3^2+\cdots+n^2=\frac{n(n+1)(2n+1)}{6}$

2.回代运算量

求x_n需做1次除法,求x_{n-1}需做1次乘法和1次除法,..., 求x_1需$n-1$次乘法和1次除法,因此所需乘除次数:

$$N_2=1+2+...+n=\frac{n(n+1)}{2}$$

因此, $N=N_1+N_2=\frac{n^3}{3}+\frac{n^2}{2}-\frac{n}{6}$，即, 运算量为$o(n^3)$

例3.2

利用高斯消去法求解方程组:

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 12 \\ 12x_1 - 8x_2 + 6x_3 + 10x_4 = 34 \\ 3x_1 - 13x_2 + 9x_3 + 3x_4 = 27 \\ -6x_1 + 4x_2 + x_3 - 18x_4 = -38 \end{cases}$$

解:

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 12 \\ 12x_1 - 8x_2 + 6x_3 + 10x_4 = 34 \\ 3x_1 - 13x_2 + 9x_3 + 3x_4 = 27 \\ -6x_1 + 4x_2 + x_3 - 18x_4 = -38 \end{cases}$$

利用 $r_i - \frac{\alpha_{i1}}{\alpha_{11}} r_1$ ，i=2, 3, 4. 得

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 12 \\ -4x_2 + 2x_3 + 2x_4 = 10 \\ -12x_2 + 8x_3 + x_4 = 21 \\ 2x_2 + 3x_3 - 14x_4 = -26 \end{cases}$$

利用 $r_i - \frac{\alpha_{i2}^{(2)}}{\alpha_{22}^{(2)}} r_2$ ，i=3, 4. 得

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 12 \\ -4x_2 + 2x_3 + 2x_4 = 10 \\ 2x_3 - 5x_4 = -9 \\ 4x_3 - 13x_4 = -21 \end{cases}$$

利用 $r_i - \frac{\alpha_{i3}^{(3)}}{\alpha_{33}^{(3)}} r_3$ ，i=4. 得

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 12 \\ -4x_2 + 2x_3 + 2x_4 = 10 \\ 2x_3 - 5x_4 = -9 \\ -3x_4 = -3 \end{cases}$$

回代，可得准确解为

$$[x_1, x_2, x_3, x_4] = [1, -3, -2, 1]^T$$

选主元以减少误差

由于计算机使固定精度计算，这样在每次算术计算中可能引入微小的误差。

例3.3： 值$x_1 = x_2 = 1.000$是如下方程组的解

$$
\begin{aligned}
1.133x_1 + 5.281x_2 &= 6.414 \\
24.14x_1 - 1.210x_2 &= 22.93
\end{aligned}
$$

使用4位有效数字精度求解其近似值。

解：第 2 行减去第 1 行乘以倍数 $m_{21} = 24.14/1.133 = 21.31$，得到上三角线性方程组。使用 4 位有效数字精度计算，可得到新的系数，如下所示：

$$
\begin{aligned}
a_{22}^{(2)} &= -1.210 - 21.31 \times 5.281 = -1.210 - 112.5 = -113.7 \\
a_{23}^{(2)} &= 22.93 - 21.31 \times 6.414 = 22.93 - 136.7 = -113.8
\end{aligned}
$$

计算后的上三角线性方程组为

$$
\begin{aligned}
1.133x_1 + 5.281x_2 &= 6.414 \\
-113.7x_2 &= -113.8
\end{aligned}
$$

利用回代法可得 $x_2 = -113.8/(-113.7) = 1.001$ 和 $x_1 = (6.414 - 5.28 \times 1.001)/(1.133) = (6.414 - 5.286)/1.133 = 0.9956$。 ■

该误差是由于倍数$m_{21}=21.31$的值。为改善该不足，尝试交换上述方程的第一行和第二行，来减少m_{21}的值。

使用4位有效数字精度计算和高斯消去法求解如下方程组

$$
\begin{aligned}
24.14x_1 - 1.210x_2 &= 22.93\\
1.133x_1 + 5.281x_2 &= 6.414
\end{aligned}
$$

解：这次用第2行减去第1行乘以倍数 $m_{21}=1.133/24.14=0.04693$。新的系数为

$$
\begin{aligned}
a_{22}^{(2)} &= 5.281 - 0.04693\times(-1.210) = 5.281 + 0.05679 = 5.338\\
a_{23}^{(2)} &= 6.414 - 0.04693\times 22.93 = 6.414 - 1.076 \quad = 5.338
\end{aligned}
$$

计算后的上三角线性方程组为

$$
\begin{aligned}
24.14x_1 - 1.210x_2 &= 22.93\\
5.338x_2 &= 5.338
\end{aligned}
$$

利用回代法可得 $x_2=5.338/5.338=1.000$ 和 $x_1=(22.93+1.210\times 1.000)/24.14=1.000$。

思想

选主元策略的目的在于：每次消元之前，在剩余元素中选择绝对值最大的非零元素作为主元，然后经过换行换到主对角线上，进而消去列中的剩余元素。

算法: Gauss选主元消去算法

求方程组 Ax=b 的解.

输入:增广矩阵 $A_{n\times(n+1)}$=（$A|b$）.

输出: 近似解 $x_k=a_{k,n+1}$(k=1,2,…,n) 或失败信息.

消元过程

for k = 1,2,…,n-1 do *Step 1 - Step 4*

Step 1 寻找行号 i_k , 使得

Step 2 如果 $a_{i_k,k} \neq 0$， $\left|a_{i_k,k}\right| = \max\limits_{k\le i\le n}\left|a_{i,k}\right|, i = k, k+1, \cdots, n$

则交换第k行和i_k行；否则转***Step 7***

算法: Gauss列主元消去算法（续）

Step 3 **for i=k+1,…,n** 计算 $l_{ik} = {a_{ik}}/{a_{kk}}$

Step 4 **for j=k+1,…,n+1** 计算 $a_{ij} = a_{ij} - l_{ik} a_{kj}$

回代过程

Step 5 $x_n = {a_{n,n+1}}/{a_{n,n}}$

Step 6 **for i=n-1,…,1** 计算 $x_i = (a_{i,n+1} - \sum_{j=i+1}^{n} a_{i,j} x_j) / a_{i,i}$

Step 7 **Output (系数矩阵奇异); /*不成功 */ STOP.**

Matlab源程序：GaussXuanzhuyuan.m

Matlab源程序:

GaussXuanzhuyuan.m

```
function X = GaussXuanzhuyuan(A,B)

%Input    - A is an N x N nonsingular matrix
%         - B is an N x 1 matrix
%Output - X is an N x 1 matrix containing the
% solution to AX=B.
%Initialize X and the temporary storage matrix C
[N N]=size(A);
X=zeros(N,1);
C=zeros(1,N+1);

%Form the augmented matrix: Aug=[A|B]
Aug=[A B];

for p=1:N-1
   %Partial pivoting for column p
   [Y,j]=max(abs(Aug(p:N,p)));
   %Interchange row p and j
   C=Aug(p,:);
   Aug(p,:)=Aug(j+p-1,:);
   Aug(j+p-1,:)=C;
  if Aug(p,p)==0
     'A was singular.  No unique solution'
    break
  end
  %Elimination process for column p
  for k=p+1:N
    m=Aug(k,p)/Aug(p,p);
    Aug(k,p:N+1)=Aug(k,p:N+1)-
m*Aug(p,p:N+1);
  end
end

%Back Substitution on [U|Y] using
Program 3.1
X=backsub(Aug(1:N,1:N),Aug(1:N,N+1));
```

作业3.1

求解方程组

1. $$\begin{aligned} 2x_1+4x_2-6x_3&=-4\\ x_1+5x_2+3x_3&=10\\ x_1+3x_2+2x_3&=5 \end{aligned}$$

$$\begin{aligned} 2x_1+4x_2-6x_3&=-4\\ 3x_2+6x_3&=12\\ 3x_3&=3 \end{aligned}$$

6. 求解抛物线 $y=A+Bx+Cx^2$ 的参数，抛物线经过点(1,6)，(2,5)和(3,2)。

算法与程序

GaussXuanzhuyuan.m

2. 使用程序 3.2 求 6 次多项式 $y=a_1+a_2x+a_3x^2+a_4x^3+a_5x^4+a_6x^5+a_7x^6$ 的系数，它经过点(0,1)，(1,3)，(2,2)，(3,1)，(4,3)，(5,2)，(6,1)。使用 plot 命令画出多项式，标出给出的经过点，并解释图中的误差。

回顾

第三章 线性方程组$Ax=b$的数值解法

3.1 引言
3.2 线性代数的基础知识
3.3 直接解法：
  3.3.1 Gauss消去法
  3.3.2 三角分解法
  3.3.3 直接解法的误差分析
3.4 迭代解法
  3.4.1 迭代法的基本概念
  3.4.2 Jacobi迭代法
  3.4.3 Gauss-Seidel迭代法
  3.4.4 超松弛(SOR)迭代法（选讲）
  3.4.5 共轭梯度法（选讲）

§ 3.4.1 迭代法的基本概念

➢回顾向量范数：

设 $x=(x_1,x_2,\cdots,x_n)^T$

❶ 1-范数：$\|x\|_1=\sum_{i=1}^{n}|x_i|$

❷ 2-范数：$\|x\|_2=\left(\sum_{i=1}^{n}x_i^2\right)^{1/2}=\sqrt{(x,x)}$

❸ ∞-范数：$\|x\|_\infty=\max_{1\le i\le n}|x_i|$

上述定义说明：根据向量范数的等价性，如果向量序列收敛，则对任何一种向量范数而言均收敛。

§ 3.4.1 迭代法的基本概念

回顾矩阵范数

记 $A=(a_{ij})_{n\times n}$

❶1范数: $\|A\|_1=\max\limits_{1\le j\le n}\sum\limits_{i=1}^{n}|a_{ij}|$ 列和范数

❷ ∞ 范数: $\|A\|_\infty=\max\limits_{1\le i\le n}\sum\limits_{j=1}^{n}|a_{ij}|$ 行和范数

❸2范数:

$$\|A\|_2=\sqrt{\lambda_1}=\left[\rho(A^TA)\right]^{1/2}$$

$$\boxed{\rho(A)=\max_{1\le i\le n}|\lambda_i|}$$

其中λ_1是A^TA 的最大特征值

谱半径

§ 3.3.2 三角分解法 （LU分解）

它是基本Gauss消元法的一种等价变形

在3.3.1节可以看到，求解上三角矩阵方程组很容易。现在介绍将给定矩阵A分解为下三角矩阵L和上三角矩阵U的乘积的概念，其中下三角矩阵L的主对角线为1，上三角矩阵U的对角线元素非零。

定义 3.4 如果非奇异矩阵A可表示为下三角矩阵L和上三角矩阵U的乘积

$$A = LU$$

则A存在一个三角分解。

如果 $|A| \neq 0$ 可三角分解，则4×4维矩阵表示如下，其中 $u_{kk} \neq 0$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

定理3.1 如果非奇异矩阵$A \in R^{n \times n}$存在三角分解，即存在矩阵L和U满足

$$A = LU$$

则有

$$\begin{cases} Ly = b \\ Ux = y \end{cases}$$

下面讨论如何得到矩阵的三角分解。

构造下列矩阵的三角分解：

$$\boldsymbol{A} = \begin{bmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{bmatrix}$$

解:通过将单位矩阵放在 A 的左边来构造矩阵 L。对每个用来构造上三角矩阵的行变换,将倍数 m_{ij} 放在左边的对应位置。初始矩阵为

$$A=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}4 & 3 & -1\\ -2 & -4 & 5\\ 1 & 2 & 6\end{bmatrix}$$

用第 1 行消去矩阵 A 的第 1 列中 a_{11} 下面的元素。第 2 行和第 3 行分别减去第 1 行乘以倍数 $m_{21}=-0.5$和 $m_{31}=0.25$。将倍数放到矩阵的左边相应位置,结果为

$$A=\begin{bmatrix}1 & 0 & 0\\ -0.5 & 1 & 0\\ 0.25 & 0 & 1\end{bmatrix}\begin{bmatrix}4 & 3 & -1\\ 0 & -2.5 & 4.5\\ 0 & 1.25 & 6.25\end{bmatrix}$$

用第 2 行消去第 2 列中对角线下方的元素。第 3 行减去第 2 行乘以倍数 $m_{32}=-0.5$,再将倍数放入矩阵左边,则可得到矩阵 A 的三角分解:

$$A=\begin{bmatrix}1 & 0 & 0\\ -0.5 & 1 & 0\\ 0.25 & -0.5 & 1\end{bmatrix}\begin{bmatrix}4 & 3 & -1\\ 0 & -2.5 & 4.5\\ 0 & 0 & 8.5\end{bmatrix} \tag{8}$$

L　　　　U

带状矩阵的分解

定义3.1 设$A=(\alpha_{ij})$是n阶方阵,对小于n的正整数p和q,当 $j>i+q$ 或 $i>j+p$时,有$\alpha_{ij}=0$,则称A是具有上带宽q和下带宽p的带状矩阵.

以带状矩阵为系数矩阵的方程组称为带状方程组.

当$p=q=1$时称为三对角阵

$$T=\begin{pmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & a_3 & b_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & a_n & b_n \end{pmatrix}$$

定理3.2 设n阶矩阵A有LU分解,$A=LU$,如果A是上下带宽分别为q和p的带状矩阵,则L是带宽为p的下三角 阵,U是带宽为q的上三角阵

由定理3.2知，存在分解T=LU

其中
$$L=\begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_n & 1 \end{pmatrix} \quad U=\begin{pmatrix} \mu_1 & r_1 & & & \\ & \mu_2 & r_2 & & \\ & & \ddots & \ddots & \\ & & & \mu_{n-1} & r_{n-1} \\ & & & & \mu_n \end{pmatrix}$$

求解三对角方程组 Tx=d, 其中 $d=(d_1,d_2\cdots,d_n)^T$

由$A=LU$可知,将原方程分解为两个方程　$Ly=d$　$Ux=y$

得到计算x和y的公式

$$\begin{cases} y_1=d_1 \\ y_k=d_k-l_k y_{k-1}, k=2,3,\dots,n \end{cases}$$

追的过程

$$\begin{cases} x_1=y_n/\mu_n \\ x_k=\dfrac{1}{\mu_k}(y_k-r_k x_{k+1}), k=n-1,n-2,\dots,2,1 \end{cases}$$

赶的过程

以上解法称为追赶法

然而，存在非奇异矩阵A不能直接进行三角分解。

如下面的例3.4

$$A=\begin{bmatrix}1 & 2 & 6\\ 4 & 8 & -1\\ -2 & 3 & 5\end{bmatrix}$$

证明：设 A 存在一个直接 LU 分解，则

$$\begin{bmatrix}1 & 2 & 6\\ 4 & 8 & -1\\ -2 & 3 & 5\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ m_{21} & 1 & 0\\ m_{31} & m_{32} & 1\end{bmatrix}\begin{bmatrix}u_{11} & u_{12} & u_{13}\\ 0 & u_{22} & u_{23}\\ 0 & 0 & u_{33}\end{bmatrix}$$

上式中右边的矩阵L和U相乘与对应的矩阵A的元素进行比较

$$\begin{cases}u_{11}=1;\ u_{12}=2,\ u_{13}=6;\\ m_{21}u_{11}=m_{21}=4,\\ m_{21}u_{12}+u_{22}=8\Rightarrow u_{22}=0,\\ m_{21}u_{13}+u_{23}=-1\Rightarrow u_{23}=-25;\\ m_{31}u_{11}=m_{31}=-2,\\ m_{31}u_{12}+m_{32}u_{22}=3\Rightarrow -4=3\quad \text{error},\\ m_{31}u_{13}+m_{32}u_{23}+u_{33}=5;\end{cases}$$

引入如下置换矩阵的概念

定义 3.2

$N\times N$ **置换矩阵** P 是在每一行和每一列只有一个元素为 1，而其他元素为 0 的矩阵。

$P=[p_{ij}]$的元素有如下形式：

$$p_{ij}=\begin{cases}1 & j=k_i\\ 0 & \text{其他}\end{cases}$$

例如，下列 4×4 矩阵是一个置换矩阵：

$$P=\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$$

定理3.3 如果 P 是一个置换矩阵，则它是非奇异的，且 $P^{-1}=P'$。

定理3.4 如果 A 是非奇异矩阵，则存在一个置换矩阵 P，使得 PA 存在三角分解

$$PA=LU$$

定理的证明可参见高级线性代数教材。

如果将例3.4中的第2行和第3行进行交换，则得到的PA有一个三角分解. 此时，P= $\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}$ 计算PA 的乘积可得

$$\boldsymbol{PA}=\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}\begin{bmatrix}1&2&6\\4&8&-1\\-2&3&5\end{bmatrix}=\begin{bmatrix}1&2&6\\-2&3&5\\4&8&-1\end{bmatrix}$$

对PA进行LU分解，可得

$$PA=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}1&2&6\\-2&3&5\\4&8&-1\end{pmatrix}$$

$$=\begin{pmatrix}1&0&0\\2&1&0\\-4&0&1\end{pmatrix}\begin{pmatrix}1&2&6\\0&7&17\\0&0&-25\end{pmatrix}=LU$$

Matlab实现

Matlab命令 [L,U,P]=lu(A)可得到下三角矩阵L，上三角矩阵U和上述定理中的置换矩阵P

```
>>A=[1 2 6 ;4 8 -1;-2 3 -5];
>>[L,U,P]=lu(A)
L=
     1.0000   0        0
    -0.5000  1.0000   0
     0.2500   0        1.0000
U=
     4.0000  8.0000  -1.0000
     0        7.0000  4.5000
     0        0        6.2500
P=
    0 1 0
    0 0 1
    1 0 0

>>inv(P)*L*U
    1   2  6
    4   8 -1
    -2  3  5
```

Matlab源程序：

Sanjiaofenjiefa.m

Matlab源程序： Sanjiaofenjiefa.m

```
function X = Sanjiaofenjiefa(A,B)
%Input    - A is an N x N matrix
%           - B is an N x 1 matrix
%Output - X is an N x 1 matrix
%containing the solution to AX = B.
[N,N]=size(A);
X=zeros(N,1);   Y=zeros(N,1);
C=zeros(1,N);  R=1:N;
for p=1:N-1
   %Find the pivot row for column p
   [max1,j]=max(abs(A(p:N,p)));
   %Interchange row p and j
      C=A(p,:);
      A(p,:)=A(j+p-1,:);
      A(j+p-1,:)=C;
      d=R(p);
      R(p)=R(j+p-1);
      R(j+p-1)=d;
 if A(p,p)==0
      'A is singular.  No unique solution'
      break
   end
%Calculate multiplier and place in
%subdiagonal portion of A
      for k=p+1:N
         mult=A(k,p)/A(p,p);
     A(k,p) = mult;
         A(k,p+1:N)=A(k,p+1:N)-
mult*A(p,p+1:N);
      end
end

%Solve for Y
Y(1) = B(R(1));
for k=2:N
   Y(k)= B(R(k))-A(k,1:k-1)*Y(1:k-1);
end

%Solve for X
X(N)=Y(N)/A(N,N);
for k=N-1:-1:1
   X(k)=(Y(k)-A(k,k+1:N)*X(k+1:N))/A(k,k);
end
```

作业3.2

3. 对下列矩阵求解它的三角分解 $\boldsymbol{L}$ 和 $\boldsymbol{U}$。

(a) $\begin{bmatrix} -5 & 2 & -1 \\ 1 & 0 & 3 \\ 3 & 1 & 6 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 0 & 3 \\ 3 & 1 & 6 \\ -5 & 2 & -1 \end{bmatrix}$

4. 对下列矩阵求解它的三角分解 $\boldsymbol{L}$ 和 $\boldsymbol{U}$。

(a) $\begin{bmatrix} 4 & 2 & 1 \\ 2 & 5 & -2 \\ 1 & -2 & 7 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & -2 & 7 \\ 4 & 2 & 1 \\ 2 & 5 & -2 \end{bmatrix}$

算法与程序

1. 使用程序 3.3 求解线性方程组 $\boldsymbol{AX} = \boldsymbol{B}$，其中

$$\boldsymbol{A} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & -1 & 3 & 5 \\ 0 & 0 & 2 & 5 \\ -2 & -6 & -3 & 1 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

使用 MATLAB 中的`[L,U,P]=lu(A)`命令检查得到的答案。

§ 3.3.3 直接解法的误差分析

一、扰动方程组的误差界

由实际问题得到的方程组的系数矩阵或者常数向量的元素，本身会存在一定的误差；这些初始数据的误差在计算过程中就会向前传播，从而影响到方程组的解。

初始数据误差和方程组的近似解的误差之间关系

例3.5 考察方程组：

$$\begin{bmatrix} 3 & 1 \\ 3.0001 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 4.0001 \end{bmatrix}$$

精确解为 $x^* = (1 \quad 1)^T$

设方程组存在扰动

$$\begin{bmatrix} 3 & 1 \\ 2.9999 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 4.0002 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 1 & 4 \\ -0.0001 & 0 & 0.0002 \end{bmatrix}$$

精确解为 $x^* = (-2 \quad 10)^T$

上例说明该方程组的解对初始元素的扰动非常敏感。

设方程组为$Ax=b$ 系数矩阵A和常数向量b的扰动δA和δb，

实际求解的方程组为 $(A+\delta A)x=(b+\delta b)$

定义3.3

如果 δA 和 δb 很小，而 δx 很大，则成方程组$Ax=b$是病态方程组，称系数矩阵A为关于求解方程组或求逆的病态矩阵。反之，称方程组$Ax=b$是良态方程组，称系数矩阵A为关于求解方程组或求逆的良态矩阵。

病态方程组对任何算法都将产生数值不稳定性

➢求解病态方程组时，常用的几种处理原则

❶采用高精度的算术运算；

❷采用某些特殊的数值方法求解；

❸重新寻找出现病态的原因，改变原问题的提法。

❹采用预处理方法；

$$Ax=b \Leftrightarrow PAQQ^{-1}x=Pb \Leftrightarrow \bar{A}\bar{x}=\bar{b}$$

选讲

其中 $\bar{A}=PAQ; \bar{x}=Q^{-1}x; \bar{b}=Pb$

可逆矩阵 P和Q 的选择要求满足：PAQ的条件数大于A的条件数，其中矩阵A的条件数记为

$$cond(A) := \|A\| \left\|A^{-1}\right\|$$

更多关于条件数的介绍，详见相关数学论著

选讲

当cond(A)>>1时，则方程组是“病态”的；
当cond(A)较小时，则方程组是“良态”的. 通常的条件数有：

$$(1)\ Cond(A)_\infty = \| A \|_\infty \cdot \| A^{-1} \|_\infty$$

$$(2)\ Cond(A)_2 = \| A \|_2 \cdot \| A^{-1} \|_2 = \sqrt{\lambda_{max}(A^T A) / \lambda_{min}(A^T A)}$$

特别地，若 A 对称，则 $cond(A)_2 = \dfrac{\max|\lambda|}{\min|\lambda|}$

例题 已知 $A=\begin{pmatrix} 1 & 0.99 \\ 0.99 & 0.98 \end{pmatrix}$ 求A的条件数.

解：由 $\det(\lambda I - A) = 0 \Rightarrow \begin{cases} \lambda_1 = 1.980050504 \\ \lambda_2 = -0.000050504 \end{cases}$

$$cond(A)_2 = \left| \frac{\lambda_1}{\lambda_2} \right| \approx 39206 >> 1.$$

说明由A构成的系数矩阵方程组是“病态”的。

3.4 线性方程组的迭代解法

这一节主要讲述如何把第2章介绍的迭代法扩展到更高维数。

思路

与解$f(x)=0$的不动点迭代相似，将方程组$\boldsymbol{Ax=b}$等价改写成$\boldsymbol{x=Bx+f}$形式，从而建立迭代格式

$$x^{(k+1)} = B\,x^{(k)} + f$$

从 x0 出发，生成迭代序列$\{x^{(k)}\}$

迭代法是一种逐次逼近的方法,与直接法比较,具有:
程序简单,存储量小的优点。
特别适用于求解系数矩阵为大型稀疏矩阵的方程组。

上述定义说明：根据矩阵范数的等价性，如果矩阵序列收敛，则对任何一种矩阵范数而言均收敛。

定义3.4（矩阵序列的极限）

定义了矩阵范数$\|\cdot\|$的空间$\boldsymbol{R}^{n\times n}$，如果$\exists \boldsymbol{A}\in \boldsymbol{R}^{n\times n}$使得

$$\lim_{k\to\infty}\left\|\boldsymbol{A}^{(k)}-\boldsymbol{A}\right\|=0$$

则称$\left\{\boldsymbol{A}^{(k)}\right\}$收敛于A，记作$\lim\limits_{k\to\infty}\boldsymbol{A}^{(k)}=\boldsymbol{A}$

选用F-范数

$$\lim_{k\to\infty}\left\|A^{(k)}-A\right\|_F=0\Leftrightarrow\lim_{k\to\infty}a_{ij}^{(k)}=a_{ij};i,j=1,2,\cdots,n$$

例3.6：

$$A^{(k)}=\begin{pmatrix}\frac{1}{k} & 1 & \frac{k}{2k+1}\\ 0 & ke^{-k} & k^2e^{-k}\\ \frac{1}{k^2} & -1 & e^{-k}\sin k\end{pmatrix}\to\begin{pmatrix}0 & 1 & \frac{1}{2}\\ 0 & 0 & 0\\ 0 & -1 & 0\end{pmatrix}$$

迭代公式的构造

➢迭代法的一般迭代格式：

$$x^{(k+1)} = F_k(x^{(k)}, x^{(k-1)}, \cdots, x^{(k-m)}); k = 0,1,\cdots$$

第k+1步与前m+1步有关，称之为多步迭代法 $(m \geq 1)$

m=0称之为单步迭代法 $x^{(k+1)} = F_k(x^{(k)}); k = 0,1,\cdots$

如果F_k是线性的，称之为单步线性迭代法，即

$$x^{(k+1)} = B_k x^{(k)} + f_k; k = 0,1,\cdots$$

迭代矩阵

如果B_k 和 f_k 与k无关，称之为单步定常线性迭代法：

$$\boxed{x^{(k+1)} = Bx^{(k)} + f;k = 0,1,\cdots}$$

设 $A \in R^{n\times n}, b \in R^n, \det(A) \neq 0$ 如果方程组

$$Ax = b \Leftrightarrow x = Bx + f$$

则可以构造单步定常线性迭代法 $x^{(k+1)} = Bx^{(k)} + f$

当迭代公式产生的序列$\left\{x^{(k)}\right\}_{k=0}^{\infty}$收敛到方程组的解 x^*

即 $\lim\limits_{k\to\infty} x^{(k)} = x^*$，则称该迭代法是收敛的。

定义3.5　下面三个命题是等价的：

❶ 迭代法 $\boldsymbol{x}^{(k+1)} = \boldsymbol{B}\boldsymbol{x}^{(k)} + \boldsymbol{f}$ 收敛；

❷ $\rho(\boldsymbol{B}) < 1$

❸ 至少存在一种从属矩阵范数 $\|\cdot\|$，使得 $\|B\| < 1$

由于谱半径不易计算，实际验证时主要利用❸，采用矩阵的1-范数、∞-范数或者F-范数。

➢迭代法的收敛速度：

设迭代法 $x^{(k+1)} = Bx^{(k)} + f$ 收敛，即 $\rho(B) < 1$

定义3.6

$$R_k(B) = -\ln \|B^k\|^{\frac{1}{k}}$$

平均压缩率

称之为迭代法 $x^{(k+1)} = Bx^{(k)} + f$ 的平均收敛率。

$$x^{(k)} - x^* = B(x^{(k-1)} - x^*) = \cdots = B^k(x^{(0)} - x^*)$$

$$\|x^{(k)} - x^*\| \le \|B^k\| \|(x^{(0)} - x^*)\|$$

上式说明：$\|B^k\|$可看作第k次迭代误差范数的压缩率

记 $e^{(k)} = x^{(k)} - x^*$ $\qquad$ $e^{(k)} = B^k e^{(0)}$

$$\left\|e^{(k)}\right\| \le \left\|B^k\right\| \left\|e^{(0)}\right\| \to 0 \quad \forall \varepsilon > 0, \exists K, k > K, \left\|B^k\right\| \le \varepsilon$$

$$\left\|B^k\right\| \le \varepsilon \Leftrightarrow k \ge \frac{-\ln \varepsilon}{-\frac{1}{k}\ln\left\|B^k\right\|} = \boxed{\frac{-\ln \varepsilon}{-\ln\left\|B^k\right\|^{\frac{1}{k}}}}$$

上式说明：最小迭代次数与$-\ln\left\|B^k\right\|^{\frac{1}{k}}$成反比

定义3.7

$$R(B) = -\ln \rho(B)$$

$$\lim_{k\to\infty} \left\|B^k\right\|^{\frac{1}{k}} = \rho(B)$$

称之为迭代法 $x^{(k+1)} = Bx^{(k)} + f$的渐进收敛率，或者渐进收敛速度，简称收敛速度。

迭代次数的近似估计式 $k \approx \dfrac{-\ln \varepsilon}{R(B)}$

上式说明：谱半径$\rho(B)$越小，收敛速度越快。

§3.4.2 Jacobi（雅可比）迭代法

设方程组 $Ax=b; A=(a_{ij})_{n\times n}, b=(b_i)_{1\times n}; \det(A)\neq 0$

将系数矩阵分裂为： $A=D-L-U$

其中 $D=diag(a_{11},a_{22},\ldots,a_{nn})$

$$-L=\begin{pmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ a_{31} & a_{32} & 0 & & \\ \vdots & \vdots & \cdots & \ddots\ 0 & \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & 0 \end{pmatrix} \quad -U=\begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ & 0 & a_{23} & \cdots & a_{2n} \\ & & 0 & \cdots & \vdots \\ & & & \ddots\ 0 & a_{n-1,n} \\ & & & & 0 \end{pmatrix}$$

如果 $a_{ii}\neq 0(i=1,2,\cdots,n)$ 原方程组可化为

$$x=D^{-1}(L+U)x+D^{-1}b\ \boxed{=Bx+f}$$

其中 $B=D^{-1}(L+U)=(I-D^{-1}A);\ f=D^{-1}b$

相应的迭代格式 $x^{(k+1)}=Bx^{(k)}+f;k=0,1,2,\cdots$

上述方法称为Jacobi迭代法，简称J法或简单迭代法

分量形式：

$$\boxed{x_i^{(k+1)}=\frac{b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}}{a_{ii}};i=1,2,\cdots,n}$$

例3.7

$$\begin{aligned} 4x - y + z &= 7 \\ 4x - 8y + z &= -21 \\ -2x + y + 5z &= 15 \end{aligned}$$

上述方程可表示成如下形式：

$$x = \frac{7 + y - z}{4}$$

$$y = \frac{21 + 4x + z}{8}$$

$$z = \frac{15 + 2x - y}{5}$$

这样就提出了下列雅可比迭代过程：

$$x_{k+1} = \frac{7 + y_k - z_k}{4}$$

$$y_{k+1} = \frac{21 + 4x_k + z_k}{8}$$

$$z_{k+1} = \frac{15 + 2x_k - y_k}{5}$$

如果从 $\boldsymbol{P}_0 = (x_0, y_0, z_0) = (1,2,2)$ 开始，则上式中的迭代将收敛到解(2,4,3)。将 $x_0 = 1, y_0 = 2$ 和 $z_0 = 2$ 代入上式中每个方程的右边，即可得到如下新值：

$$x_1 = \frac{7 + 2 - 2}{4} = 1.75$$

$$y_1 = \frac{21 + 4 + 2}{8} = 3.375$$

$$z_1 = \frac{15 + 2 - 2}{5} = 3.00$$

新的点 $P_1=(1.75,3.375,3.00)$ 比 P_0 更接近(2,4,3)。使用迭代过程(3)生成点的序列 $\{P_k\}$ 将收敛到解(2,4,3)(如表 3.2 所示)。 ■

表 3.2　求解线性方程组(1)的收敛的雅可比迭代

k	x_k	y_k	z_k
0	1.0	2.0	2.0
1	1.75	3.375	3.0
2	1.84375	3.875	3.025
3	1.9625	3.925	2.9625
4	1.99062500	3.97656250	3.00000000
5	1.99414063	3.99531250	3.00093750
⋮	⋮	⋮	⋮
15	1.99999993	3.99999985	2.99999993
⋮	⋮	⋮	⋮
19	2.00000000	4.00000000	3.00000000

这个过程称为**雅可比迭代**,可用来求解某些类型的线性方程组。经过 19 步迭代,迭代过程收敛到一个精度为 9 位有效数字的近似值(2.00000000,4.00000000,3.00000000)。

§3.4.3 Gauss-Seidel（高斯-赛德尔）迭代法

在Jacobi迭代公式中，计算$x_i^{(k+1)}$时，利用已经算出来的新的$x_1^{(k+1)}, x_2^{(k+1)}, \cdots, x_{i-1}^{(k+1)}$值，从而得到Gauss-Seidel迭代法。

Gauss-Seidel迭代法是Jacobi迭代法的一种改进

➢ G-S迭代法的分量形式：

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}; i = 1, 2, \cdots, n$$

例3.8：利用Jacobi和Gauss-Seidel迭代法求解方程组

$$\begin{pmatrix} 10 & 3 & 1 \\ 2 & -10 & 3 \\ 1 & 3 & 10 \end{pmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix}$$

解：

Jacobi迭代格式

$$\begin{cases} x_1^{(k+1)} = \left(14 - 3x_2^{(k)} - x_3^{(k)}\right) \big/ 10 \\ x_2^{(k+1)} = \left(-5 - 2x_1^{(k)} - 3x_3^{(k)}\right) \big/ (-10) \\ x_3^{(k+1)} = \left(14 - x_1^{(k)} - 3x_2^{(k)}\right) \big/ 10 \end{cases}$$

Gauss-Seidel迭代格式

$$\begin{cases} x_1^{(k+1)} = \left(14 - 3x_2^{(k)} - x_3^{(k)}\right) \Big/ 10 \\ x_2^{(k+1)} = \left(-5 - 2x_1^{(k+1)} - 3x_3^{(k)}\right) \Big/ (-10) \\ x_3^{(k+1)} = \left(14 - x_1^{(k+1)} - 3x_2^{(k+1)}\right) \Big/ 10 \end{cases}$$

取初值 $x = (0 \quad 0 \quad 0)^T$

计算结果

Jacobi迭代法

要求精度	迭代次数	方程组的近似解
0.001	9	(1.0002507 1.0000694 1.0002507)
0.0001	10	(0.9999541 1.0001253 0.9999541)
0.00001	14	(0.9999981 1.0000020 0.9999981)

Gauss-Seidel迭代法

要求精度	迭代次数	方程组的近似解
0.001	5	(0.9997916 0.9998479 1.0000664)
0.0001	7	(0.9999929 0.9999949 1.0000022)
0.00001	8	(1.0000013 1.0000009 0.9999996)

假设线性方程组Ax=b的矩阵A是严格对角占优的。则Jacobi迭代法和Gauss-Seidel迭代法的Matlab程序实现如下:

Jacobi.m

GaussSeidel.m

Jacobi.m

```
function X=jacobi(A,B,P,delta, max1)

% Input    - A is an N x N nonsingular matrix
%          - B is an N x 1 matrix
%          - P is an N x 1 matrix; the initial guess
%          - delta is the tolerance for P
%          - max1 is the maximum number of iterations
% Output - X is an N x 1 matrix: the jacobi approximation to
%            the solution of AX = B

N = length(B);
for k=1:max1
   for j=1:N
      X(j)=(B(j)-A(j,[1:j-1,j+1:N])*P([1:j-1,j+1:N]))/A(j,j);
   end
   err=abs(norm(X'-P));
   relerr=err/(norm(X)+eps);
   P=X';
      if (err<delta)|(relerr<delta)
     break
   end
end

X=X';
```

GaussSeidel.m

```
function X=GaussSeidel(A,B,P,delta, max1)
% Input    - A is an N x N nonsingular matrix
%          - B is an N x 1 matrix
%          - P is an N x 1 matrix; the initial guess
%          - delta is the tolerance for P
%          - max1 is the maximum number of iterations
% Output - X is an N x 1 matrix: the gauss-seidel
approximation to
%            the solution of AX = B

N = length(B);

for k=1:max1
   for j=1:N
      if j==1
        X(1)=(B(1)-A(1,2:N)*P(2:N))/A(1,1);
      elseif j==N
         X(N)=(B(N)-A(N,1:N-1)*(X(1:N-1))')/A(N,N);
      else
%X contains the kth approximations and P the (k-1)st
        X(j)=(B(j)-A(j,1:j-1)*X(1:j-1)'-
A(j,j+1:N)*P(j+1:N))/A(j,j);
      end
   end
 err=abs(norm(X'-P));
   relerr=err/(norm(X)+eps);
   P=X';
      if (err<delta)|(relerr<delta)
     break
   end
end

X=X';
```

作业3.3

在习题 1 到习题 8 中：

(a) 初始值 $P_0=0$，利用雅可比迭代求解 P_k，$k=1,2,3$。雅可比迭代收敛到解吗？

(b) 初始值 $P_0=0$，利用高斯－赛德尔迭代求解 P_k，$k=1,2,3$。高斯－赛德尔迭代收敛到解吗？

5.
$$\begin{aligned} 5x - y + z &= 10 \\ 2x + 8y - z &= 11 \\ -x + y + 4z &= 3 \end{aligned}$$

6.
$$\begin{aligned} 2x + 8y - z &= 11 \\ 5x - y + z &= 10 \\ -x + y + 4z &= 3 \end{aligned}$$

算法与程序

4. 利用高斯－赛德尔迭代法求解下列带状方程。

$$\begin{array}{rrrrrrl}
12x_1 & -\ 2x_2 & +\ x_3 & & & & =5 \\
-2x_1 & +\ 12x_2 & -\ 2x_3 & +\ x_4 & & & =5 \\
x_1 & -\ 2x_2 & +\ 12x_3 & -\ 2x_4 & +\ x_5 & & =5 \\
 & x_2 & -\ 2x_3 & +\ 12x_4 & -\ 2x_5 & +\ x_6 & =5 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & x_{46} & -\ 2x_{47} & +\ 12x_{48} & -\ 2x_{49} & +\ x_{50} & =5 \\
 & & x_{47} & -\ 2x_{48} & +\ 12x_{49} & -\ 2x_{50} & =5 \\
 & & & x_{48} & -\ 2x_{49} & +\ 12x_{50} & =5
\end{array}$$

回顾

第三章 线性方程组$Ax=b$的数值解法

3.1 引言
3.2 线性代数的基础知识
3.3 直接解法：
- 3.3.1 Gauss消去法
- 3.3.2 三角分解法 （LU分解）
- 3.3.3 直接解法的误差分析

3.4 迭代解法
- 3.4.1 迭代法的基本概念（收敛判定条件）
- 3.4.2 Jacobi迭代法
- 3.4.3 Gauss-Seidel迭代法
- 3.4.4 超松弛(SOR)迭代法（选讲）
- 3.4.5 共轭梯度法（选讲）

§ 3.4.4 超松弛(SOR)迭代法（选讲）

思想

类似于Gauss-Seidel迭代法的改进方法，利用第k次迭代值和第k +1次的Gauss-Seidel迭代值作加权平均

Gauss-Seidel迭代法的计算公式：

$$\bar{x}_i^{(k+1)} \Leftarrow x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}$$

作加权平均

$$x_i^{(k+1)} = \omega \bar{x}_i^{(k+1)} + (1-\omega) x_i^{(k)}$$

➢超松弛(SOR)迭代法的分量形式：

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

$$i = 1, 2, \cdots, n$$

其中ω称为松弛因子； $\omega = 1$时即为Gauss-Seidel迭代

➢**SOR迭代法的迭代矩阵：** 记 $A = D - L - U$

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega D^{-1}(b + Lx^{(k+1)} + Ux^{(k)})$$

$x^{(k+1)} = Sx^{(k)} + f$ 其中 $f = \omega(D-\omega L)^{-1} b$

迭代矩阵 $S = (D-\omega L)^{-1}[(1-\omega)D + \omega U]$

例3.9：写出SOR迭代法求解下列方程组的迭代格式

$$\begin{pmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 24 \\ 30 \\ -24 \end{pmatrix}$$

解： SOR迭代法的迭代公式

$$\begin{cases} x_1^{(k+1)} = (1-\omega)x_1^{(k)} + \dfrac{1}{4}\omega(24 - 3x_2^{(k)}) \\ x_2^{(k+1)} = (1-\omega)x_2^{(k)} + \dfrac{1}{4}\omega(30 - 3x_1^{(k+1)} + x_3^{(k)}) \\ x_3^{(k+1)} = (1-\omega)x_3^{(k)} + \dfrac{1}{4}\omega(-24 + x_2^{(k+1)}) \end{cases}$$

取初始向量 $x^{(0)} = (1 \quad 1 \quad 1)^T$

选取不同的 ω值进行计算，结果见下表

ω的值	要求精度	迭代次数	方 程 组 的 近 似 解
$\omega=1$	0.001	12	(3.0012790 3.9989342 -5.0002665)
	0.0001	16	(3.0001952 3.9998374 -5.0000407)
	0.00001	21	(3.0000186 3.9999845 -5.0000039)
$\omega=0.95$	0.001	12	(3.0020191 3.9982705 -5.0004444)
	0.0001	18	(3.0001673 3.9998567 -5.0000368)
	0.00001	23	(3.0000210 3.9999820 -5.0000046)
$\omega=1.25$	0.001	8	(2.9997451 4.0000653 -4.9998924)
	0.0001	10	(2.9999853 4.0000031 -4.9999935)
	0.00001	12	(2.9999993 4.0000001 -4.9999996)
$\omega=1.5$	0.001	13	(3.0006104 4.0001741 -5.0007434)
$\omega=1.95$	0.001	151	(2.9995106 4.0017780 -5.0027919)

二、SOR迭代法的收敛性：

定理3.5 对 $\forall A=(a_{ij})\in R^{n\times n}$， 设其对角元皆非零，

则对所有实数 ω，有 $\rho(S)\geq|\omega-1|$

其中 $S=(D-\omega L)^{-1}[(1-\omega)D+\omega U]$

证明： $A=D-L-U$

设迭代矩阵 S 的特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$

$$\det(S)=\lambda_1\lambda_2\cdots\lambda_n.$$

$$\begin{aligned}\det(S)&=\det(D-\omega L)^{-1}\det([(1-\omega)D+\omega U])\\&=\det D^{-1}\det[(1-\omega)D]=(1-\omega)^n\end{aligned}$$

$$|1-\omega|^n \leq |\lambda_1||\lambda_2|\cdots|\lambda_n| \leq \left(\max_{1\leq i\leq n}|\lambda_i|\right)^n$$

$$\boxed{|1-\omega| \leq \rho(S)}$$

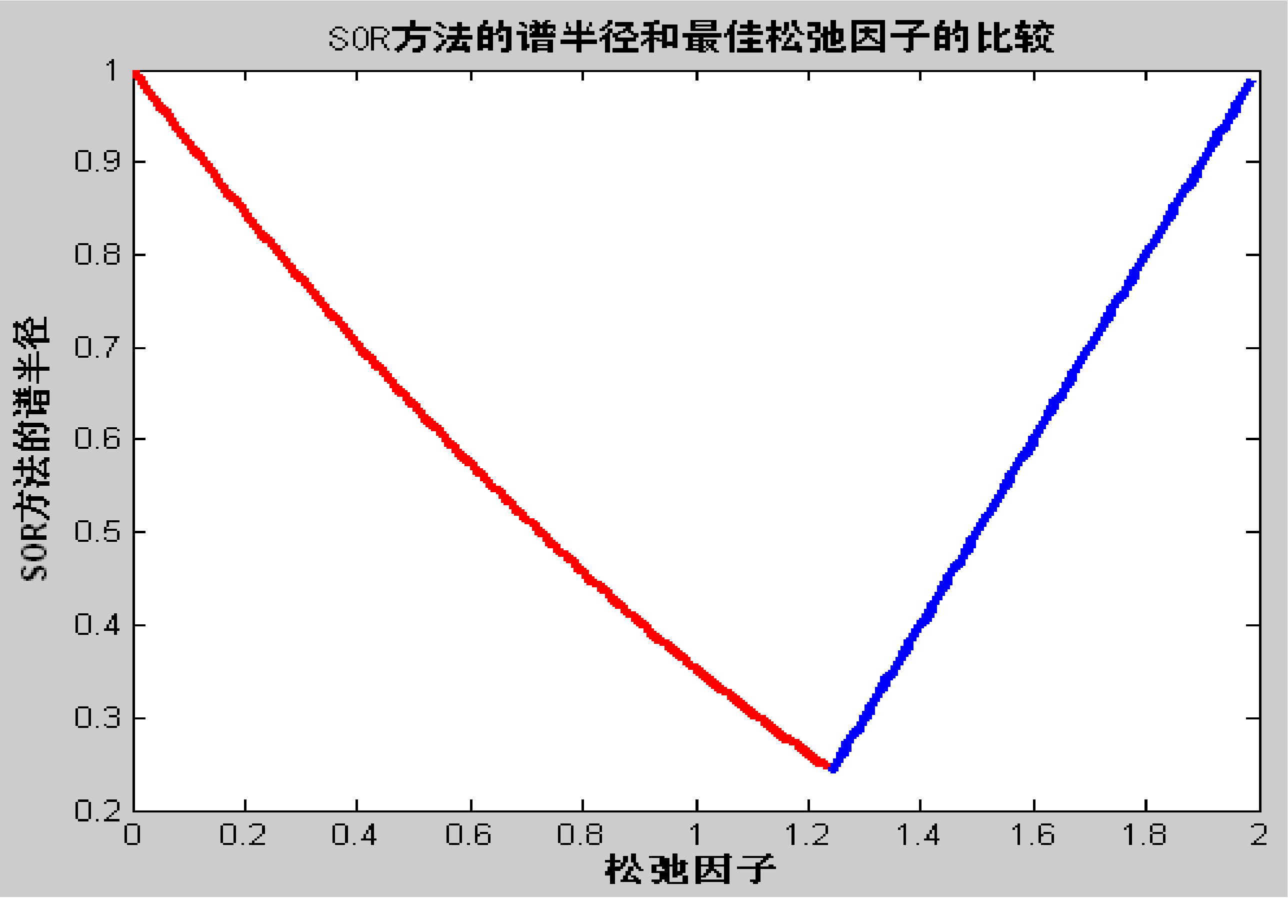
谱半径

推论3.1　如果求解方程组 $Ax=b$ 的SOR法收敛，则

$$|1-\omega|<1 \quad (0<\omega<2)$$

推论3.2　设 $A\in R^{n\times n}$为对称正定矩阵，且$0<\omega<2$

则求解方程组 $Ax=b$的SOR法收敛。



关于最佳松弛因子的进一步讨论，可参考文献

《数值计算方法》（下册），林成森著

《矩阵迭代分析》，**R.S.Varga**著，蒋尔雄等译

§3.4.5 共轭梯度法（选讲）

思想

共轭梯度法是一种变分方法，将求解方程组问题等价转化为一个二次函数的极值问题。

变分方法是以变分学和变分原理为基础的一种近似计算方法，是解决力学和其他领域问题的有效数学工具，如最速降线问题等。

与方程组等价的变分问题

设 $A=(a_{ij})\in R^{n\times n}$ 为对称正定矩阵，$Ax=b$

其中 $x=(x_1,x_2,\cdots,x_n)^T$；$b=(b_1,b_2,\cdots,b_n)^T$

定义二次函数 $\varphi: R^n\to R$

$$\boxed{\varphi(x)=\frac{1}{2}(Ax,x)-(b,x)}\boxed{=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}x_i x_j-\sum_{j=1}^{n}b_j x_j}$$

➢二次函数 $\varphi(x)$ 的基本性质:

❶对 $\forall x \in R^n, \nabla\varphi(x) = Ax - b$

❷对 $\forall x, y \in R^n, \alpha \in R$

$$\varphi(x + \alpha y) = \varphi(x) + \alpha(Ax - b, y) + \frac{\alpha^2}{2}(Ay, y)$$

❸设 $x^* = A^{-1}b$ 为 $Ax = b$ 的解，则

$$\varphi(x^*) = -\frac{1}{2}(Ax^*, x^*)$$，且对 $\forall x \in R^n$ 有

$$\varphi(x) - \varphi(x^*) = \frac{1}{2}(Ax, x) - (Ax^*, x) + \frac{1}{2}(Ax^*, x^*)$$

$$= \frac{(A(x - x^*), x - x^*)}{2}$$

定理3.6 设A对称正定，则x^*为$Ax=b$的解的充要条件是

$$\varphi(x^*)=\min_{x\in R^n}\varphi(x)$$

证明：

必要性由上述性质❸易知，下证充分性：

假设存在$\bar{x}$使$\varphi(x)$达到最小，则$\varphi(\bar{x})-\varphi(x^*)\le 0$

再由性质❸知，

$$\frac{(A(\bar{x}-x^*),\bar{x}-x^*)}{2}\ge 0$$

$$\boxed{\bar{x}=x^*}$$

上述定理说明：求解方程组的解等价于求上述二次函数的最小值。
常用方法：迭代解法

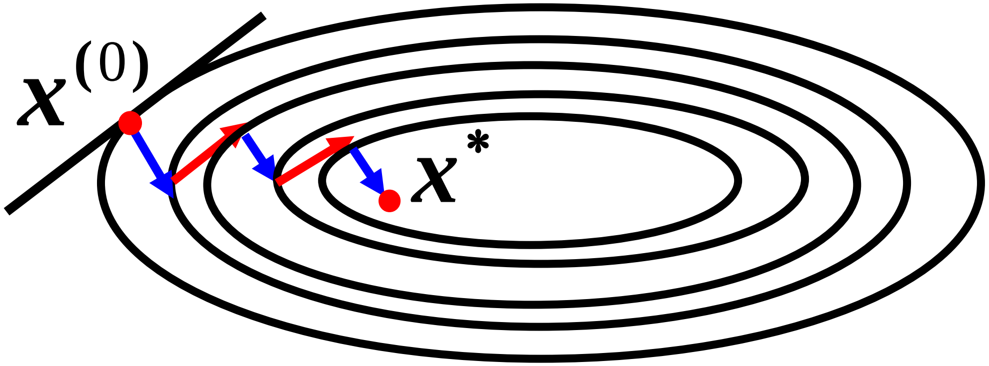
最速下降法

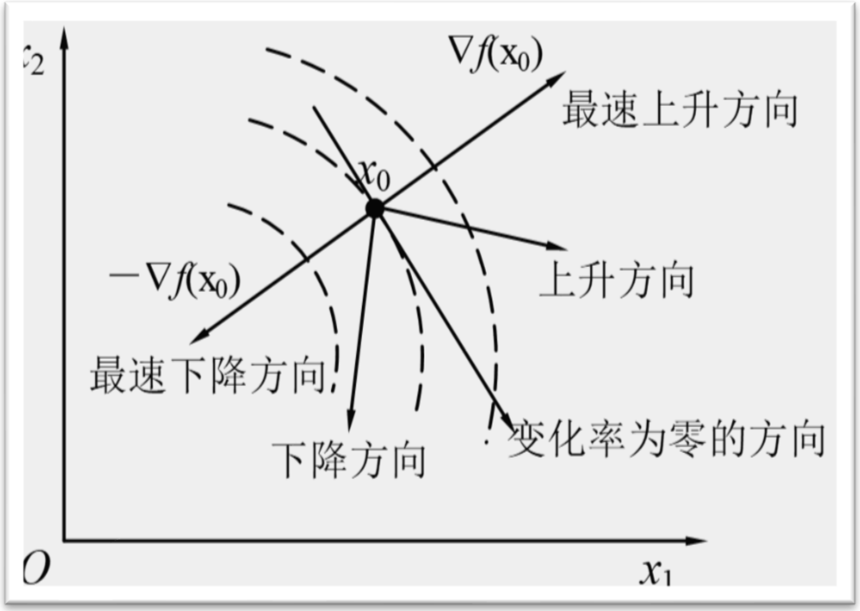
思想

最速下降法是指每次沿着函数值下降最快的方向寻找最小值点。

而函数值下降最快的方向是函数的负梯度方向

➢几何意义：





等值面

垂直

➤最速下降法的实现过程

选取初始向量 $x^{(0)}$，由二次函数$\varphi(x)$ 的基本性质❶

$$-\nabla\varphi(x^{(0)})=b-Ax^{(0)}\boxed{=r^{(0)}}$$

如果$r^{(0)}=0$，则 $x^{(0)}$ 就是方程组的解；

如果$r^{(0)}\neq 0$，则沿 $r^{(0)}$ 方向进行一维极小搜索：

$$\boxed{\text{求步长}\alpha\text{，使得}\min_{\alpha}\varphi(x^{(0)}+\alpha r^{(0)})}$$

$$\frac{d}{d\alpha}\varphi(x^{(0)}+\alpha r^{(0)})=0$$

$$=\frac{d}{d\alpha}[\varphi(x^{(0)})+\alpha(Ax^{(0)}-b,r^{(0)})+\frac{\alpha^2}{2}(Ar^{(0)},r^{(0)})]$$

$$\alpha_0=\frac{(r^{(0)},r^{(0)})}{(Ar^{(0)},r^{(0)})}$$

注意到 $\dfrac{d^2}{d^2\alpha}\varphi(x^{(0)}+\alpha r^{(0)})=(Ar^{(0)},r^{(0)})>0$

$$\min_{\alpha}\varphi(x^{(0)}+\alpha r^{(0)})=\varphi(x^{(0)}+\alpha_0 r^{(0)})$$

令 $x^{(1)}=x^{(0)}+\alpha_0 r^{(0)}$，从而完成第一次迭代。

下面以 $x^{(1)}$ 为新的初值，重复上述过程。

$$\alpha_0=\frac{(r^{(0)},r^{(0)})}{(Ar^{(0)},r^{(0)})}$$

➢最速下降法的算法

选取初值 $x^{(0)} \in R^n$

For k=0,1,2,$\cdots$

$$r^{(k)} = b - Ax^{(k)}$$

$$\alpha_k = \frac{(r^{(k)}, r^{(k)})}{(Ar^{(k)}, r^{(k)})}$$

$$x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)}$$

如果 $\left\| r^{(k)} \right\| < \varepsilon$ ，停止

否则，进行下一次循环

搜索方向是正交的：

$$(r^{(k+1)}, r^{(k)}) = 0$$

$$r^{(k+1)} = b - Ax^{(k+1)}$$

$$= b - A(x^{(k)} + \alpha_k r^{(k)})$$

$$= r^{(k)} - \alpha_k Ar^{(k)}$$

缺陷：收敛速度慢

三、共轭梯度法

设$A=(a_{ij})\in R^{n\times n}$为对称正定矩阵，$Ax=b$

其中 $x=(x_1,x_2,\cdots,x_n)^T$；$b=(b_1,b_2,\cdots,b_n)^T$

定义3.7 设$A\in R^{n\times n}$为对称正定矩阵，若R^n中向量组

$\left\{p^{(0)},p^{(1)},\cdots,p^{(l)}\right\}$ 满足$(Ap^{(i)},p^{(j)})=0\quad i\neq j$

则称它是R^n中的一个A—共轭（A—正交）向量组。

利用一维极小搜索方法确定一组A—共轭方向代替最速下降法中的正交方向来进行迭代。

➢共轭梯度法的实现过程

选取初始向量$x^{(0)}$，$p^{(0)}=r^{(0)}$，$\alpha^{(0)}=\dfrac{(r^{(0)},r^{(0)})}{(Ar^{(0)},r^{(0)})}$

$$x^{(1)}=x^{(0)}+\alpha_0 p^{(0)}$$

$$x^{(2)}=x^{(1)}+\alpha_1 p^{(1)}=x^{(0)}+\alpha_0 p^{(0)}+\alpha_1 p^{(1)}$$

下面为了讨论方便，令$x^{(0)}=0$

$$\boxed{x^{(2)}=\alpha_0 p^{(0)}+\alpha_1 p^{(1)}}$$

其中 $\alpha_1, p^{(1)}$ 满足：

$$\boxed{\begin{aligned}&\varphi(x^{(2)})=\min_{\alpha}\varphi(x^{(1)}+\alpha_1 p^{(1)})\\ \Leftrightarrow\ &\varphi(x^{(2)})=\min_{x\in Span\{p^{(0)},p^{(1)}\}}\varphi(x)\end{aligned}}$$

设 $x \in Span\{P^{(0)}, P^{(1)}\}$，记为

$$x = y + \alpha p^{(1)}; y \in Span\{p^{(0)}\}$$

由性质❷知 $\varphi(x) = \varphi(y + \alpha p^{(1)})$

$$= \varphi(y) + \alpha(Ay, p^{(1)}) - \alpha(b, p^{(1)}) + \frac{\alpha^2}{2}(Ap^{(1)}, p^{(1)})$$

令 $(Ay, p^{(1)}) = 0; y \in Span\{p^{(0)}\}$

即 $(Ap^{(0)}, p^{(1)}) = 0$

则 $p^{(0)}, p^{(1)}$是 $A-$共轭向量组

$$\boxed{\varphi(x) = \varphi(y) + \frac{\alpha^2}{2}(Ap^{(1)}, p^{(1)}) - \alpha(b, p^{(1)})}$$

$$\min_{x\in Span\{p^{(0)},p^{(1)}\}}\varphi(x)=\min_{y\in Span\{p^{(0)}\}}\varphi(y)$$
$$+\min_{\alpha}\left[\frac{\alpha^2}{2}(Ap^{(1)},p^{(1)})-\alpha(b,p^{(1)})\right]$$

而 $\min_{y\in Span\{p^{(0)}\}}\varphi(y)$ 的解为$y=x^{(1)}$

$\min_{\alpha}\left[\frac{\alpha^2}{2}(Ap^{(1)},p^{(1)})-\alpha(b,p^{(1)})\right]$的解为

$$\alpha_1=\frac{(b,p^{(1)})}{(Ap^{(1)},p^{(1)})}=\frac{(r^{(1)},p^{(1)})}{(Ap^{(1)},p^{(1)})}$$

令 $p^{(1)} = r^{(1)} + \beta_0 p^{(0)}$ 由 $(Ap^{(0)}, p^{(1)}) = 0$

$$\beta_0 = -\frac{(r^{(1)}, Ap^{(0)})}{(p^{(0)}, Ap^{(0)})}$$

$$\alpha_1 = \frac{(r^{(1)}, p^{(1)})}{(Ap^{(1)}, p^{(1)})}$$

从而得到第2次迭代 $x^{(2)} = x^{(1)} + \alpha_1 p^{(1)}$

上述迭代思想推广到任意次迭代就是共轭梯度法

设按照上述方法已经得到 $p^{(0)}, p^{(1)}, \cdots, p^{(k-1)}$ 和 $x^{(k)}$

下面确定α_k和 p_k，计算$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$

通过一维极小搜索：

$$\frac{d}{d\alpha}\varphi(x^{(k)} + \alpha p^{(k)}) = 0$$

$$\alpha_k = \frac{(r^{(k)}, p^{(k)})}{(Ap^{(k)}, p^{(k)})}$$

为了讨论方便，令 $x^{(0)} = 0$

$$x^{(k+1)} = \alpha_0 p^{(0)} + \alpha_1 p^{(1)} + \cdots + \alpha_k p^{(k)}$$

$$x^{(k+1)} \in Span\{p^{(0)}, \cdots, p^{(k)}\}$$

记 $x = y + \alpha p^{(k)}; y \in Span\{p^{(0)}, p^{(1)}, \cdots, p^{(k-1)}\}$

由性质❷知 $\varphi(x) = \varphi(y + \alpha p^{(k)})$

$$= \varphi(y) + \alpha(Ay, p^{(k)}) - \alpha(b, p^{(k)}) + \frac{\alpha^2}{2}(Ap^{(k)}, p^{(k)})$$

令 $(Ay, p^{(k)}) = 0$; $y \in Span\{p^{(0)}, p^{(1)}, \cdots, p^{(k-1)}\}$

即 $(Ap^{(j)}, p^{(k)}) = 0 \quad j = 0, 1, \cdots, k-1$

则 $p^{(0)}, p^{(1)}, \cdots, p^{(k)}$ 是 A—共轭向量组

令 $p^{(k)} = r^{(k)} + \beta_{k-1} p^{(k-1)}$ 由 $(Ap^{(k-1)}, p^{(k)}) = 0$

$$\beta_{k-1} = -\frac{(r^{(k)}, Ap^{(k-1)})}{(p^{(k-1)}, Ap^{(k-1)})}$$

$$\alpha_k = \frac{(r^{(k)}, p^{(k)})}{(Ap^{(k)}, p^{(k)})}$$

从而得到第k+1次迭代 $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$

同时注意到

$$(r^{(k+1)}, p^{(k)}) = (b - Ax^{(k+1)}, p^{(k)})$$
$$= (r^{(k)}, p^{(k)}) - \alpha(Ap^{(k)}, p^{(k)}) = 0$$

$k = 0, 1, 2, \cdots$

$$(r^{(k)}, p^{(k)}) = (r^{(k)}, r^{(k)} + \beta_{k-1} p^{(k-1)}) = (r^{(k)}, r^{(k)})$$

$$\beta_k = -\frac{(r^{(k+1)}, Ap^{(k)})}{(p^{(k)}, Ap^{(k)})} = -\frac{(r^{(k+1)}, \alpha_k^{-1}(r^{(k)} - r^{(k+1)}))}{(p^{(k)}, Ap^{(k)})}$$

$$= \frac{\alpha_k^{-1}(r^{(k+1)}, r^{(k+1)})}{(p^{(k)}, Ap^{(k)})}$$

$$\boxed{\beta_k = \frac{(r^{(k+1)}, r^{(k+1)})}{(r^{(k)}, r^{(k)})}}$$

定理　按照上述方法定义的算法具有如下性质：

❶ $(r^{(i)}, r^{(j)}) = 0, i \neq j$

❷ $p^{(0)}, p^{(1)}, \cdots, p^{(k)}$ 是 $A-$共轭向量组；

❸用C-G法求解n阶方程组，理论上最多迭代n步。

➤共轭梯度法的算法

选取初值 $x^{(0)} \in R^n$

计算 $r^{(0)} = b - Ax^{(0)}$

$$p^{(0)} = r^{(0)}$$

For k=0 , 1 , 2 , ⋯ , n

计算 $\alpha_k = \dfrac{(r^{(k)}, r^{(k)})}{(Ap^{(k)}, p^{(k)})}$

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$$

$$r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}$$

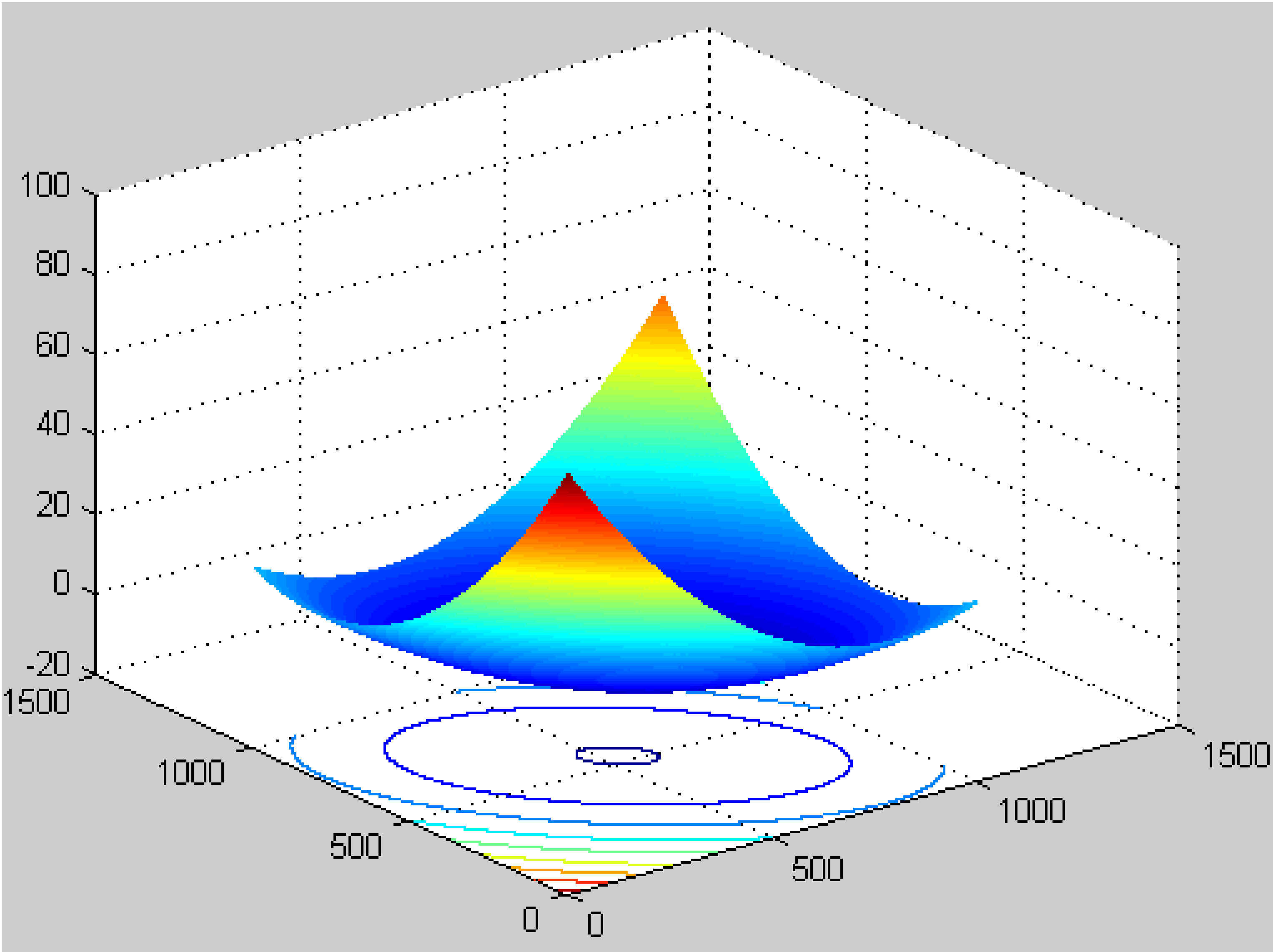
如果 $r^{(k+1)} = 0$ ，停止

否则，计算

$$\beta_k = -\frac{(r^{(k+1)}, r^{(k+1)})}{(r^{(k)}, r^{(k)})}$$

$$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}$$

进行下一次循环；



返回

本章教学要求及重点难点

- 回顾线性代数基本知识
- 掌握Gauss消去法、三角分解法及其误差分析
- 掌握迭代过程的稳定性、敛散性证明
- 掌握迭代法的基本思想
- 重点： Jacobi迭代法、Gauss-Seidel迭代法
  的基本原理
- 难点：迭代法的收敛性分析与证明
- 了解超松弛(SOR)迭代法、梯度下降法以及共轭梯度法