

# 一种加权的 Simpler GMRES 算法

杨圣炜, 卢琳璋\*

(厦门大学数学科学学院, 福建 厦门 361005)

**摘要:** GMRES 方法是求解大规模非对称稀疏线性方程组最常用的方法, 实际应用中存在着许多对标准 GMRES 进行改进的算法, 比如 Simpler GMRES 和 Weighted GMRES. Simpler GMRES 通过改进 GMRES 中基的生成过程来减小计算量, 同时保持较好的收敛性, Weighted GMRES 是采用加权技术来加快 GMRES 方法的收敛速度, 但是增加了计算量. 本文提出了一种新称为 Weighted Simpler GMRES 的方法, 它以 Simpler GMRES 方法为基础, 结合 Weighted GMRES 方法得到. 实验表明, 对某些问题, Weighted Simpler GMRES 方法的收敛性优于 Simpler GMRES 和 GMRES, 计算量小于 Weighted GMRES.

**关键词:** GMRES; Simpler GMRES; Weighted GMRES

中图分类号: O 241

文献标识码: A

文章编号: 0438-0479(2008)04-0484-05

本文讨论如何求解一个大规模稀疏线性系统

$$Ax = b \quad (1)$$

其中  $A \in R^{n \times n}$  是稀疏且非奇异.

目前已经出现了许多求解式(1)的方法, 其中一大类就是 Krylov 子空间方法. 而 Krylov 方法中使用最广泛的是 GMRES(Generalized minimum residual method) 算法<sup>[1]</sup>, GMRES 对于大型稀疏非对称矩阵特别有效, 同时在科学和工程计算中应用广泛. 另一种著名的 Krylov 方法就是 QMR 算法<sup>[2]</sup>.

1994 年, Walker 在文献[3]中对 GMRES 方法进行了变形, 提出了 Simpler GMRES(以下简称 SGMRES)方法. SGMRES 方法的优点在于在每一次迭代的最后避免了求解一个最小二乘问题, 这样可以减少计算量, 但是它的收敛效果没有 GMRES 出色(可能会出现停滞的现象). 1998 年, Essai 在文献[4]中对 GMRES 方法进行了改进, 提出了 Weighted GMRES(以下简称 WGMRES)方法. WGMRES 的优点在于有良好的收敛效果, 但是计算量增大了. 本文结合 WGMRES 的优点对 SGMRES 进行改进, 进而提出了一种新的, 称为 Weighted Simpler GMRES(以下简称 WSGMRES)的方法.

## 1 GMRES, SGMRES 和 WGMRES

GMRES 方法是利用 Arnoldi 过程生成一组正交

基  $v_1, v_2, \dots, v_k$ . 令  $V_k = [v_1, v_2, \dots, v_k]$ ,

$$\text{span}\{v_1, v_2, \dots, v_k\} = K_k(A, v_1) = \text{span}\{v_1, Av_1, A^2v_1, \dots, A^{k-1}v_1\}.$$

则 Arnoldi 过程满足以下等式:

$$AV_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T,$$

其中  $H_k = (h_{i,j})$  是一个  $k$  阶的上 Hessenberg 矩阵,  $e_k$  是  $k$  维向量  $(0, 0, \dots, 0, 1)^T$ . GMRES 方法就是寻找式(1)的近似解  $x_k = x_0 + z_k$ , 其中  $z_k \in K_k(A, r_0)$ , 使得,

$$r_k = b - Ax_k \perp AK_k,$$

此方法应用较广, 具体可以参考文献[1].

SGMRES 是 GMRES 的一种变形, 它生成基的过程与 GMRES 方法稍有不同, 但同样都是利用 Krylov 子空间  $K_k(A, v_1)$ , 具体算法可以参考文献[3].

SGMRES 满足

$$A[v_1, v_2, \dots, v_k] = [v_1, v_2, \dots, v_k] R_k \quad (2)$$

其中  $v_1, v_2, \dots, v_k$  是 Krylov 子空间  $K_k$  的一组标准正交基.

这种方法的优点在于最后要解的是  $R_k y_k = [\xi_1, \xi_2, \dots, \xi_k]^T$  这样一个三角方程组, 由于  $R_k$  是一个上三角矩阵. 这样避免了求解最小二乘问题, 计算量就小于 GMRES 方法. 缺点就是收敛性没有 GMRES 好.

WGMRES 是对 GMRES 方法的一种改进. GMRES 生成的是一组 Krylov 子空间的一组正交基, 而 WGMRES 生成的是一组 Krylov 子空间 D 正交基, 具体可以参考文献[4]. 实际上 WGMRES 可以看成是一个预处理过程, 由于 WGMRES 过程满足关系:

$$AL_k = L_k H_k + h_{k+1,k} l_{k+1} e_k^T \quad (3)$$

其中  $L_k^T D L_k = I$ ,  $L_k = [l_1, l_2, \dots, l_k]$ .

收稿日期: 2007-09-26

基金项目: 国家自然科学基金重点基金(10531080)资助

\* 通讯作者: lzlu@xmu.edu.cn

如果令  $Q_k = \sqrt{DL}_k$ , 那么式(3) 就改写成,

$$BQ_k = Q_k H_{k+1} + h_{k+1, k} q_{k+1} e_k^T \quad (4)$$

其中  $B = \sqrt{DA} \sqrt{D}^{-1}$ ,  $Q_k = [q_1, q_2, \dots, q_k]$ ,  $q_{k+1} = \sqrt{DL}_{k+1}$ ,  $Q_k^T Q_k = I$ .

WGMRES 方法可以看作对  $A$  进行预处理, 把  $A$  转化成  $B$ , 然后对  $B$  应用 GMRES 算法. 这种方法的优点在于收敛性优于 GMRES, 但是计算量比较大. 本文结合这两种方法的优点提出一种新的方法 WSGMRES, 这种方法结合了以上两种方法的优点, 在后面的数值实验中可以得到验证.

## 2 WSGMRES

先定义一下  $D$  内积, 如果  $d_i > 0 (1 \leq i \leq n)$ ,  $D = \text{diag}\{d_1, d_2, \dots, d_n\}$  是一个对角阵,  $u$  和  $v$  是两个向量,  $u, v \in R^n$ , 那么它们的  $D$  内积就是:

$$(u, v)_D = v^T D u = \sum_{i=1}^n d_i u_i v_i,$$

如果  $(u, v)_D = 0$  那么称  $u$  和  $v$  关于  $D$  相互正交, 也可以表示成  $u \perp_D v$ .

同样定义(加权) 范数:

$$\|u\|_D = \sqrt{(u, u)_D} = \sqrt{u^T D u} = \sqrt{\sum_{i=1}^n d_i u_i^2}, \quad \forall u \in R^n.$$

对于矩阵的情况, 定义:

$$\|K\|_D = \sqrt{\lambda_{\max}(K^T D K)} \quad (5)$$

其中  $\lambda_{\max}(K^T D K)$  表示  $K^T D K$  最大的特征值.

令  $d = (d_1, d_2, \dots, d_n)^T$ , 选取  $\|d\| = \sqrt{n}$ , 这样当  $d_1 = d_2 = \dots = d_n = 1$  时,

$$(u, v)_D = (u, v) \quad (6)$$

接下来介绍一下 Weighted Simpler 方法.

算法 1(Weighted Simpler 算法)

初始化: 选取  $r_0$ , 使得:  $\|r_0\|_D = 1$ .

迭代:

1) For  $k = 1 \dots m$ :

2) 计算:  $v_k = A v_{k-1}$ , 当  $k = 1$  时,  $v_1 = A r_0$ .

3) 当  $k > 1$  时, for  $i = 1 \dots k-1$ :

$$\textcircled{1} \rho_k = (v_i, v_k)_D.$$

$$\textcircled{2} v_k = v_k - \rho_k v_i.$$

4) endfor.

5)  $\rho_k = \|v_k\|_D$ , 如果  $\rho_k = 0$ , 停止. 否则, 令  $v_k =$

$v_k / \rho_k$ .

$$6) R_k = \begin{bmatrix} \rho_{11} & \dots & \rho_{1k} \\ \vdots & \ddots & \vdots \\ 0 & \dots & \rho_{kk} \end{bmatrix}.$$

7) endfor.

从 Weighted Simpler 算法可以推出, 令  $V_k = [r_0, v_1, \dots, v_{k-1}]$ ,  $W_k = [v_1, v_2, \dots, v_k]$ , 那么就有

$$A V_k = W_k R_k \quad (7)$$

其中

$$R_k = \begin{bmatrix} \rho_{11} & \dots & \rho_{1k} \\ \vdots & \ddots & \vdots \\ 0 & \dots & \rho_{kk} \end{bmatrix},$$

$W_k = [v_1, v_2, \dots, v_k]$  是  $D$  正交的一组基, 并且满足

$$\text{span}\{W_k\} = \text{span}\{A r_0, A^2 r_0, \dots, A^k r_0\},$$

$$\text{span}\{V_k\} = K_k(A, r_0) = \text{span}\{r_0, A r_0, \dots, A^{k-1} r_0\}.$$

求解方程就是寻找  $z_k \in K_k(A, r_0)$ ,  $x_k = x_0 + z_k$ , 使得:

$$r_k = (b - A x_k) \perp_D A K_k \quad (8)$$

其中  $\text{span}\{A K_k\} = \text{span}\{A V_k\} = \text{span}\{W_k\}$ .

令  $z_k = V_k y_k$ ,  $y_k = [\eta_1, \eta_2, \dots, \eta_k]^T$ , 那么:

$$A z_k = A [r_0, v_1, \dots, v_{k-1}] y_k \quad (9)$$

所以

$$r_0 = r_k + A V_k y_k = r_k + W_k \omega_k = r_k + W_k R_k y_k \quad (10)$$

令  $R_k y_k = \omega_k = [\xi_1, \xi_2, \dots, \xi_k]^T$ , 可以推出

$$r_0 = r_k + \sum_{i=1}^k \xi_i v_i \quad (11)$$

$$r_k = r_{k-1} - \xi_k v_k \quad (12)$$

根据式(8)、(12) 可得

$$\|r_k\|_D^2 + \|\xi_k v_k\|_D^2 = \|r_{k-1}\|_D^2,$$

所以

$$\begin{aligned} \|r_k\|_D &= \sqrt{\|r_{k-1}\|_D^2 - \xi_k^2} = \\ & \|r_{k-1}\|_D \sqrt{1 - \xi_k^2 / \|r_{k-1}\|_D^2} = \\ & \|r_{k-1}\|_D \sin(\arccos(\xi_k / \|r_{k-1}\|_D)). \end{aligned}$$

假设  $\|r_0\|_D = 1$ , 那么这个过程可以写成:

算法 2

1)  $\rho = \|r_0\|_D = 1$ .

2) for  $k = 1, \dots, m$ :

3)  $\rho = \rho \sin(\arccos(\xi_k / \rho))$ .

4) endfor.

第 3 步生成的  $\rho$  就是  $r_k$ .

接下来推导如何求出  $\xi_i (1 \leq i \leq m)$ .

由于  $r_k \perp_D W_k$ , 所以  $r_k \perp_D v_k$ , 根据式(12) 可得:

$$v_k^T D r_k = v_k^T D r_{k-1} - \xi_k v_k^T D v_k = 0, \text{ 所以}$$

$$\xi_k = v_k^T D r_{k-1} \quad (13)$$

那么就有以下的迭代式子:

$$\begin{cases} \xi_j = v_k^T D r_{k-1} \\ r_k = r_{k-1} - \xi_j v_k \\ R_k y_k = [\xi_1, \xi_2, \dots, \xi_j]^T, y_k = [\eta_1, \eta_2, \dots, \eta_k]^T \\ z_k = V_k y_k \\ x_k = x_0 + z_k \end{cases} \quad (u, v)^D = \sum_{i=1}^n d_i u_i v_i \quad (15)$$

数值实验表明, 利用式(15)进行计算会比式(14)进行计算速度上快约 90%, 而误差会减小约 40%<sup>[5]</sup>. 所以在数值实验中采用式(15)进行计算. 根据式(6), 当  $d = (1, 1, \dots, 1)^T$  时, 也就是  $D$  是单位阵时, 显然 WSGMRES 过程与 SGMRES 是一样的.

因为  $r_{k-1} = r_0 - \sum_{i=1}^{k-1} \xi_i v_i$  令  $\|r_0\|_D = \rho, r_0 = \rho/\rho$  得到

$$\begin{cases} \xi_j = v_k^T D r_{k-1} \\ r_k = r_{k-1} - \xi_j v_k \\ R_k y_k = [\xi_1, \xi_2, \dots, \xi_j]^T, y_k = [\eta_1, \eta_2, \dots, \eta_k]^T \\ \text{if } k = 1, z_1 = \eta_1 r_0 \\ \text{if } k > 1, z_k = \eta_k r_{k-1} + \sum_{i=1}^{k-1} (\eta_{k+1} + \eta_i \xi_i) v_i \\ x^k = x_0 + z^k \end{cases}$$

根据式(7)、(8)和(9)WSGMRES 最后就是要解这样一个方程

$$A V_k y_k = W_k R_k y_k = r_0 \quad (16)$$

现在考虑方程

$$W_k l = r_0 \quad (17)$$

在 GMRES 方法中, 式(17)是有精确解的,  $l = \|r_0\|_2 e_1$ , 但是这并不代表 GMRES 有精确解, 因为 GMRES 最后要求解一个最小二乘问题, 而不是式(17). 而在 WSGMRES 方法中式(17)不一定存在精确解, 下面分两种情况进行讨论:

1) 当  $v_1, v_2, \dots, v_k$  和  $r_0$  线性无关时, (17) 不可能存在精确解, 那么问题就转化成求解:

$$\begin{aligned} \min \|W_k l - r_0\|_D &\Leftrightarrow W_k^T D W_k l = \\ W_k^T D r_0 &\Leftrightarrow l = W_k^T D r_0, \end{aligned}$$

那么方程就转化成  $R_k y_k = W_k^T D r_0$ .

根据式(8)和(11), 可以得到  $R_k y_k = [\xi_1, \xi_2, \dots, \xi_j]^T, \xi_j = v_j^T D r_0$ , 这与上面的结果是一致的.

2) 当  $v_1, v_2, \dots, v_k$  和  $r_0$  线性相关时, 式(17) 显然一定存在精确解. 那么方程(1) 就存在精确解, 因此初值的选取是非常重要的.

考虑另一个方程:

$$A V_k y_k = r_0 \quad (18)$$

在 GMRES 方法中,  $V_k$  是一组正交基, 因此  $\kappa_2(V_k)$  是比较好的. 在 WSGMRE 方法中,  $V_k$  不是一组正交基, 所以现在要讨论一下  $V_k$  的条件数.

根据式(5) 定义  $\kappa_D(V_k) = \|V_k\|_D \|V_k^{-1}\|_D$  为  $V_k$  的条件数.

引理1 假设  $M \in R^{k \times k}$ , 并且  $M = [a, e_2, \dots, e_k]$ , 其中  $a = (\alpha_1, \alpha_2, \dots, \alpha_k)^T, \alpha_1 \neq 0$ , 那么  $\kappa_2(M) \leq (\|a\|_2^2 + 1) / |\alpha_1|$ .

证明

$$\kappa_2(M) = \|M\|_2 \|M^{-1}\|_2,$$

若  $u = (\mu_1, \mu_2, \dots, \mu_k)^T$  是一个单位向量, 令  $u' = (1 - \mu_1^2)^{-1/2} (0, \mu_2, \dots, \mu_k)^T$ , 那么就有

$$\begin{aligned} \|Mu\|_2 &= \|\mu_1 a + (1 - \mu_1^2)^{1/2} u'\|_2 \leq \\ &|\mu_1| \|a\|_2 + (1 - \mu_1^2)^{1/2} \|u'\|_2 \leq \end{aligned}$$

$$\sqrt{\|a\|_2^2 + \|u'\|_2^2} = \sqrt{\|a\|_2^2 + 1}.$$

接下来给出 WSGMRES 的算法.

算法3 (WSGMRES 算法) 初始化: 给定  $x, tol$ , 令  $r = b - Ax, \rho = \|r\|_D$ , 如果  $\rho < tol$ ,  $x$  就是要求的解.

否则, 令  $r = r/\rho, \rho = 1$ .

开始迭代:

1) For  $k = 1 \dots m$ :

2) 计算:  $v_k = A v_{k-1}$ , 当  $k = 1$  时,  $v_1 = Ar$ .

3) 当  $k > 1$  时, for  $i = 1 \dots k - 1$ :

$$\textcircled{1} \rho_k = (v_i, v_k)_D.$$

$$\textcircled{2} v_k = v_k - \rho_k v_i.$$

4) endfor.

5)  $\rho_k = \|v_k\|_D$ . 如果  $\rho_k \leq tol$ , 令  $k = k - 1$ , 跳到第 10 步. 否则, 令  $v_k = v_k/\rho_k$ .

$$6) R_k = \begin{bmatrix} \rho_{11} & \dots & \rho_{1k} \\ \vdots & \ddots & \vdots \\ 0 & \dots & \rho_{kk} \end{bmatrix}$$

7)  $\xi_j = v_j^T D r, \rho = \rho \sin(\arccos(\xi_j/\rho))$ . 如果  $\rho_0 \leq tol$ , 跳到第 10 步.

$$8) r = r - \xi_j v_k.$$

9) endfor.

10) 求解  $R_k y_k = [\xi_1, \xi_2, \dots, \xi_j]^T$ , 其中  $y_k = [\eta_1, \eta_2, \dots, \eta_k]^T$ .

$$11) z = \begin{cases} \text{if } k = 1, z_1 = \eta_1 r \\ \text{if } k > 1, z_k = \eta_k r + \sum_{i=1}^{k-1} (\eta_{k+1} + \eta_i \xi_i) v_i, \end{cases}$$

$$12) x = x + \rho z.$$

13)  $r = b - Ax$ , 如果  $\|r\|_2 \leq tol$ ,  $x$  就是要求的解; 否则, 令  $\rho = \|r\|_D, \rho = 1$ . 重新进行迭代.

在实际的计算中,

$$(u, v)^D = v^T D u \quad (14)$$

所以  $\|M\|_2 \leq \sqrt{\|a\|_2^2 + 1}$ . 同样,  $M^{-1} = [a', e_2, \dots, e_k]$ , 其中  $a' = (1/\alpha_1, -\alpha_2/\alpha_1, \dots, -\alpha_k/\alpha_1)^T$ . 同样可以得到

$$\|M^{-1}\|_2 \leq \sqrt{\|a'\|_2^2 + 1} = \sqrt{\|a\|_2^2 + 1} / |\alpha_1|.$$

所以

$$\kappa_2(M) = \|M\|_2 \|M^{-1}\|_2 \leq (\|a\|_2^2 + 1) / |\alpha_1|.$$

定理 1 对于  $V_k$ , 有  $\kappa_D(V_k) \leq (\|r_0\|_D + 1) / \|r_{k-1}\|_D$ , 其中  $V_k = [r_0 / \|r_0\|_D, v_1, \dots, v_{k-1}]$ .

证明 根据式(11), 令  $N_k = [r_{k-1} / \|r_{k-1}\|_D, v_1, \dots, v_{k-1}]$ , 可以得到

$$V_k = N_k M_k,$$

其中

$$M_k = \begin{bmatrix} \|r_{k-1}\|_D / \|r_0\|_D & 0 & \dots & 0 \\ \xi_j / \|r_0\|_D & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \vdots \\ \xi_{k-1} / \|r_0\|_D & 0 & \dots & 1 \end{bmatrix}.$$

因为

$$\|V_k\|_D = \sqrt{\lambda_{\max}(V_k^T D V_k)} = \sqrt{\lambda_{\max}(M_k^T N_k^T D N_k M_k)},$$

并且  $N_k^T D N_k = I$ , 所以

$$\|V_k\|_D = \sqrt{\lambda_{\max}(M_k^T M_k)} = \|M_k\|_2,$$

因此  $\kappa_D(V_k) = \kappa_2(M_k)$ .

根据引理 1 可以得到

$$\kappa_2(M_k) \leq (\|a\|_2^2 + 1) / |\alpha_1|,$$

$$a = (\|r_{k-1}\|_D / \|r_0\|_D, \xi_j / \|r_0\|_D, \dots, \xi_{k-1} / \|r_0\|_D)^T,$$

$$\alpha_1 = \|r_{k-1}\|_D / \|r_0\|_D.$$

令  $r_0 = \|r_0\|_D r'_0$ , 根据式(11) 可以得到:  $r_0 = N_k a$ . 可以推出

$$\|r'_0\|_D = \|a\|_2 = 1.$$

因此  $\kappa_2(M_k) \leq \|r_0\|_D (\|r'_0\|_D^2 + 1) / \|r_{k-1}\|_D$ . 故

$$\kappa_D(V_k) \leq \|r_0\|_D (\|r'_0\|_D^2 + 1) / \|r_{k-1}\|_D = 2 \|r_0\|_D / \|r_{k-1}\|_D.$$

$r_0$  表示初始残量,  $r_{k-1}$  表示第  $k-1$  步的残量, 而在实际的计算中,  $k$  一般比较小, 所以  $\|r_0\|_D / \|r_{k-1}\|_D$  一般不会很大, 所以  $\kappa_D(V_k)$  也不会很大, 对于式(18) 这个线性方程组的求解问题就是良态的.

### 3 数值实验

下面在 Pentium IV 1.50 GHz, 内存 256 MB 的计算机上使用 Matlab 6.5 进行数值实验, 来说明

WSGMRES 的收敛速度通常要比 GMRES 和 SGMRES 要快. 在以下的数值实验中都取  $m = 20$ , 初始向量  $x_0 = (0, 0, \dots, 0)^T$ , 在 WGMRES 中的  $D = \text{diag}\{d_1, d_2, \dots, d_n\}$ , 其中  $d_i = \sqrt{n} |r(i)| / \text{norm}(r)$ ,  $r(i)$  表示  $r$  向量的第  $i$  个元素. 在计算  $(a, b)_D$  的过程中是利用式(15) 进行计算, 这样不仅大大减小了计算量, 也大大减小了存储量, 不需要存储整个  $D$  矩阵, 只需要存储  $n$  维向量  $d = (d_1, d_2, \dots, d_n)$ . 在求解  $y_k$  的过程中, 并不是采用算法 3 中的方法, 而是利用式(16) 进行计算.

在图 1 中, 点划线(-·-) 表示 GMRES 方法, 点线(...) 表示 SGMRES 方法, 而 WSGMRES 方法用虚线(-) 表示. 横坐标表示迭代步数, 纵坐标表示残量 ( $\text{norm}(r)$ ) 的大小.

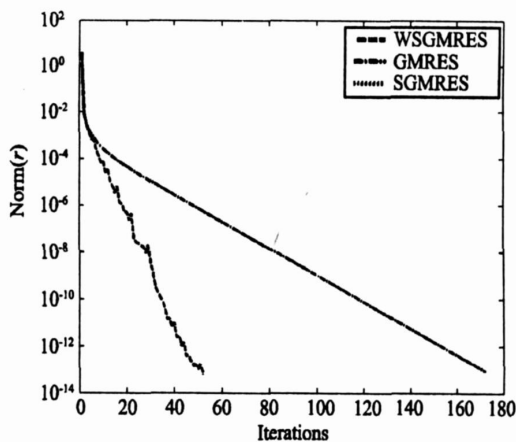


图 1 3 种算法关于 add20 的收敛比较

Fig. 1 Performance of GMRES, SGMRES and WSGMRES for add20

例子 1 选取的矩阵是 add20, 来自于矩阵市场, 在半导体系统设计技术中应用广泛. 这是一个 2 395 阶的方阵, 这个矩阵有 17 319 个非零元素, 它的条件数是  $1.76e+04$ . 把残量 ( $\text{norm}(r)$ ) 控制在  $10^{-13}$  附近.

从图 1 中可以看到 WSGMRES 的收敛速度非常快. 只需要 51 步残量就可以达到  $6.3631e-014$ , 而 SGMRES 和 GMRES 需要 171 步残量才能达到  $8.8449e-014$  和  $8.8340e-014$ , SGMRES 和 GMRES 这两条曲线基本上一致.

例子 2 第 2 个例子, 选取的是 bfw782a 矩阵, 也是来自矩阵市场, 应用于一般特征值问题. 这是一个 782 阶的方阵, 有 7 516 个非零元, 条件数是  $4.6e+03$ . 同样控制残量在  $10^{-13}$  附近.

从图 2 中可以看出 WSGMRES 的收敛速度非常快. 只需要 150 步残量就可以达到  $9.9296e-014$ , 而 SGMRES 和 GMRES 分别需要 476 和 510 步残量才能

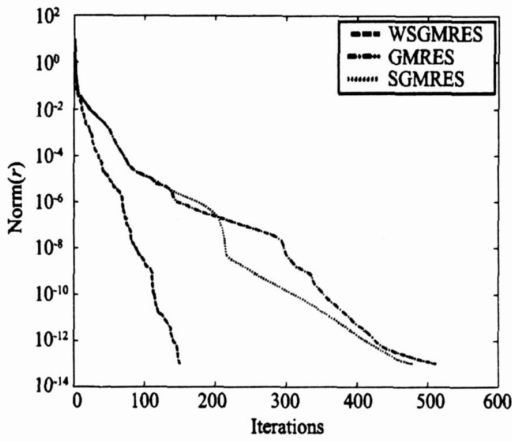


图2 3种算法关于 bfw\_782a 的收敛比较  
Fig.2 Performance of GMRES, SGMRES and WSGMRES for bfw\_782a

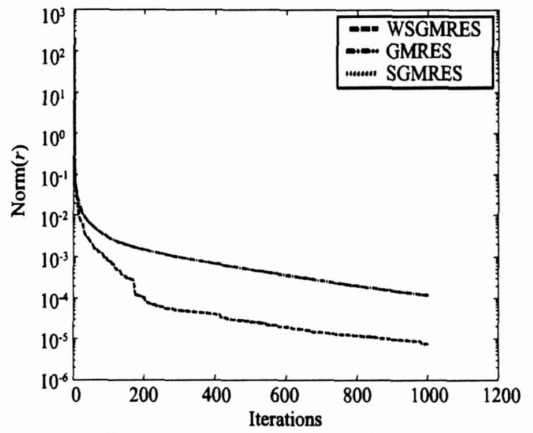


图3 3种算法关于 can\_445 的收敛比较  
Fig.3 Performance of GMRES, SGMRES and WSGMRES for can\_445

达到  $9.8818 \times 10^{-14}$  和  $9.8361 \times 10^{-14}$ .

例子3 选取的是 can\_445 矩阵, 同样来自于矩阵市场. 这是一个 445 阶的对称方阵, 有 3 809 个非零元, 条件数是  $3.7080 \times 10^{17}$ , 在飞行设计有限元结构问题中有广泛的应用. 控制迭代步数在 1 000 步以内.

从图3中可以看出 WSGMRES 的收敛速度比 GMRES 和 SGMRES 都要快, GMRES 和 SGMRES 这两条曲线基本重合, WSGMRES 在 1 000 步时残量可以达到  $7.3187 \times 10^{-6}$ , SGMRES 和 GMRES 只能达到  $1.1854 \times 10^{-4}$ .

#### 4 结论及进一步的工作

本文研究了求解大型线性稀疏系统的一种新方法, 从数值实验的结果来看, 这种方法的效果还是相当不错的. 但是本文没有对这种方法的收敛性做理论分析, 对  $D$  矩阵的取法也没有进一步的分析. 这些方面仍然在研究中.

#### 参考文献:

- [1] Saad Y, Schultz M H. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear system [J]. SIAM J Sci Statist Comput, 1986, 7: 856- 869.
- [2] Freund R, Nachtigal N M. QMR: A quasi- minimal residual method for non- Hermitian linear systems[J]. Numer Math, 1991, 60: 315- 339.
- [3] Walker H F, Zhou L. A Simpler GMRES[J]. Numer Lin Alg Appl, 1994, 1: 571- 581.
- [4] Essai A. Weighted FOM and GMRES for solving non-symmetric linear systems [J]. Numer Algorithm, 1998, 18: 277- 292.
- [5] Najafi H S, Ghazvini H. Weighted restarting method in the weighted Arnoldi algorithm for computing the eigenvalues of a nonsymmetric matrix[J]. Applied Mathematics and Computation, 2006, 175: 1276- 1287.
- [6] Boojhawon R, bhuruth M. Restarted Simpler GMRES augmented with harmonic Ritz vectors[J]. Future Generation Computer Systems, 2004, 20: 389- 397.

## A Weighted Simpler GMRES

YANG Sheng-wei, LU Lir-zhang\*

(School of Mathematical Sciences, Xiamen University, Xiamen 361005, China)

**Abstract:** GMRES (Generalized minimum residual method) is the most popular method for solving large scale nonsymmetric sparse linear systems. There exist a large variety of modifications to the standard GMRES algorithm, such as Simpler GMRES and Weighted GMRES. Simpler GMRES is less expensive than GMRES but Weighted GMRES converges faster. In this paper, we give a new method called Weighted Simpler GMRES, which is based on Simpler GMRES and combines it with Weighted GMRES. The Numerical experiments show that for some problems, Weighted Simpler GMRES converges faster than Simpler GMRES and GMRES, and it is less expensive than Weighted GMRES.

**Key words:** GMRES; Simpler GMRES; Weighted GMRES