线性方程组的求解 (含多重网格法)

匡昱潼, 薛旭恒

北京师范大学数学科学学院

2023 年 5 月 23 日

目录

1. 问题背景
2. 线性方程组的直接解法
3. 线性方程组的迭代解法
4. Krylov 子空间法

问题背景

自 1946 年第一台电子计算机问世以来, 科学与工程计算经过半个世纪的发展已经成为 20 世纪最重要的科学进步之一. 科学计算已与理论研究及科学实验并列成为当今世界科学活动的三种主要方式.
大部分科学与工程问题最终都要归结于一个矩阵计算问题, 特别是求解线性方程组的问题. 例如: 数据分析, 最优化及非线性方程组和偏微分方程数值解等. 利用电子计算机来快速、有效地求解线性方程组是非常值得研究的问题.

问题背景

一维模型

我们考虑一维边值问题:

$$\begin{cases} -\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} = f(x), & 0 < x < 1, \\ u(0) = 0, & u(1) = 0. \end{cases} \tag{1}$$

用中心差分格式代替二阶导数, 我们得到如下代数方程组:

$$\boldsymbol{L}\boldsymbol{u} = \boldsymbol{f},$$

其中,$\boldsymbol{u} = (u_1, \cdots, u_n)^T$, $\boldsymbol{f} = (f_1, \cdots, f_n)^T$,

$$\boldsymbol{L} = \frac{1}{h^2} \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

问题背景

二维模型

二维 Poisson 方程的 Dirichlet 边值问题 $(\Omega = (0,1) \times (0,1))$:

$$\begin{cases} -\Delta u = f(x,y) & \forall (x,y) \in \Omega, \\ u(x,y) = 0, & \forall (x,y) \in \partial\Omega. \end{cases} \tag{2}$$

使用五点差分法求解, 得到差分逼近解 U 所应满足的 $(N-1) \times (N-1)$ 阶线性代数方程组:

$$T_{n-1}U + UT_{n-1} = h^2 F,$$

其中:

$$U = [U_{ij}], \quad F = [f_{ij}],$$

$$T_{n-1} = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbf{R}^{(N-1)\times(N-1)}.$$

问题背景

二维模型

为了使之变为一个可以求解的线性方程组, 我们需要对 U, F 和 M 中的元素进行重排. 我们采取自然顺序排列, 即先按 j 由小到大排列,j 相同的按 i 由小到大排列. 用自然顺序排列后得到的线性方程组具有如下的形状:

$$Au = h^2 f,$$

其中,

$$A = \begin{bmatrix} T_{N-1} + 2I_{N-1} & -I_{N-1} & & \\ -I_{N-1} & \ddots & \ddots & \\ & \ddots & \ddots & -I_{n-1} \\ & & -I_{N-1} & T_{N-1} + 2I_{N-1} \end{bmatrix}.$$

易知,A 是 $(N-1)^2$ 阶方阵. 如果顶点的排列次序变了,A 的形状也要改变, 但可以证明 A 的特征值不变.

线性方程组的直接解法

我们很容易直接求解三角形方程组 (前代法, 回代法), 即

$$Ly = b,$$

其中 L 为下三角矩阵或上三角矩阵. 因此对于一般的线性方程组

$$Ax = b,$$

如果我们能够将 A 分解为 $A = LU$, 其中 L 为下三角矩阵,U 为上三角矩阵. 那么方程组可直接解出.

线性方程组的直接解法

Gauss 消去法

我们可以通过 Gauss 变换来求出 A 的三角分解, 称为 Gauss 消去法.

Algorithm 1 Gauss 消去法

for $k = 1 : n-1$ **do**

$$A(k+1:n, k) = A(k+1:n, k)/A(k, k)$$

$$A(k+1:n, k+1:n) = A(k+1:n, k+1:n) - A(k+1:n, k)A(k, k+1:n)$$

end for

该方法的运算量为 $\frac{2}{3}n^3$.

线性方程组的直接解法

Gauss 消去法成功执行的条件

当且仅当 Gauss 消去过程中的主元 $a_{kk}^{(k-1)}(k=1,2,\cdots,n-1)$ 均不为零时, 算法1才能进行到底. 我们有一个矩阵的三角分解存在的充分条件:

定理

若 $A \in \mathbb{R}^{n\times n}$ 的顺序主子阵 $A_k \in \mathbb{R}^{k\times k}(k=1,\cdots,n-1)$ 均非奇异, 则存在唯一的单位下三角阵 $L \in \mathbb{R}^{n\times n}$ 和上三角阵 $U \in \mathbb{R}^{n\times n}$, 使得 $A = LU$.

然而,A 非奇异并不能保证其顺序主子阵非奇异, 但只要 A 非奇异, 方程组就存在唯一的解. 同时, 在 Gauss 消去过程中, 若是相应的主元太小, 也会造成较大的计算误差. 因此我们需要修改 Gauss 消去法使之适应于非奇异矩阵.

线性方程组的直接解法

选主元三角分解

为了弥补 Gauss 消去法的不足, 同时控制计算代价, 人们提出了列主元 Gauss 消去法, 只要 A 非奇异, 列主元 Gauss 消去法就可以进行到底. 得到分解 $PA = LU$.

Algorithm 2 列主元 Gauss 消去法

for $k = 1 : n-1$ **do**
  确定 p $(k \leqslant p \leqslant n)$, 使得

$$|A(p,k)| = \max\{|A(i,k)| : i = k : n\}$$

  $A(k, 1:n) \leftrightarrow A(p, 1:n)$ (交换第k行和第p行)
  $u(k) = p$ (记录置换矩阵P_k)
  if $A(k,k) \neq 0$ **then**

$$A(k+1:n, k) = A(k+1:n, k)/A(k,k)$$

$$A(k+1:n, k+1:n) = A(k+1:n, k+1:n) - A(k+1:n, k)A(k, k+1:n)$$

  else
    stop (矩阵奇异)
  end if
end for

线性方程组的直接解法

平方根法

对于一般的方阵，为了消除 LU 分解的局限性和误差过分积累，采用了选主元的方法，但对于对称正定矩阵而言，选主元是不必要的.

定理 (Cholesky 分解)

若 $A \in \mathbb{R}^{n\times n}$ 对称正定，则存在一个对角元均为正数的下三角阵 $L \in \mathbb{R}^{n\times n}$，使得

$$A = LL^T.$$

上式称为 Cholesky 分解，其中的 L 称作 A 的 Cholesky 因子.

我们可以直接比较 $A = LL^T$ 两边的元素来计算 L.

Algorithm 3 平方根法

for $k = 1 : n$ **do**
  $A(k,k) = \sqrt{A(k,k)}$
  $A(k+1:n,k) = A(k+1:n,k)/A(k,k)$
  for $j = k+1 : n$ **do**
    $A(j:n,j) = A(j:n,j) - A(j:n,k)A(j,k)$
  end for
end for

该方法的运算量为 $\frac{1}{3}n^3$, 为 Gauss 消去法的一半.

线性方程组的直接解法

数值实验

考虑线性方程组如下, 其右端项为

$$b = \frac{1-a^n}{1-a}(1, 1, \cdots, 1)^{\mathrm{T}}, \tag{3}$$

系数矩阵为 $A = B + \Delta B$, 这里

$$A = \begin{bmatrix} 1 & a & a^2 & \cdots & a^{n-2} & a^{n-1} \\ a & a^2 & a^3 & \cdots & a^{n-1} & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a^{n-1} & 1 & a & \cdots & a^{n-3} & a^{n-2} \end{bmatrix}, \tag{4}$$

矩阵 ΔB 的元素是服从正态分布的随机数.

线性方程组的直接解法

数值实验

取 $a = 1.01, n = 1000$, 分别用顺序消去法和列主元消去法求解该方程组, 得到结果如下表所示.

表: 直接法解方程 1 比较

算法	CPU 运行时间	绝对残差
顺序消去法	3.08	2.20E-3
列主元消去法	3.24	3.88E-10

由于方程 1 矩阵 A 性质较好, 直接消去法和列主元消去法所得结果相同, 故仅比较列主元消去法和平方根法求解方程组的结果, 如下表所示.

表: $N = 1024$ 时直接法解方程 1 比较

算法	CPU 运行时间	绝对残差
列主元消去法	2.86	5.30E-11
平方根法	1.07	1.55E-10

线性方程组的迭代解法

在实际应用中，特别是偏微分方程数值求解时，常常遇到的是大型稀疏线性方程组的求解问题，而直接法不能保持 A 的稀疏性，因此需要寻找能够保持稀疏性的有效算法. 其中一类主要的方法是迭代法.
迭代法是按照某种规则构造一个向量序列 $\{x_k\}$，使其极限向量 x_* 是方程组 $Ax = b$ 的精确解.
而对于迭代法来说，我们需要关注以下几个问题:

1. 如何构造迭代序列?
2. 迭代序列是否收敛，或其收敛条件?
3. 收敛的速度如何?

称形如

$$x_k = Mx_{k-1} + g \tag{5}$$

的迭代法为单步线性定常迭代法, 其中 $M \in \mathbb{R}^{n\times n}$ 称为迭代矩阵,$g \in \mathbb{R}^n$ 称为常数项,$x_0 \in \mathbb{R}^n$ 称为初始向量.

定理

解线性方程组的单步线性定常迭代法(5)收敛的充分必要条件是其迭代矩阵 M 的谱半径小于 1, 即

$$\rho(M) < 1.$$

考虑非奇异线性代数方程组

$$Ax = b.$$

令

$$A = D - L - U,$$

其中

$$A = [a_{ij}], D = diag(a_{11}, a_{22}, \cdots, a_{nn}),$$

L 为 A 对角线下方元素构成的矩阵,U 为对角线上方元素构成的矩阵.

在单步线性定常迭代法(5)中, 令

$$M = D^{-1}(L+U), g = D^{-1}b;$$

即为 Jacobi 迭代法. 令

$$M = (D-L)^{-1}U, g = (D-L)^{-1}b;$$

即为 Gauss-Seidel 迭代法. 令

$$M = (D-\omega L)^{-1}[(1-\omega)D+\omega U], g = \omega(D-\omega L)^{-1}b;$$

即为 SOR 迭代法.

Jacobi 迭代法是在 $x = D^{-1}(L+U)x + D^{-1}b$ 的基础上将 x_{k-1} 带入式子右边直接产生的.
Gauss-Seidel 迭代法在计算 x_k 的分量时, 其中第一个分量仍用 x_{k-1} 的各个分量计算; 当计算 x_k 的第二个分量 $x_2^{(k)}$ 时, 用 $x_1^{(k)}$ 代替 $x_1^{(k-1)}$, 其他分量仍用 $x_i^{(k-1)}$, 依此类推得到的, 在编程时可以减少储存量.
SOR 迭代法则在 Gauss-Seidel 迭代法的基础上, 令 $\Delta x = x_{k+1} - x_k$, 则有

$$x_{k+1} = x_k + \Delta x.$$

在修正项 Δx 前面加上参数 ω, 即可得到松弛迭代法的迭代格式

$$\begin{aligned} x_{k+1} &= x_k + \omega(x_{k+1} - x_k) \\ &= (1-\omega)x_k + \omega((D-L)^{-1}Ux_k + (D-L)^{-1}b), \end{aligned}$$

Algorithm 4 Jacobi 迭代法

$x_0 =$ 初值
$k = 0; r = b - Ax; \rho = r^{\mathrm{T}} r$
$D = diag(diag(A))$
while $(\sqrt{\rho} > \varepsilon \|b\|_2)$ and $(k < k_{\max})$ **do**
  $x_{k+1} = D^{-1}((D - A)x_k + b)$
  $k = k + 1$
  $r = b - Ax; \rho = r^{\mathrm{T}} r$
end while

Algorithm 5 G-S 迭代法

$x_0 =$ 初值
$k = 0; r = b - Ax; \rho = r^{\mathrm{T}} r$
$D = tril(A); U = -triu(A, 1)$
while $(\sqrt{\rho} > \varepsilon \|b\|_2)$ and $(k < k_{\max})$ **do**
  $x_{k+1} = D^{-1}(Ux_k + b)$
  $k = k + 1$
  $r = b - Ax; \rho = r^{\mathrm{T}} r$
end while

Algorithm 6 SOR 迭代法

$x_0 =$ 初值　给定ω
$k = 0; r = b - Ax; \rho = r^{\mathrm{T}} r$
$D = diag(diag(A)); L = -tril(A, -1); U = -triu(A, 1)$
while $(\sqrt{\rho} > \varepsilon \|b\|_2)$ and $(k < k_{\max})$ **do**
　$x_{k+1} = (D - \omega L)^{-1}(((1 - \omega)D + \omega U)x_k + \omega b)$
　$k = k + 1$
　$r = b - Ax; \rho = r^{\mathrm{T}} r$
end while

线性方程组的迭代解法

收敛性分析

定理

若线性方程组的系数矩阵 A 对称, 而且其对角元 $a_{ii} > 0 (i = 1, \cdots, n)$, 则 *Jacobi* 迭代法收敛的充分必要条件是 A 和 $2D - A$ 都正定.

定理

若线性方程组的系数矩阵 A 是对称正定的, 则 *G-S* 迭代法收敛.

定理

若线性方程组的系数矩阵 A 是严格对角占优的或不可约对角占优的, 则 *Jacobi* 迭代法和 *G-S* 迭代法都收敛.

定理

SOR 迭代法收敛的必要条件是 $0 < \omega < 2$.

定理

若线性方程组的系数矩阵 A 是严格对角占优的或不可约对角占优的, 且 $\omega \in (0, 1)$, 则 *SOR* 迭代法收敛.

定理

若线性方程组的系数矩阵 A 是实对称的正定矩阵, 则当 $0 < \omega < 2$ 时,*SOR* 迭代法收敛.

线性方程组的迭代解法

数值实验 1

我们考虑二维模型问题(2), 其中

$$A = \begin{bmatrix} T_{N-1} + 2I_{N-1} & -I_{N-1} & & \\ -I_{N-1} & \ddots & \ddots & \\ & \ddots & \ddots & -I_{n-1} \\ & & -I_{N-1} & T_{N-1} + 2I_{N-1} \end{bmatrix}.$$

系数矩阵 A 具有以下几个特点:

- A 是块三对角矩阵, 共有五条对角线上有非零元素;
- A 是不可约对角占优的;
- A 是对称正定的, 而且是稀疏的.

由此, 问题 $Ax = f$ 可以使用 $Jacobi$ 迭代法,$G - S$ 迭代法, 共轭梯度法进行求解.

线性方程组的迭代解法

数值实验 1

A 的特征值为:

$$\lambda_{pq} = 2\left(2 - \cos\frac{p\pi}{N} - \cos\frac{q\pi}{N}\right),$$

若使用 Jacobi 迭代法计算问题, 其迭代矩阵为 $M = I - \frac{1}{4}A$, 所以 M 的特征值为:

$$\mu_{pq} = 1 - \frac{1}{4}\lambda_{pq} = \frac{1}{2}\left(\cos\frac{p\pi}{N} + \cos\frac{q\pi}{N}\right),$$
$$p, q = 1, \cdots, N-1.$$

于是 $\rho(M) = \cos\frac{\pi}{N} = \cos h\pi$, 因此有 $Jacobi$ 迭代法时的渐进收敛速度:

$$R_\infty(B) = -\ln\rho(B) = -\ln\cos h\pi \sim \frac{1}{2}\pi^2 h^2, \quad h \to 0.$$

类似地分析,$G-S$ 迭代法在此问题上的渐进收敛速度为:

$$R_\infty(L_1) = -\ln\rho(L_1) = -2\ln\rho(B) \sim \pi^2 h^2, \quad h \to 0.$$

取误差上限为 1×10^{-10}, 比较 Jacobi 迭代法、G-S 迭代法和 SOR 迭代法求解二维问题(2)的结果, 如下表所示.

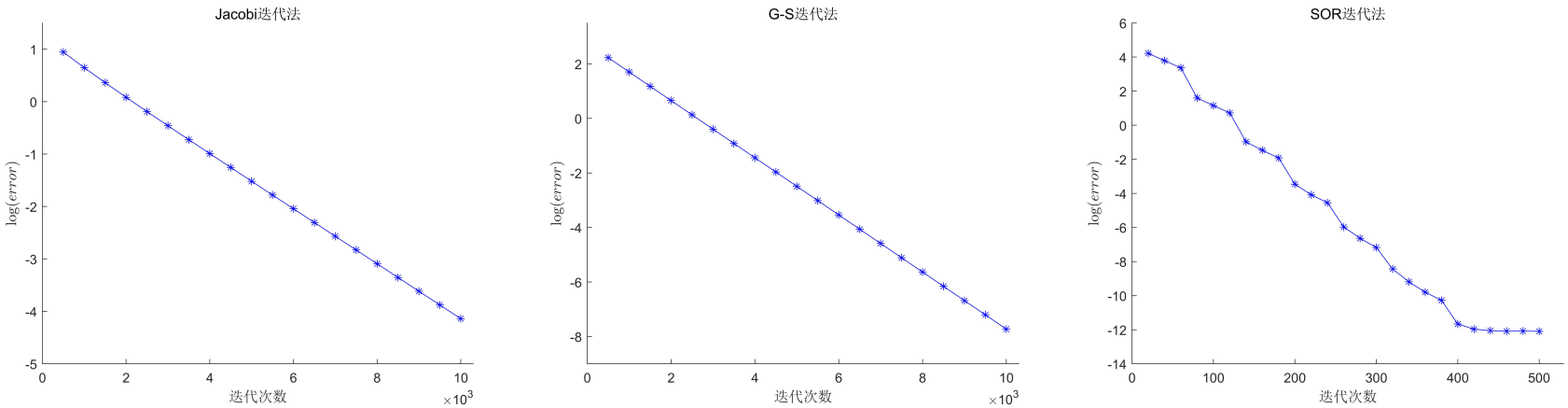
表: $N = 64$ 时迭代法解二维问题比较

迭代法	迭代次数	CPU 运行时间	相对残差
Gauss 迭代法	24337	1.9645	9.99E-11
G-S 迭代法	12170	0.7605	9.97E-11
SOR 迭代法	370	0.0543	9.73E-11

线性方程组的迭代解法

数值实验 1

分别比较 Jacobi 迭代法、G-S 迭代法和 SOR 迭代法求解二维问题(2)时残差随迭代次数的变化关系, 如下图所示.



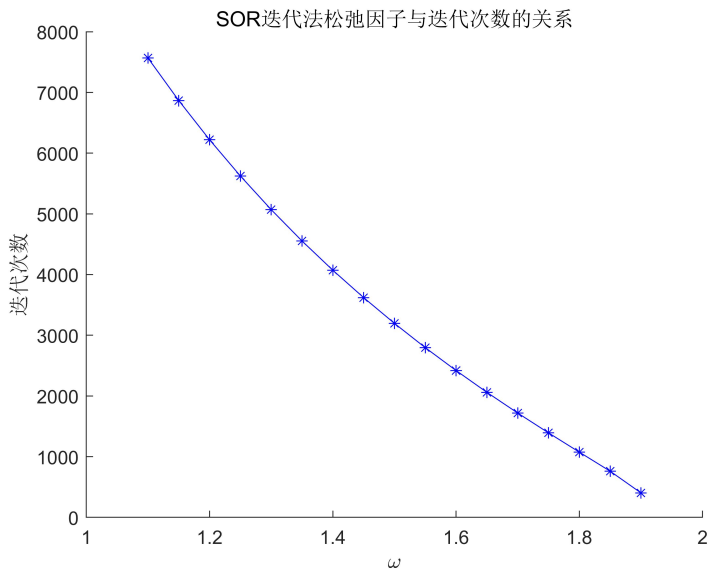
(a) Jacobi 迭代法　(b) G-S 迭代法　(c) SOR 迭代法

图: 迭代次数与残差关系图

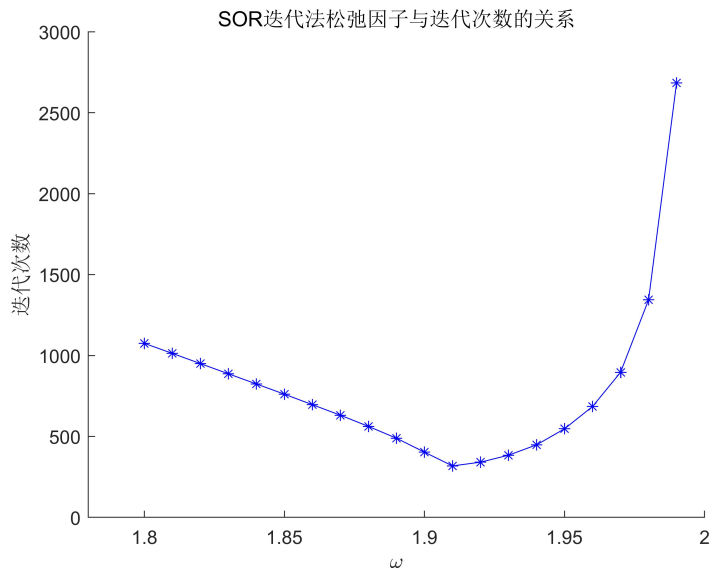
线性方程组的迭代解法

数值实验 1

在使用 SOR 迭代法解二维问题(2)时，设置误差上限为 1×10^{-10}，改变 ω 的取值，探究其与迭代次数的关系，所得结果如下图所示.



(a) $1.1 \leqslant \omega \leqslant 1.9$



(b) $1.80 \leqslant \omega \leqslant 1.99$

图: SOR 迭代法松弛因子与迭代次数的关系

下面我们更进一步分析迭代法的收敛性质, 以一维问题(1)为例. 为了能更清晰的看到传统迭代法的不足，我们引入 Jacobi 松弛迭代法:

$$x_{j+1} = x_j - \omega \left(D^{-1} A x_j - D^{-1} b \right).$$

其迭代矩阵和特征值为:

$$M = I - \omega h^2 A$$

$$\lambda_k(\omega) = 1 - 4\omega \sin^2 \frac{kh\pi}{2}, k = 1, 2, \cdots, N-1.$$

并设其相应特征向量为 $\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_{N-1}$.

现在我们考虑迭代法求解方程组 $Ax = b$ 的初始向量是 x_0, 其误差按特征向量展开为:

$$x_0 - x^* = \sum_{k=1}^{N-1} \alpha_k \boldsymbol{e}_k,$$

迭代 v 步后, 误差向量成为:

$$x_v - x^* = \sum_{k=1}^{N-1} \lambda_k^v(\omega) \alpha_k \boldsymbol{e}_k,$$

现在我们固定 $\omega = \frac{1}{4}$, 则此时 $\lambda_k(\frac{1}{4}) = 1 - \sin^2 \frac{kh\pi}{2}$. 可以看到, 对于高频误差项, 迭代法的效果很好. 但对于低频分量, 迭代速度就很慢.

我们可以考虑一个网格序列 $\{\Omega_m\}_{m=0}^M$, 注意到细网 Ω_{m+1} 上的误差向量的低频分量对应到粗网 Ω_m 相当于频率变高了, 因此如果我们可以将两层网格一并来考虑.
我们接下来考虑第 m 层和第 $m+1$ 层网格上的方程组:

$$\boldsymbol{L}_m\boldsymbol{u}_m = \boldsymbol{f}_m, \boldsymbol{L}_{m+1}\boldsymbol{u}_{m+1} = \boldsymbol{f}_{m+1}.$$

引入二重网格方法.

第 1 步　以 u_{m+1}^{old} 为迭代初值, 对方程组 $\boldsymbol{L}_{m+1}\boldsymbol{u}_{m+1} = \boldsymbol{f}_{m+1}$ 迭代 ν_1 次 (可以是 Jacobi-ω 迭代, 或 SOR 等, 称松弛 ν_1 次), 结果记为

$$\bar{\boldsymbol{u}}_{m+1} = \text{Relax}^{\nu_1}(\boldsymbol{u}_{\text{m+1}}^{\text{old}}, \boldsymbol{L}_{\text{m+1}}, \boldsymbol{f}_{\text{m+1}}).$$

第 2 步　余量计算

$$\boldsymbol{r}_{m+1} = \boldsymbol{f}_{m+1} - \boldsymbol{L}_{m+1}\bar{\boldsymbol{u}}_{m+1}.$$

第 3 步　余量限制

$$\boldsymbol{r}_m = \boldsymbol{I}_{m+1}^m \boldsymbol{r}_{m+1}.$$

第 4 步　解粗网方程组

$$\boldsymbol{L}_m \boldsymbol{\omega}_m = \boldsymbol{r}_m.$$

第 5 步　校正量延拓

$$\tilde{\boldsymbol{\omega}}_{m+1} = \boldsymbol{I}_m^{m+1}\boldsymbol{\omega}_m.$$

第 6 步　粗网校正

$$\bar{\bar{\boldsymbol{u}}}_{m+1} = \bar{\boldsymbol{u}}_{m+1} + \tilde{\boldsymbol{\omega}}_{m+1}.$$

第 7 步　后光滑迭代: 以 $\bar{\boldsymbol{u}}_{m+1}$ 为初值, 再对方程组做 ν_2 次松弛迭代, 记为

$$\boldsymbol{u}_{m+1}^{\text{new}} = \text{Relax}^{\nu_2}(\boldsymbol{u}_{\text{m+1}}, \boldsymbol{L}_{\text{m+1}}, \boldsymbol{f}_{\text{m+1}}).$$

我们可以将第 2 步到第 6 步的粗网矫正步骤记作一个迭代:

$$\bar{\bar{\boldsymbol{u}}}_{m+1} = \bar{\boldsymbol{u}}_{m+1} - \boldsymbol{I}_m^{m+1}\boldsymbol{L}_m^{-1}\boldsymbol{I}_{m+1}^m\left(\boldsymbol{L}_{m+1}\bar{\boldsymbol{u}}_{m+1} - \boldsymbol{f}_{m+1}\right)$$

但尽管粗网矫正迭代拥有很好的效果，我们可以证明这个迭代并不收敛，因此我们需要结合光滑迭代和粗网矫正迭代.

引理

第 m 层网格和第 $m+1$ 层网格的双重网格迭代法的迭代矩阵为:

$$M_{m+1}(\nu) = \left(I - I_m^{m+1} L_m^{-1} I_{m+1}^m L_{m+1}\right) S_{m+1}^\nu, \quad S_{m+1}^\nu = I - \omega h_{m+1}^2 L_{m+1}.$$

定理

$$\rho(M_{m+1}(\nu)) = \varrho_\nu - O(h_{m+1}^2), \|M_{m+1}(\nu)\| = \zeta_\nu - O(h_{m+1}^2), \quad \varrho_\nu, \zeta_\nu < 1.$$

其中 ϱ_ν, ζ_ν 的值与 m 无关, 仅与 ν 的选择有关,$\rho(M_{m+1}(\nu)), \|M_{m+1}(\nu)\|$ 虽然与 m 有关, 但也会随着 m 的增加很快达到 ϱ_ν, ζ_ν.
而关于 ν 的选取, 虽然 ϱ_ν, ζ_ν 会随着 ν 的增大而减小, 但并不是以指数形式减小, 而是以 $\frac{C}{\nu}$ 的速度. 由于 ν 的增大也会增加计算量, 取两倍的 ν 可以使 $\zeta_{2\nu} \approx \frac{\zeta_\nu}{2}$, 但也会使计算量几乎加倍. 因此 ν 不宜取太大, 如果 $\zeta_\nu^2 \leqslant \frac{\zeta_\nu}{2}, i.e. \zeta_\nu \leqslant \frac{1}{2}$,$\nu$ 就比 2ν 要好.

对于第 4 步粗网格上方程的求解,$L_m\omega_m = r_m$, 这一步的解 ω_m 的作用是计算细网格上方程的解 ω_{m+1} 的一个估计 $\tilde{\omega}_{m+1}$, 所以实际上我们也不需要 ω_m 的精确解, 可以转而求 ω_m 的估计 $\tilde{\omega_m}$. 因此我们可以用一些迭代的过程来得到 $\tilde{\omega_m}$:

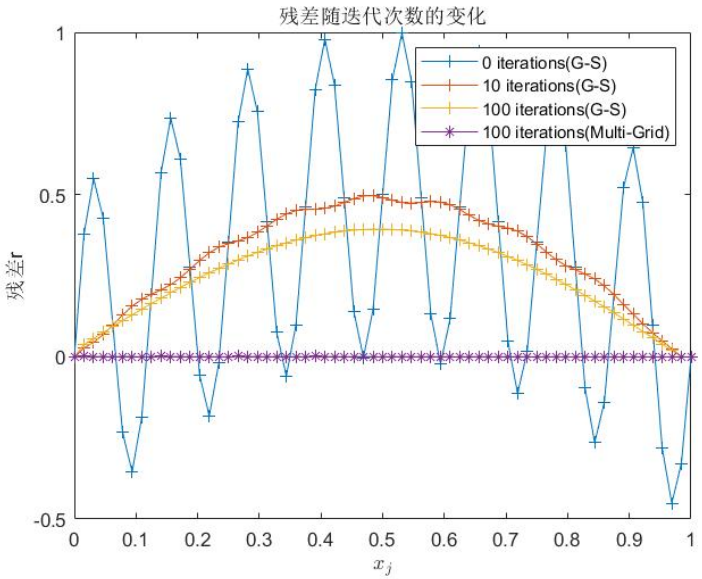
$$\omega_m^0 \mapsto \omega_m^1 \mapsto \cdots \mapsto \omega_m^\gamma = \tilde{\omega_m}.$$

从而我们可以继续考虑使用类似双重网格的思想求解方程 $L_m\omega_m = r_m$, 一直到网格最粗的时候, 这时候可以直接显式解出方程. 因此我们常常使用多重网格法, 根据 γ 选取的不同, 我们有不同形状的多重网格, 如 V 型网格,W 型网格.

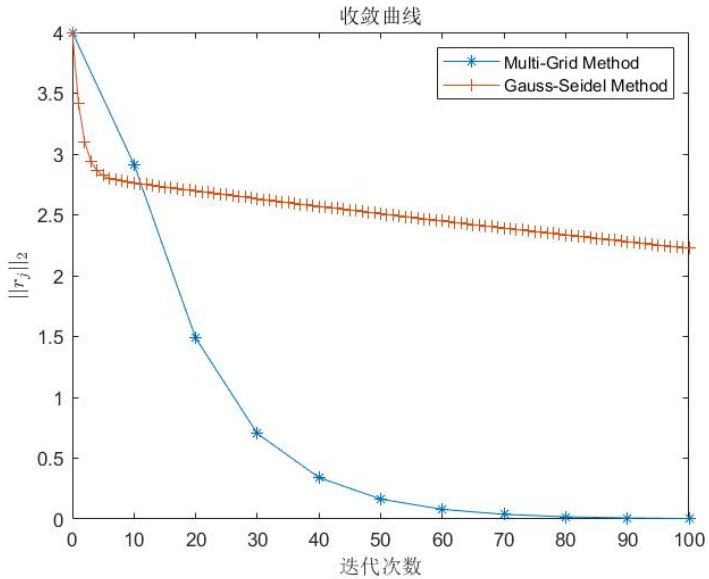
线性方程组的迭代解法

数值实验 2-多重网格

下图给出了 $N = 64$ 时使用 V 型多重网格迭代和 G-S 迭代求解一维问题(1)的收敛特性:



(a) 残差随迭代次数的变化



(b) 多重网格法与 G-S 迭代法的收敛曲线

图: 多重网格法与 G-S 迭代法求解一维模型的对比

下表给出了不同光滑迭代次数对于多重网格法求解效率的影响，残差界设置为 1×10^{-10}.

表: $N = 64$ 时不同光滑迭代次数 ν 对多重网格法的影响

光滑迭代次数 ν	迭代次数	CPU 运行时间	相对残差
1	35	0.0767	5.55E-11
2	9	0.0127	7.99E-11
3	6	0.0138	7.69E-11
4	6	0.0150	6.40E-11

下图给出了 $N = 64$ 时使用 V 型多重网格迭代和 G-S 迭代求解二维问题(2)的收敛特性:

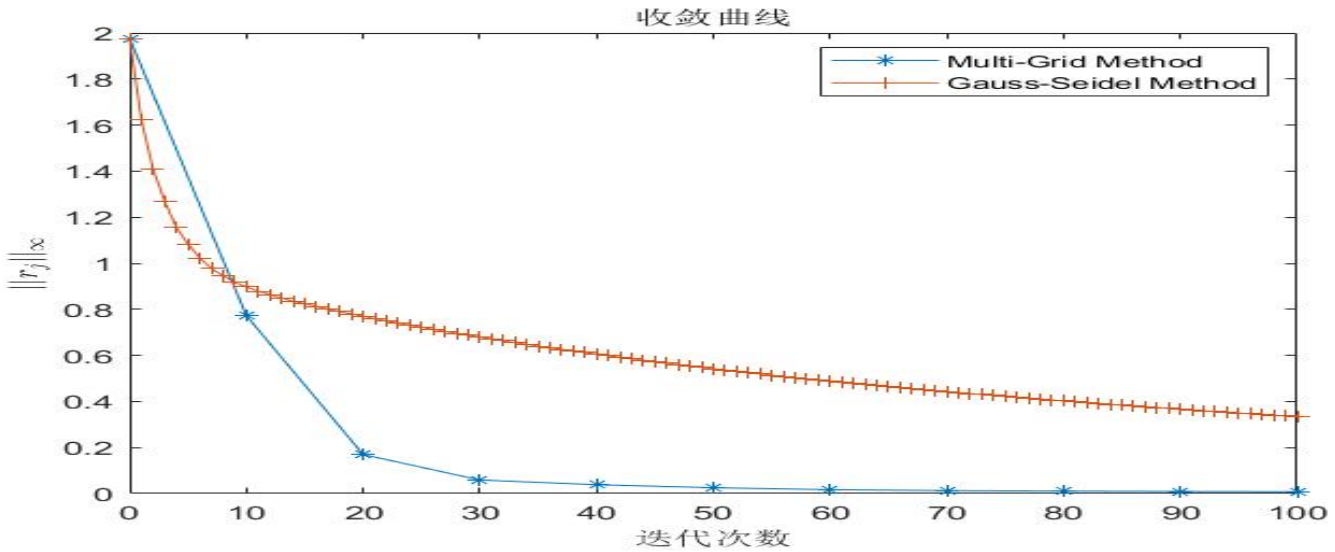


图: 多重网格法与 G-S 迭代法求解二维模型的对比

线性方程组的迭代解法

数值实验 2-多重网格

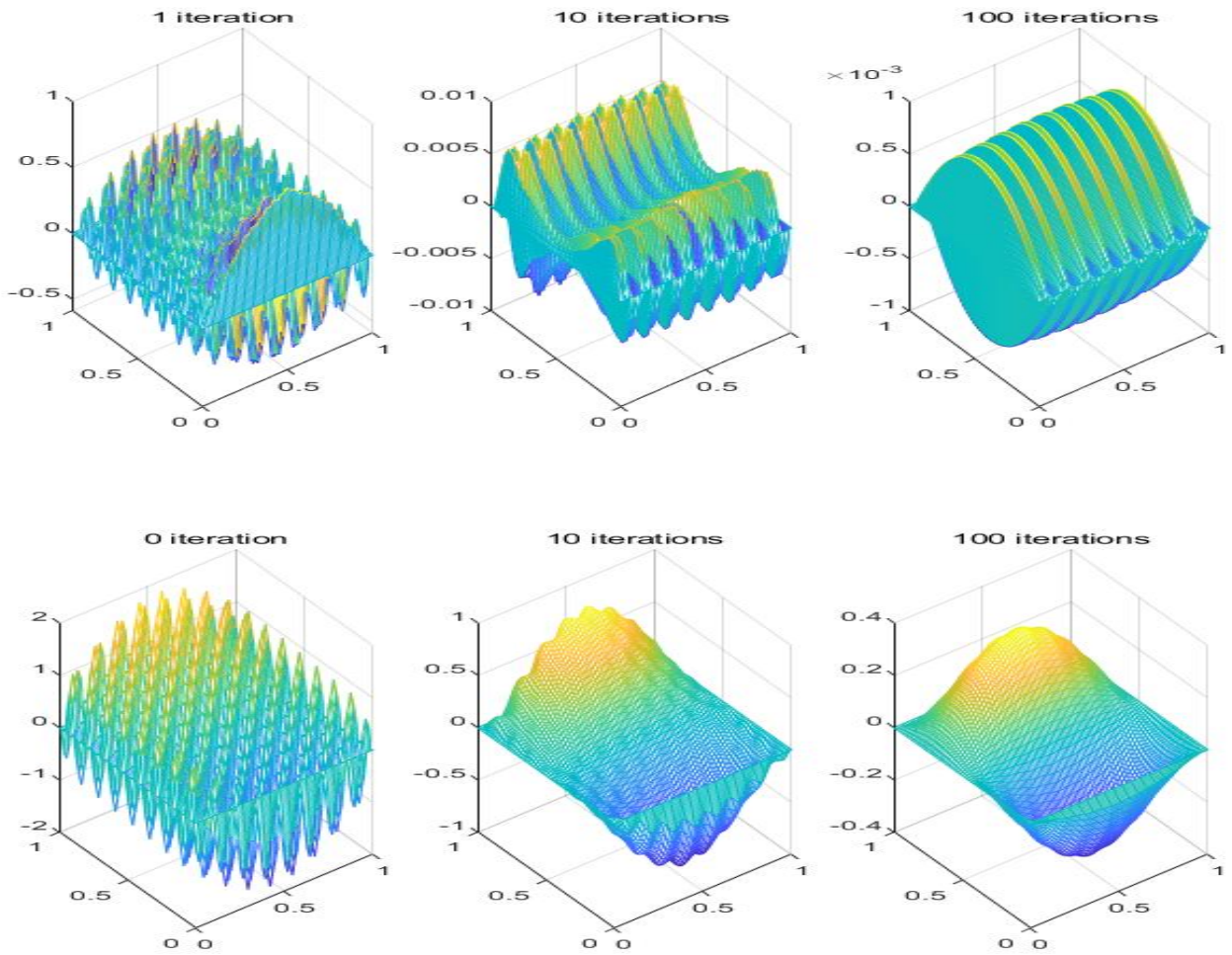


图: 多重网格法 (上方三图) 与 G-S 迭代法 (下方三图) 求解二维模型的对比

Krylov 子空间法

迭代法求解线性方程组具有很多麻烦，例如最佳松弛因子 ω_{opt} 的确定，特别是系数矩阵性质不好的时候，判断这个矩阵能否用迭代法求解都将变得困难，下面我们从最优化问题的角度介绍一类目前求解大型稀疏线性方程组最受欢迎的一类方法.

考虑线性方程组 $Ax = b$ 的求解问题，定义二次泛函

$$\varphi(x) = x^{\mathrm{T}} Ax - 2b^T x.$$

在 A 对称正定的情况下，求解方程组 $Ax = b$ 的解等价于求二次泛函 $\varphi(x)$ 的极小值点.

Krylov 子空间法

对于二次泛函 $\varphi(x)$ 的极小值点的求解, 可先给定一个初始向量 x_0 和一个搜索方向 p_0, 在直线 $x = x_0 + \alpha p_0$ 找一个点

$$x_1 = x_0 + \alpha_0 p_0$$

使得对所有实数 α, 有

$$\varphi(x_0 + \alpha_0 p_0) \leqslant \varphi(x_0 + \alpha p_0).$$

之后, 重复上述过程, 先在 x_k 点确定搜索方向 p_k, 再在直线 $x = x_k + \alpha p_k$ 上确定步长 αp_k, 使得

$$\varphi(x_k + \alpha_k p_k) \leqslant \varphi(x_k + \alpha p_k).$$

最后求出 $x_{k+1} = x_k + \alpha_k p_k$.

下面介绍一种确定搜索方向和步长的算法: 最速下降法.
先假定已经选定搜索方向 p_k, 则令

$$f(\alpha) = \varphi(x_k + \alpha_k p_k),$$

通过求解使得 $f(\alpha)$ 最小的 α, 我们得到对应的最佳步长为

$$\alpha_k = \frac{r_k^{\mathrm{T}} p_k}{p_k^{\mathrm{T}} A p_k}.$$

其中 $r_k = b - Ax_k$. 再确定搜索方向 p_k, 采取最简单直观的做法, 令 p_k 为负梯度方向, 即 $p_k = r_k$.

Algorithm 7 最速下降法

$x_0 =$ 初值
$r_0 = b - Ax_0;\ k = 0$
while $r_k \neq 0$ **do**
  $k = k + 1$
  $\alpha_{k-1} = r_{k-1}^{\mathrm{T}} r_{k-1} / r_{k-1}^{\mathrm{T}} A r_{k-1}$
  $x_k = x_{k-1} + \alpha_{k-1} r_{k-1}$
  $r_k = b - Ax_k$
end while

定理

设 A 的特征值为 $0 < \lambda_1 \leqslant \cdots \leqslant \lambda_n$, 则由上述算法产生的序列 $\{x_k\}$ 满足

$$\|x_k - x_*\|_A \leqslant (\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1})^k \|x_0 - x_*\|_A,$$

其中 $x_* = A^{-1}b, \|x\|_A = \sqrt{x^{\mathrm{T}} A x}$.

可以看到最速下降法的收敛快慢由 $\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}$ 的大小决定. 如果 $\lambda_1 \ll \lambda_n$, 收敛速度将非常慢.

由于负梯度方向从整体上看并非最佳搜索方向, 考虑另一种寻找搜索方向的方法: 共轭梯度法.
给定初始向量 x_0, 第一步仍取 $p_0 = r_0$, 于是得到 α_0, x_1, r_1 同最速下降法. 对以后各步, 在过点 x_k 由向量 r_k 和 p_{k-1} 所张成的二维平面

$$\pi_2 = \{x = x_k + \xi r_k + \eta p_{k-1} : \xi, \eta \in \mathbb{R}\}$$

内找出使泛函 φ 下降最快的方向作为新的下山方向 p_k. 之后, 确定 α_k, 方法同最速下降法. 最后计算 $x_{k+1} = x_k + \alpha_k p_k$.

考虑 φ 在 π_2 上的限制 $\psi(\xi,\eta)=\varphi(x_k+\xi r_k+\eta p_{k-1})$, 可计算求得

$$p_k = r_k + \beta_{k-1} p_{k-1}$$

, 其中 $\beta_{k-1}=\frac{\eta_0}{\xi_0}, (\xi_0,\eta_0)$ 是使得 $\psi(\xi,\eta)$ 达到极小点时的取值.
综合上述分析, 给出如下计算公式.

$$\alpha_k = \frac{r_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} A p_k}, \qquad x_{k+1} = x_k + \alpha_k p_k,$$

$$r_{k+1} = b - A x_{k+1}$$

$$\beta_k = \frac{r_{k+1}^{\mathrm{T}} r_{k+1}}{r_k^{\mathrm{T}} r_k}, \qquad p_{k+1} = r_{k+1} + \beta_k p_k.$$

Algorithm 8 共轭梯度法

$x =$ 初值
$k = 0; r = b - Ax; \rho = r^{\mathrm{T}} r$
while $(\sqrt{\rho} > \varepsilon \|b\|_2)$ and $(k < k_{\max})$ **do**
  $k = k + 1$
  if $k = 1$ **then**
    $p = r$
  else
    $\beta = \rho / \tilde{\rho}; p = r + \beta p$
  end if
  $\omega = Ap; \alpha = \rho / p^{\mathrm{T}} \omega; x = x + \alpha p$
  $r = r - \alpha \omega; \tilde{\rho} = \rho; \rho = r^{\mathrm{T}} r$
end while

定理

如果 $A = I + B$, 而且 $\mathrm{rank}(B) = r$, 则共轭梯度法至多迭代 $r + 1$ 步即可得到线性方程组的精确解.

定理

用共轭梯度法求得的 x_k 有如下的误差估计:

$$\|x_k - x_*\|_A \leqslant 2(\frac{\sqrt{\kappa_2} - 1}{\sqrt{\kappa_2} + 1})^k \|x_0 - x_*\|_A,$$

其中 $\kappa_2 = \kappa_2(A) = \|A\|_2 \|A^{-1}\|_2$.

定理

由共轭梯度法得到的向量组 $\{r_i\}$ 和 $\{p_i\}$ 具有下面的性质:

1. $p_i^T r_j = 0, 0 \leqslant i < j \leqslant k;$
2. $r_i^T r_j = 0, i \neq j, 0 \leqslant i, j \leqslant k;$
3. $p_i^T A p_j = 0, i \neq j, 0 \leqslant i, j \leqslant k;$
4. $span\{r_0, \cdots, r_k\} = span\{p_0, \cdots, p_k\} = \mathcal{K}(A, r_0, k+1)$, 其中

$$\mathcal{K}(A, r_0, k+1) = span\{r_0, Ar_0, \cdots, A^k r_0\},$$

通常称之为 *Krylov* 子空间.

定理

用共轭梯度法计算得到的近似解 x_k 满足

$$\varphi(x_k) = \min\{\varphi(x) : x \in x_0 + \mathcal{K}(A, r_0, k)\},$$

或

$$\|x_k - x_*\|_A = \min\{\|x - x_*\|_A : x \in x_0 + \mathcal{K}(A, r_0, k)\},$$

其中,$\|x\|_A = \sqrt{x^T A x}$, x_* 是方程组 $Ax = b$ 的解.

共轭梯度法只能用于求解对称正定线性方程组, 但我们可以将共轭梯度法的思想推广到一般情形. 此时,φ 不一定有最小值, 但我们可以求 $x_k \in x_0 + \mathcal{K}(A, r_0, k)$, 使得:

$$\|b - Ax_k\|_2 = \min\left\{\|b - Ax\|_2 : x \in x_0 + \mathcal{K}(A, r_0, k)\right\}.$$

基于这种思路, 我们可以得出用于求解非对称线性方程组的广义极小残量法, 即 GMRES 方法.

使用基于 Gram-Schmidt 方法的 Arnoldi's 方法计算 Krylov 子空间 $span\{v_1, Av_1, \cdots, A^{k-1}v_1\}$ 的一组 l_2 标准正交基 $\{v_1, v_2, \cdots, v_k\}$. 令 V_k 是列为 $\{v_1, v_2, \cdots, v_k\}$ 的 $N \times k$ 矩阵,$\overline{H}_k$ 是非零元为 h_{ij} 的 $(k+1) \times k$ 矩阵.

$$h_{i,j} = (Av_j, v_i), \quad i = 1, 2, \cdots, j,$$
$$h_{j+1,j} = \|\hat{v}_{j+1}\|.$$

从而我们有如下关系:

$$AV_k = V_{k+1}\overline{H}_k.$$

现在我们想要求解如下最小二乘问题:

$$\min_{z\in K_k} \|b - A[x_0 + z]\| = \min_{z\in K_k} \|r_0 - Az\|,$$

令 $z = V_k y$, 我们可以将这个问题视为关于 y 的函数:

$$J(y) = \|\beta v_1 - AV_k y\|,$$

其中 $\beta = \|r_0\|, v_1 = \frac{r_0}{\|r_0\|}$. 因此我们有:

$$J(y) = \left\|V_{k+1}\left[\beta e_1 - \bar{H}_k y\right]\right\|,$$

由于 V_{k+1} 是正交矩阵, 我们有:

$$J(y) = \left\|\beta e_1 - \bar{H}_k y\right\|.$$

Algorithm 9 GMRES 方法

1. 选定 x_0, 计算 $r_0 = f - Ax_0, v_1 = r_0/\|r_0\|$.
2. 迭代: 对 $j = 1, 2, \cdots, k, \cdots$, 直到满足 $|h_{j+1,j}| < \varepsilon$:
   $h_{i,j} = (Av_j, v_i), i = 1, 2, \cdots, j,$
   $\hat{v}_{j+1} = Av_j - \sum_{i=1}^{j} h_{i,j}v_i,$
   $h_{j+1,j} = \|\hat{v}_{j+1}\|$
   $v_{j+1} = \hat{v}_{j+1}/h_{j+1,j}.$
3. 最后, 得到近似解 $x_k = x_0 + V_k y_k$, 这里 y_k 是使得 $\left\|\beta e_1 - \bar{H}_k y\right\|, y \in \mathbb{R}^k$ 最小的值.

注: 最小二乘问题 $\min_y \left\|\beta e_1 - \bar{H}_k y\right\|$ 的求解可以使用 Givens 变换.

GMRES 方法的优点 (特点):

1. 对于 Givens 变换求得的单位矩阵 $Q_k, Q_k\bar{H}_k = R_k$, 我们有:

$$J(y) = \left\|\beta e_1 - \bar{H}_k y\right\| = \left\|Q_k[\beta e_1 - \bar{H}_k y]\right\| = \left\|g_k - R_k y\right\|.$$

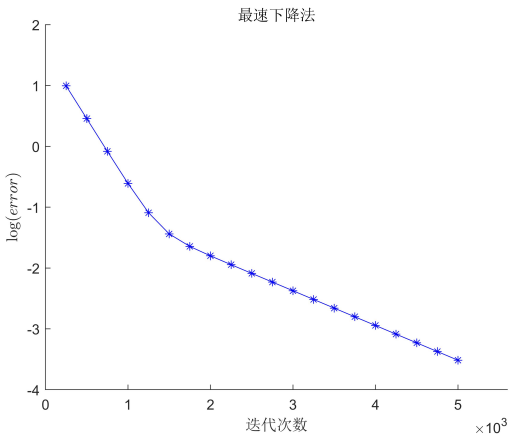
   g_k 的最后一个元素 (即第 $k+1$ 个元素) 即为我们需要判断的残差 $J(y_k)$. 因此不需要额外的计算.
2. GMRES 方法不会中断, 除非得到了精确解.
3. 对于一个 $N \times N$ 的问题,GMRES 方法至多进行到第 N 步即可得到方程的解.
4. 对于不断增加的 N,GMRES 方法所需要的存储空间和计算规模会不断增大, 可以使用重启动的 GMRES 方法 GMRES(m) 来克服这个问题.

取误差上限为 1×10^{-10}, 比较最速下降法, 共轭梯度法和 GMRES 法求解二维问题(2)的结果, 如下表所示.

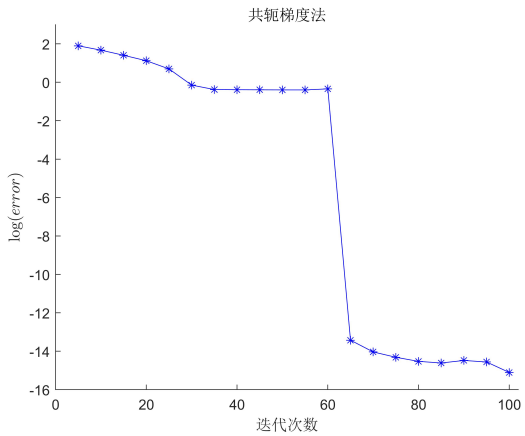
表: $N = 64$ 时 Krylov 子空间法解二维问题比较

算法	迭代次数	CPU 运行时间	相对残差
最速下降法	16383	0.6366	9.98E-11
共轭梯度法	63	0.0032	9.14E-15
GMRES 法	63	0.0341	3.06E-13

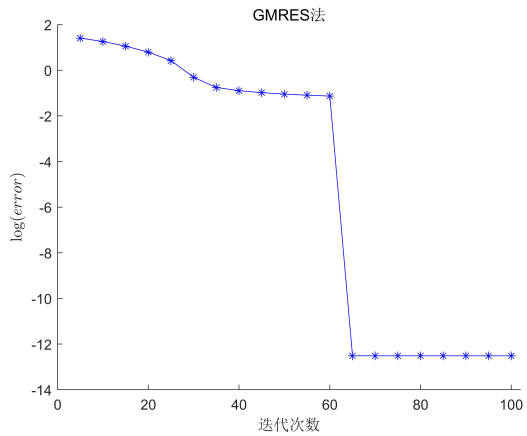
分别利用最速下降法, 共轭梯度法和 GMRES 法求解二维问题(2), 观察其残差随迭代次数的变化关系, 得到结果如下图.



(a) 最速下降法



(b) 共轭梯度法



(c) GMRES 法

图: 迭代次数与残差关系图

考虑线性方程组如下，其系数矩阵 A 和右端项分别为

$$A = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & n \\ 1 & 2 & -1 & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & 1 & n-1 & -1 \\ -n & 0 & \cdots & 0 & 1 & n \end{bmatrix}, \quad b = A \begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \\ 1 \end{bmatrix} \tag{6}$$

分别用共轭梯度法和 GMRES 方法求解该方程组, 得到结果如下表所示.

表: 共轭梯度法与 GMRES 方法比较

算法	迭代次数	绝对残差
共轭梯度法	1000	1.07E+3
GMRES 方法	221	9.60E-11

致谢

谢谢!