第3章 迭代法

3.1 雅可比迭代和高斯-塞德尔迭代

前一章讨论了求解线性方程组的直接法，这些方法主要适用于中小规模的矩阵. 例如 $n\times n$ 矩阵的 LU 分解需要 $\frac{2}{3}n^3$ 次浮点运算，如果 $n=100$，则浮点运算次数为 6.7×10^5，这在标准台式计算机上只需一秒钟就可完成. 但如果 $n=1000$，浮点运算次数约为 6.7×10^8，这时所需要的运算时间就可观了(除了需要成千倍的浮点运算外，还和内存管理有关.). 由于实际中所遇到的矩阵往往是上万阶(例如气象学领域中的偏微分方程数值解问题)甚至是几十万阶(例如对遗传学数据的分析)，因此需要用更有效的方法解决这类问题.

对于大型稠密矩阵，如果不能用简单矩阵近似这类矩阵，那么计算的时间会更长. 将元素从存储器移动到处理器所花费的时间可能要比实际计算的时间还要长.

但幸运的是实际中的大型矩阵通常都是稀疏的. 对于大型稀疏矩阵有很多种迭代方法，如果能将迭代法与稀疏矩阵的存储特性结合起来，那么计算时间将会明显减少. 一般来说，同样条件下直接法需要 $O(n^3)$ 而迭代法只需要 $O(n^2)$. 最好的情况下，稀疏矩阵的计算量可以达到 $O(N)$，这里 N 是矩阵中非零元素的个数.

这一节主要讨论线性方程组的两类迭代解法.

考虑方阵线性方程组 $Ax=b$，迭代法的目的是要建立一种迭代格式 $x^{k+1}=F(x^k)$，其中初始值 $x^0\in\mathbf{R}^n$ 是给定的，F 是容易计算的简单函数. 方法之一是将 A 分解为上三角、下三角、对角三个部分，这种做法称为**矩阵分裂**：

$$A=U+L+D \tag{3.1}$$

其中 U 矩阵的严格上三角部分为 A 的严格上三角部分，其余部分为零，L 矩阵的严格下三角部分为 A 的严格下三角部分，其余部分为零，D 是由 A 的主对角元素构成的对角矩阵. 例如

$$\begin{pmatrix}1&2&3\\4&5&6\\7&8&9\end{pmatrix}=\begin{pmatrix}0&2&3\\0&0&6\\0&0&0\end{pmatrix}+\begin{pmatrix}0&0&0\\4&0&0\\7&8&0\end{pmatrix}+\begin{pmatrix}1&0&0\\0&5&0\\0&0&9\end{pmatrix}$$
$$=U+L+D$$

就是形如(3.1)的分裂. 这样

$$Ax=b$$

$$\begin{aligned}(U+L+D)x &= b\\ Dx &= -(U+L)x+b\\ x &= -D^{-1}(U+L)x+D^{-1}b\end{aligned}\tag{3.2}$$

(本节始终假定 A 的对角元素非零). 由于这个方程组具有固定点迭代 $x=g(x)$ 形式,所以或许

$$x^{k+1} = -D^{-1}(U+L)x^k + D^{-1}b \tag{3.3}$$

可以成为一个有用的迭代格式. 事实上,(3.3)称为**雅可比迭代**,由于 D 是对角阵,因此 D^{-1} 的计算很容易.

例 3.1.1 用雅可比迭代求解方程组 $Ax=(3,\ -1,\ 4)^T$,其中 $A=[4\ \ 2\ \ 1;\ \ 1\ \ 3\ \ 1;\ \ 1\ \ 1\ \ 4]$. 利用(3.2),先形成 $Dx=-(U+L)x+b$,这一步相当于将非对角元素移到等式的右边,即将

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

变为

$$\begin{aligned}4x_1 &= 3-2x_2-x_3\\ 3x_2 &= -1-x_1-x_3\\ 4x_3 &= 4-x_1-x_2\end{aligned}$$

方程两边同除对角元素并进行迭代,得到的式(3.3)为

$$x_1^{k+1} = \frac{1}{4}(3-2x_2^k-x_3^k)$$

$$x_2^{k+1} = \frac{1}{3}(-1-x_1^k-x_3^k)$$

$$x_3^{k+1} = \frac{1}{4}(4-x_1^k-x_2^k)$$

取初值 $x^0=(0,\ 0,\ 0)^T$,则

$$\begin{aligned}x_1^1 &= \frac{1}{4}(3-2\cdot 0-0)\\ &= \frac{3}{4}\end{aligned}$$

$$\begin{aligned}x_2^1 &= \frac{1}{3}(-1-0-0)\\ &= -\frac{1}{3}\end{aligned}$$

$$\begin{aligned}x_3^1 &= \frac{1}{4}(4-0-0)\\ &= 1\end{aligned}$$

$$\begin{aligned}x_1^2 &= \frac{1}{4}\left(3-2\cdot\frac{-1}{3}-1\right)\\ &\doteq 0.6667\end{aligned}$$

$$x_2^2=\frac{1}{3}\left(-1-\frac{3}{4}-1\right)$$
$$\doteq -0.9167$$
$$x_3^2=\frac{1}{4}\left(4-\frac{3}{4}-\frac{-1}{3}\right)$$
$$\doteq 0.8958$$

看上去这些结果将缓慢收敛到真解 $x_1=1, x_2=-1, x_3=1$. 方法并没有认识到 x_3^1 是精确的，但可以看到这些近似值的绝对误差为 $\alpha_1=\|x^1-x\|\doteq 0.7210, \alpha_2=\|x^2-x\|\doteq 0.3590$，所以方法的整体误差是不断缩减的. 继续迭代直至满足某种收敛判定准则. ■

可以将雅可比迭代公式写成分量形式. 如果 x_i^k 表示 x^k 的第 i 个分量，则雅可比迭代中求解对角元素 x_i 的第 i 个方程为

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

不断迭代

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

直到满足某种收敛判定准则. 雅可比迭代的收敛判定准则可以采用绝对误差、相对误差或残量

$$r_k=\|b-Ax_k\|$$

小于某个误差限. 这里用近似相对误差

$$\rho_k=\frac{\|x^k-x^{k-1}\|}{\|x^k\|}$$

作为收敛判定准则. 通常取误差限 τ 为 $\|x^0\|$ 的很小的百分比(但不能小于某个 $\tau_\alpha>0$，见 1.8 节).

现在的问题是：是否对所有矩阵和所有初始值 x^0 雅可比迭代都能收敛？如果收敛，收敛的速度怎样呢？这种方法是否稳定？能否对收敛进行加速？

实验证明对某些矩阵和某些初始值来说雅可比迭代确实是发散的. 为了研究这种方法的收敛性，将(3.3)改写为

$$x^{k+1}=Mx^k+\beta$$

其中矩阵 $M=-D^{-1}(U+L)$，向量 $\beta=D^{-1}b$ 是已知的. 称形如 $x^{(k+1)}=Mx^k+\beta$ 的迭代为**定常迭代**(因为 M 和 β 都与 k 无关). 对于向量范数的诱导范数(见 2.5)，有

$$\begin{aligned}\|x^{k+1}\|&=\|Mx^k+\beta\|\\&\leqslant\|M\|\,\|x^k\|+\|\beta\|\end{aligned}$$

因此

$$\begin{aligned}\|x^{k+1}\|&\leqslant\|M\|\,\|x^k\|+\|\beta\|\\&\leqslant\|M\|(\|M\|\,\|x^{k-1}\|+\|\beta\|)+\|\beta\|\end{aligned}$$

$$
\begin{aligned}
&= \| M \|^2 \| x^{k-1} \| + \| M \| \| \beta \| + \| \beta \| \\
&= \| M \|^2 \| x^{k-1} \| + (\| M \| + 1) \| \beta \| \\
&\leqslant \| M \|^2 (\| M \| \| x^{k-2} \| + \| \beta \|) + (\| M \| + 1) \| \beta \| \\
&= \| M \|^3 \| x^{k-2} \| + (\| M \|^2 + \| M \| + 1) \| \beta \| \\
&\vdots \\
&\leqslant \| M \|^{k+1} \| x^0 \| + (\| M \|^k + \cdots + \| M \| + 1) \| \beta \|
\end{aligned}
$$

这样，若 $\| M \| < 1$，则

$$
\| x^{k+1} \| \leqslant \| M \|^{k+1} \| x^0 \| + \frac{1}{1 - \| M \|} \| \beta \| \tag{3.4}
$$

这是因为 $\| M \|^k + \cdots + \| M \| + 1$ 是几何级数

$$
\sum_{k=0}^{\infty} \| M \|^k = \frac{1}{1 - \| M \|} \tag{3.5}
$$

的部分和. 当 $\| M \| < 1$ 时通项 $\| M \|^{k+1} \to 0$，$k \to \infty$，所以从(3.4)可以猜测 $\| M \| < 1$ 时应该是收敛的. 如果确实这样还可以得到更精确的结果. 由(2.5)节，矩阵 M 的谱半径

$$
\rho(M) = \max_{i=1}^{n} (| \lambda_i |)
$$

其中 $\lambda_1, \cdots, \lambda_n$ 是 M 的特征值. 事实上若 $\rho(M) < 1$，则 M 是**收敛的**：即当 $k \to \infty$ 时，$M^k \to 0$(零矩阵)并且

$$
\sum_{k=0}^{\infty} M^k = (I - M)^{-1}
$$

(类似于(3.5)式). $(I-M)^{-1}$ 称为 M 的**预解式**，M 本身不可逆时其预解式也可以存在. 现在再考虑线性固定点迭代

$$
\begin{aligned}
x^{k+1} &= Mx^k + \beta \\
&= M(Mx^{k-1} + \beta) + \beta \\
&= M^2 x^{k-1} + (M + I)\beta \\
&= M^2 (Mx^{k-2} + \beta) + (M + I)\beta \\
&= M^3 x^{k-2} + (M^2 + M + I)\beta \\
&\vdots \\
&= M^{k+1} x^0 + (M^k + \cdots + M + I)\beta
\end{aligned}
$$

如果 M 是收敛的，则对于任意的初始值 x^0，

$$
\begin{aligned}
\lim_{k \to \infty} x^{k+1} &= \lim_{k \to \infty} (M^{k+1} x^0 + (M^k + \cdots + M + I)\beta) \\
&= 0 + (I - M)^{-1} \beta \\
&= (I - M)^{-1} \beta
\end{aligned}
$$

事实上，对任何初始值，迭代 $x^{k+1} = Mx_k + \beta$ 都收敛到这个唯一值的充要条件是 M 是收敛的. (当 $\rho(M) \geqslant 1$ 时，方法对某些 x^0 可能是收敛的，但不是对所有的 x^0 都收敛. 对这样的 M 方法是不稳定的.)实际上，$x = (I-M)^{-1}\beta$ 就是方法的固定点，因为

$$
Mx + \beta = M(I - M)^{-1}\beta + \beta
$$

$$
\begin{aligned}
&= M\sum_{k=0}^{\infty} M^k\beta + \beta \\
&= \sum_{k=0}^{\infty} M^{k+1}\beta + \beta \\
&= \sum_{k=1}^{\infty} M^k\beta + I\beta \\
&= \sum_{k=0}^{\infty} M^k\beta \\
&= (I-M)^{-1}\beta \\
&= x
\end{aligned}
$$

(为方便记 $M^0=I$). 总之,如果 M 是收敛的,则 $x=Mx+\beta$ 有唯一的固定点 $x=(I-M)^{-1}\beta$ 且固定点迭代一定收敛到这个值.

与 1.4 节关于牛顿法的固定点分析类似,这里用 $\rho(M)<1$ 代替了 $|g'(x)|<1$. 实际上,M 就是迭代函数 $Mx+\beta$ 关于 x 的导数.

因此对于雅可比迭代及任何其他由矩阵分裂技术得到的迭代法中,若由矩阵 $A=U+L+D$ 得到的矩阵 M 是收敛的,则迭代是收敛的. 由于迭代对任何初始值都是收敛的,所以可以直接得到方法的稳定性(方法对于带有扰动的 x^k 仍旧收敛). 可以证明方法的收敛速度为 $\rho(M)^k$,即当 $k\to\infty$ 时

$$\frac{\| x^k - x \|}{\| x^0 - x \|} = O(\rho(M)^k)$$

(线性收敛). 这样 M 的谱半径最好要小一些,如果谱半径接近 1,则收敛会非常慢.

显然并不是所有的 A 都能使 $M=-D^{-1}(U+L)$ 收敛,但有一类矩阵是满足这个的条件的. 若

$$|a_{ii}| \geqslant \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \quad (i=1,2,\cdots,n)$$

则称方阵 A 是**对角占优**的,即每一行对角元素的绝对值都超过这一行其他元素的绝对值之和. 如果不等号严格成立,则称矩阵是**严格对角占优**的,反之则称为**弱对角占优**. 若 A 为严格对角占优矩阵,则 $M=-D^{-1}(U+L)$ 是收敛的且雅可比迭代也收敛. 事实上即使 A 弱对角占优,方法通常也是收敛的. 偏微分方程数值解法中产生的矩阵往往都是对角占优的.

如果 A 不是对角占优矩阵,则原则上必须检验 $\rho(M)$,以了解方法是否可用. 但这样做的成本太高,所以一般都是利用对任何范数都有

$$\rho(M) \leqslant \| M \|$$

的事实. 若对某种范数,有 $\| M \|<1$,则 M 是收敛的,因而雅可比迭代可以使用.

例 3.1.2　矩阵 $A=[1\quad 1\quad 3;\ 1\quad 3\quad 1;\ 3\quad 1\quad 1]$ 并不严格对角占优,但交换第 1 行和最后一行后,得到的 $A_1=[3\quad 1\quad 1;\ 1\quad 3\quad 1;\ 1\quad 1\quad 3]$ 是严格对角占优的. 关于 A_1 的雅可比迭代是收敛的.

矩阵 $M=[0.3\quad 0.2\quad 0.1;\ 0.2\quad 0.2\quad -0.2;\ 0.4\quad -0.5\quad 0]$ 的弗罗贝尼乌斯

范数(见 2.5 节) $\|M\| = \sqrt{0.3^2+0.2^2+\cdots+(-0.5)^2+0^2} \doteq 0.8185$,所以 $\rho(M)<1$(实际上 $\rho(M)\doteq 0.4531$). 因而迭代 $x^{k+1}=Mx^k+c$ 对任意的 c 和 x^0 都是收敛的. ■

用于检验 $\|M\|<1$ 是否成立的最简单的矩阵范数是向量范数 l_∞ 的诱导范数(见 2.5),

$$\|M\|_\rho = \max_{i=1}^{n}\left\{\sum_{j=1}^{n}|m_{ij}|\right\}$$

即 M 的最大行和. 这个矩阵范数称为**最大范数**或**下确界范数**. 虽然谱范数 $\|M\|_S$ 更接近 $\rho(M)$,但不容易计算. 例 3.1.2 中矩阵 M 的行和分别为 0.6,0.6 和为 0.9,所以 $\|M\| = 0.9$. 这个值比弗罗贝尼乌斯范数 $\|M\|_F \doteq 0.8185$ 要大一些,M 的谱范数 $\|M\|_S \doteq 0.6419$,它们都超过了谱半径 $\rho(M)\doteq 0.4531$. 根据最大范数的这个近似,每次迭代误差将下降 90%,根据弗罗贝尼乌斯范数的近似,误差将下降 82%,根据谱范数的近似,误差将下降 64%. 由谱半径得到的近似是最正确的,约为 45%. 如果需要谱半径也是可以近似的.

如果 $\rho(M)$接近 1,那么收敛就会很慢. 能否进行加速呢? 再来看一下例 3.1.1.

例 3.1.3 再考虑方程组 $Ax=(3,-1,4)^T$,其中 $A=[4\ \ 2\ \ 1;\ \ 1\ \ 3\ \ 1;\ \ 1\ \ 1\ \ 4]$. 由(3.3)

$$x_1^{k+1} = \frac{1}{4}(3-2x_2^k-x_3^k)$$

$$x_2^{k+1} = \frac{1}{3}(-1-x_1^k-x_3^k)$$

$$x_3^{k+1} = \frac{1}{4}(4-x_1^k-x_2^k)$$

仍旧取初值 $x^0=(0,0,0)^T$,则

$$x_1^1 = \frac{1}{4}(3-2\cdot 0-0) = \frac{3}{4}$$

$$x_2^1 = \frac{1}{3}(-1-x_1^0-x_3^0)$$

此时已经得到了关于真值 x_1 的更好的近似 x_1^1,因此可以用这个好一些的近似值来替代 x_1^0. 这样

$$x_2^1 = \frac{1}{3}(-1-x_1^1-x_3^0) = \frac{1}{3}\left(-1-\frac{3}{4}-0\right) = -\frac{7}{12}$$

这个值近似 $x_2=-1$ 的效果应该比前面的 $x_2^1=-1/3$ 好一些,同样,对 x_3^1

$$x_3^1 = \frac{1}{4}(4-x_1^1-x_2^1) = \frac{1}{4}\left(-1-\frac{3}{4}-\frac{-7}{12}\right)$$

$$=\frac{23}{24}$$

$$\doteq 0.9583$$

继续这个过程，一旦得到了新的近似值便马上投入使用，这样

$$x_1^2 = \frac{1}{4}\left(3-2\cdot\frac{-7}{12}-\frac{23}{24}\right)$$

$$\doteq 0.8021$$

$$x_2^2 = \frac{1}{3}\left(-1-0.8021-\frac{23}{24}\right)$$

$$\doteq -0.9201$$

$$x_3^2 = \frac{1}{4}(4-0.8021--0.9201)$$

$$\doteq 1.0295$$

结果的每个分量都比雅可比迭代的结果 $x_2=(0.7292,\ -0.8333,\ 0.6458)^T$ 好(真解为 $x=(1,\ -1,1)^T$). ■

例 3.13 的方法叫做**高斯-塞德尔(Gauss-Seidel)迭代**，这种方法充分利用了刚刚得到的新的信息(这是一个简单而又广泛应用的思想). 高斯-塞德尔迭代所对应的矩阵分裂为

$$Ax = b$$

$$(U+L+D)x = b$$

$$(L+D)x = -Ux + b$$

$$x = -(L+D)^{-1}Ux + (L+D)^{-1}b$$

(即若$(L+D)^{-1}$存在，则 $M=-(L+D)^{-1}U$)，迭代公式为

$$x^{k+1} = -(L+D)^{-1}Ux^k + (L+D)^{-1}b \tag{3.6}$$

当然使用时并不用具体形成$(L+D)^{-1}$，而是可以利用例 3.1.3 的方式，这种方法的分量形式为

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

习惯上视空的求和为 0. 这里同样假定所有 a_{ii} 都是非零的.

一般地，如果对给定的矩阵 A 雅可比迭代收敛，则高斯-塞德尔迭代也收敛并且速度更快(高斯-塞德尔迭代矩阵的谱半径更小一些)，但有时并不是这样. 如果 A 严格对角占优，则对任意初始值高斯-塞德尔迭代都是收敛的. 两种算法都有分块形式.

通常高斯-塞德尔迭代要好于雅可比迭代，但在并行处理机上则例外. 如果 $x\in\mathbf{R}^n$ 且有 n 个处理器，则雅可比迭代效率非常高(用第 i 个处理器更新 x_i)，此时高斯-塞德尔迭代没有优势. 但由于处理器的个数 p 往往小于 n，所以加速效果并没有预想的那么好. 这时虽然每个处理器需要用高斯-塞德尔迭代的思想计算 x 的 n/p 个分量(即充分利用能够得到的变量的新值)，但却不能在处理器间传递更新的值. 关于并行高斯-塞德尔迭代有许多通用策略.

雅可比迭代或高斯-塞德尔迭代都依赖于未知量的次序. 如果线性方程组中方程的次序有所改变，则得到的雅可比或高斯-塞德尔序列也是不同的.

一直以来，求解由偏微分方程离散化以及其他的应用问题产生的大型线性方程组都使用这两种方法，但目前更常用的方法是将雅可比和高斯-塞德尔与其他迭代方法结合起来，3.3 节所讨论的方法就是这样.

习题 3.1

1. a. 用例 3.1.1 的方法迭代 3 次，比较结果的相对误差.
   b. 用例 3.1.3 的方法迭代 3 次，比较结果的相对误差.
2. a. 求例 3.1.1 中雅可比迭代矩阵的谱半径，并用谱半径估计迭代 5 次的误差，将结果与用最大范数、弗罗贝尼乌斯范数以及谱范数得到的估计相比较.
   b. 求例 3.1.3 中高斯-塞德尔迭代矩阵的谱半径，并用谱半径估计迭代 5 次的误差，将结果与用最大范数、弗罗贝尼乌斯范数以及谱范数得到的估计相比较.
   c. 将得到的估计值与习题 1(a)和(b)相比较.
3. a. 证明严格对角占优矩阵的雅可比迭代一定收敛(提示：证明对某种容易计算的范数有，$\|M\|<1$).
   b. 证明严格对角占优矩阵的高斯-塞德尔迭代一定收敛.
4. a. 证明(3.6)式与例 3.1.3 描述的高斯-塞德尔迭代的标量形式是等价的.
   b. 雅可比迭代也称为**同步替换法**，高斯-塞德尔迭代也称为**逐次替换法**，为什么？
5. a. 证明对任何诱导矩阵范数都有 $\rho(A)\leqslant\|A\|$
   b. 证明若 $\rho(M)<1$，则 M 的预解式 $(I-M)^{-1}$ 一定存在.
   c. 证明高斯-塞德尔迭代矩阵 $-(L+D)^{-1}U$ 是奇异的，对雅可比迭代矩阵是否也有这个结论？

MATLAB 3.1

原则上可以用(3.3)式 $x^{k+1}=D^{-1}(U+L)x^k+D^{-1}b$ 进行雅可比迭代，用(3.6)式 $x^{k+1}=-(L+D)^{-1}Ux^k+(L+D)^{-1}b$ 进行高斯-塞德尔迭代，但由于实际中 n 往往非常大，所以即使是求解高斯-塞德尔迭代中的三角方程组 $(L+D)x^{k+1}=-Ux^k+b$ 也要耗费太多的时间. 事实上，实际计算时根据存储器的情况以及 A 的稀疏性，通常并不显式的形成 A,L,U 或 D. 一般都是编写一个能够返回 A 的指定元素或计算 Ax(给定 x)的程序，然后按 $x_1^{k+1},\cdots,x_n^{k+1}$ 的公式进行更新.

为方便，这里仍使用迭代的矩阵形式. 输入：

```
≫ A=4*eye(10);  for i=2:9;  A(i,[i-1  i+1])=[1,  1];  end
≫ A(10, 1)=1;  A(1, 10)=1;  A(1, 2)=1;  A (10, 9)=1
```

偏微分方程的应用中经常出现这类矩阵. 输入：

```
≫ D=diag(diag (A));  L=tril( A, -1);  U=triu(A, 1 );
```

对矩阵进行分裂. 现在比较求解 $Ax=b$ 的雅可比迭代和高斯-塞德尔迭代. 输入：

```
≫ b=6*ones([10  1]);
```

这时方程的真解为全1向量.输入:

```
≫ x0j=zeros([10  1]);  x0gs=x0j;              %初始值.
≫ xj=D\(-(U+L)*x0j+b);  rj=b-A*xj             %雅可比迭代.
≫ xgs=(L+D)\(-U*x0gs+b );  rgs=b-A*xgs        %高斯-塞德尔迭代.
≫ norm(rj), norm(rgs)
≫ x0j=xj; x0gs=xgs;
```

重复若干次(每次迭代都显示残向量.)可以看到高斯-塞德尔迭代残量的缩减比雅可比迭代快得多,同时 rgs 一直有一个零分量.输入:

```
≫ alpha_j=norm(xj-ones([10  1]))
≫ alpha_gs=norm(xgs-ones([10  1]))
```

计算解的绝对误差.通常 alpha_gs 要比 alpha_j 小一些,但是小的残量并不是解的保证,再迭代几次并比较残量及绝对误差.

与解方程组(见1.5节)牛顿法的阻尼技术类似,也可以对高斯-塞德尔迭代进行加速.将高斯-塞德尔迭代改写为 $x^{k+1}=x^k+\Delta^k$,其中向量 Δ^k 是 x^k 和 x^{k+1} 的差.引入新的参数 $\tilde{\omega}$ 并记 $x^{k+1}=x^k+\tilde{\omega}\Delta^k$.如果 $\tilde{\omega}=1$,则方法就是标准的高斯-塞德尔迭代.如果 $0<\tilde{\omega}<1$,则称方法为**低松弛**,如果 $\tilde{\omega}>1$,则称方法为**超松弛**,后两种情形统称为**逐次超松弛**(或**SOR**).当高斯-塞德尔迭代发散时可以通过低松弛强迫它收敛,超松弛往往用于加速收敛.其基本思想是如果 Δ^k 是从 x^k 指向 x^{k+1} 的步长,则在这个方向上应该迈得大一些.$\tilde{\omega}$ 的值一般不容易确定,但若选择的好则可以大大提高收敛速度($\tilde{\omega}$ 选的不好还会引起发散).SOR 方法可以写成 $x^{k+1}=Mx^k+c$,其中 $M=(D+\tilde{\omega}L)^{-1}((1-\tilde{\omega})D-\tilde{\omega}U)$,$c=\tilde{\omega}(D+\tilde{\omega}L)^{-1}b$.

现在用 SOR 方法求解前面的方程组.输入:

```
≫ x0gs=zeros([10  1]);  x0SOR=x0gs;  w=1.5;
≫ xgs=(L+D)\(-U*x0gs+b);  rgs=b-A*xgs;                    %高斯-塞德尔迭代.
≫ xSOR=(D+w*L)\((1-w)*D-w*U)*x0SOR+w*(D+w*L)\b;  rSOR=b-A*xSOR
                                                          %SOR迭代.
≫ norm(rgs ), norm(rSOR)
≫ x0gs=xgs; x0SOR=xSOR;
```

迭代若干次可以看到 SOR 方法似乎不如高斯-塞德尔迭代方法.看一下两个迭代矩阵的谱半径,输入:

```
≫ max(abs(eig(L+D)\(-U))))                                %高斯-塞德尔迭代.
≫ max(abs(eig((D+w*L)\((1-w)*D-w*U))))                    %SOR迭代.
```

可以得到两种方法的 $\rho(M)$.高斯-塞德尔迭代的 $\rho(M)\doteq 0.3093$,SOR 的 $\rho(M)\doteq 0.6135$,因此 SOR 方法的效果不如高斯-塞德尔方法.输入:

```
≫ bestz=Inf;  bestw=0;
```

```
≫ for w=1:0.005:2;  z=max(abs(eig((D+w* L)\((1-w) * D-w* U))));
≫ if z<bestz;  bestz=z;  bestw=w;  end;
≫ end
≫ bestw
```

可以看到 $\tilde{\omega}$ 的最优值约为 1.07. 输入：

```
≫ w=1.07;
≫ max(abs(eig((D+w * L)\((1-w) * D-w * U))))
```

可以看到关于这个 $\tilde{\omega}$ 的 $\rho(M)\doteq 0.2335$，因而效果比高斯-塞德尔方法好得多. 将带有这个 $\tilde{\omega}$ 值的 SOR 迭代与高斯-塞德尔迭代进行比较.

可以证明，SOR 方法对任何 x^0 都收敛的必要条件是 A 的对角元素都不是零且 $0<\tilde{\omega}<2$. 如果 A 是正定矩阵，且 $0<\tilde{\omega}<2$，则 SOR 方法对任何 x^0 都收敛.（高斯-塞德尔迭代相当于 $\tilde{\omega}=1$，因此对于正定矩阵，高斯-塞德尔迭代是收敛的.）实际中通常取 $\tilde{\omega}=\tilde{\omega}_k$ 且当 $k\to\infty$ 时，$\tilde{\omega}_k\to 1$. 对某些特殊矩阵，$\tilde{\omega}$ 的最优值是可以确定的，但一般情况下 $\tilde{\omega}$ 的值只能通过试探得到. 当需要对不同的 b 反复使用矩阵时，找到这个最佳的 $\tilde{\omega}$ 是值得的，况且这种情况是经常出现的.

附加题 3.1

6. 对于给定方程组，写出 SOR 方法的 MATLAB 程序. 要求输入 A、b、x^0、$\tilde{\omega}$ 以及其他必需的参数(如误差限)，返回解的近似值.
7. a. 分别取 $\tilde{\omega}=1.1, 1.3, 1.5$，对例 3.1.1 中的方程组作 5 次雅可比 SOR 迭代. 将结果与习题 1(a)的结果进行比较.
   b. 分别取 $\tilde{\omega}=1.1, 1.3, 1.5$，对例 3.1.3 中的方程组作 5 次高斯-塞德尔 SOR 迭代，将结果与习题 1(b)的结果进行比较.
8. 用图形说明 $\rho(M)=0.3$ 的迭代法优于 $\rho(M)=0.5$ 的迭代法.
9. a. 计算(手工)$n\times n$ 单位矩阵的谱范数、弗罗贝尼乌斯范数和最大范数以及谱半径.
   b. 计算(手工)矩阵 $A=[2\quad 1;\quad 1\quad 2]$的谱范数、弗罗贝尼乌斯范数和最大范数以及谱半径.
10. 设 A 为 10×10 三对角矩阵，其中主对角元为 4，上次对角线和下次对角线元为 -1，b 是全 1 向量. 取相对误差为 10^{-6}，分别用雅可比、高斯-塞德尔和适当 $\tilde{\omega}$ 的 SOR 迭代求解 $Ax=b$ 并加以说明，其中参数 $\tilde{\omega}$ 可由试探得到.
11. 生成 20 个 100×100 随机矩阵，比较其雅可比迭代和高斯-塞德尔迭代矩阵的 $\rho(M)$ (用角本文件完成.)并加以说明.
12. a. 用实验证明 SOR 方法的 $\rho(M)\geqslant|\tilde{\omega}-1|$. 如何用这个结果证明 $0<\tilde{\omega}<2$ 是 SOR 方法收敛的必要条件？
    b. 若 A 是正定三对角矩阵，则 SOR 方法中 $\tilde{\omega}$ 的最优值满足 $\rho(M)=\tilde{\omega}-1$，用实验证明这个结论(见 MATLAB3.1).