

文章编号: 1001- 2486(2010) 02- 0061- 07

求解非对称线性方程组的 s -BiCR 算法*

仲妍, 骆志刚, 吴枫

(国防科技大学 计算机学院, 湖南 长沙 410073)

摘要: 在 BiCR 算法的基础上, 提出了求解非对称线性方程组的 s -BiCR 算法。首先, 给出了 s -BiCR 的基本计算框架, 介绍了算法基本原理及参数求解方法; 其次, 通过分析 s -BiCR 中剩余向量与方向向量序列的基本性质, 推导出减少参数求解计算量的方法, 并在此基础上提出了一种更为高效的 s -BiCR 算法; 最后, 证明了 s -BiCR 的正确性, 即在第 i 步产生的近似解与 BiCR 第 i 步产生的近似解是一致的, 同时, 通过性能分析发现, s -BiCR 的同步通信次数与访存次数明显少于 BiCR, 说明该算法具有很好的并行特性和数据本地性。大量实验验证了 s -BiCR 的高效性和正确性。

关键词: 非对称线性方程组; Krylov 子空间; BiCR; s -步方法; s -BiCR

中图分类号: TP301; O241 文献标识码: A

The s -BiCR Algorithm to Solve Nonsymmetric Linear Systems

ZHONG Yan, LUO Zhi-gang, WU Feng

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Based on the BiCR algorithm, s -BiCR algorithm is proposed to solve nonsymmetric linear systems. Firstly, the basic computation frame for the s -BiCR was given by introducing the fundamental principle in the new method and the solution approach to the parameters. Next, according to the analysis of the characters of the sequences of residual vectors and direction vectors, an approach is deduced to reduce computational volume for the parameters so that a more effective advanced s -BiCR is designed. Finally, the correctness of s -BiCR is proved, that is, the i th approximate solution from s -BiCR is equal to the i th approximate solution from BiCR. In addition, by performance analysis, the number of both synchronous communication and accessing memory for s -BiCR is less than the one for BiCR so the algorithm here has better parallel feature and data locality. The effectivity and validity of s -BiCR have been confirmed by experiments.

Key words: nonsymmetric linear systems; krylov subspace; BiCR; s -step methods; s -BiCR

设线性方程组 $Ax = b$ 的系数矩阵 A 为 n 阶非对称实数矩阵。现有求解法中, 在节省内存量和减少计算开销方面, 迭代法的优势突出, 其中, Krylov 子空间方法已成为研究的热点。设 v 为 n 维向量, 由 A 和 v 构成的 Krylov 子空间定义为 $K_m(A, v) = \text{span}\{v, Av, \dots, A^{m-1}v\}$ ($m \leq n$)^[1]。

Krylov 子空间方法分成两大类: 基于 Arnoldi 正交化过程和基于 Lanczos 双正交化过程的算法。与前者相比, 后一类方法具有形式简短的迭代公式, 存储和计算量较少。BiCG 是 Lanczos 类算法中的基本算法之一, 它的构造思想源于对称线性方程组求解算法——CG 算法^[1]; 为改进其性能, 发展了 Bi-CGSTAB(l)^[2] 等方法; 随后, 针对其不规则收敛性, 结合具有平滑收敛性的 TFQMR, 又提出了 QMRCGSTAB^[3]。另一方面, 基于另一类对称线性方程组求解算法——CR^[1] 算法, 文献[4]提出了一种最新的 BiCR 算法。

为增强 Krylov 子空间方法的并行特性, 使其更加适合于并行环境下的计算, 提出了一系列标准方法的 s 步方法, 如, s -BiCG^[5] 等。鉴于 BiCR 方法优于 BiCG 方法^[4], 本文旨在 BiCR 的基础上, 通过研究现有 s 步方法的特点, 提出了 s -BiCR 算法。

* 收稿日期: 2009- 11- 30

基金项目: 国家部委基金资助项目(9140C8103050601)

作者简介: 仲妍(1982-), 女, 博士生。

1 BiCR 算法

BiCR 算法由 Sogabe^[4] 提出,主要用于求解非对称线性方程组,它是 CR 算法的推广。如果系数矩阵是对称的,那么 BiCR 便退化为 CR 算法。下面,给出标准的 BiCR 算法如下:

算法 1 BiCR 算法

计算 $r_0 = b - Ax_0$; 选取 r_0^* (例如 $r_0^* = r_0$); 令 $p_{-1} = p_{-1} = \mathbf{0}, \beta_{-1} = 0$,

For $i = 0, 1, \dots$ 直至收敛

$$p_n = r_n + \beta_{n-1} p_{n-1}, \quad p_n^* = r_n^* + \beta_{n-1} p_{n-1}^*, \quad (Ap_n = Ar_n + \beta_{n-1} Ap_{n-1}), \quad \alpha_n = \frac{(r_n^*, Ar_n)}{(A^T p_n^*, Ap_n)}$$

$$x_{n+1} = x_n + \alpha_n p_n, \quad r_{n+1} = r_n - \alpha_n Ap_n, \quad r_{n+1}^* = r_n^* - \alpha_n A^T p_n^*, \quad \beta_n = \frac{(r_{n+1}^*, Ar_{n+1})}{(r_n^*, Ar_n)}$$

2 s -BiCR 算法

2.1 s -BiCR 算法基本框架

沿用 Krylov 子空间记号,令 $K_m(A, r_0) = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}, K_m(A^T, r_0^*) = \text{span}\{r_0^*, A^T r_0^*, \dots, (A^T)^{m-1}r_0^*\}$ 。 s -BiCR 方法的基本原理是:利用一次迭代过程完成 BiCR 方法的连续 s 次迭代,并确保第 i 步的近似解与 BiCR 第 is 步的近似解保持一致,这样同步次数相对 BiCR 而言可极大减少,从而加快了收敛速度。具体地,在第 i 步迭代过程中,抽取 Krylov 子空间 $K_{(i+2)s}(A, r_0)$ 中相互独立的 s 个剩余向量 $R_{i+1} = \{r_{i+1}, Ar_{i+1}, \dots, A^{s-1}r_{i+1}\}$ 构造方向向量 P_{i+1} ,抽取 $K_{(i+2)s}(A^T, r_0^*)$ 中相互独立的 s 个剩余向量 $R_{i+1}^* = \{r_{i+1}^*, A^T r_{i+1}^*, \dots, (A^T)^{s-1}r_{i+1}^*\}$ 构造方向向量 P_{i+1}^* ,并使得 $AP_{i+1} \perp A^T P_i^*, A^T P_{i+1}^* \perp AP_i$ 。下面给出 s -BiCR 算法的基本框架(见算法 2)。

算法 2 s -BiCR

令 $r_0 = b - Ax_0, R_0 = P_0 = [r_0, Ar_0, \dots, A^{s-1}r_0]$; 选取 $r_0^* = r_0$, 令 $R_0^* = P_0^* = [r_0^*, A^T r_0^*, \dots, (A^T)^{s-1}r_0^*]$ 。

For $i = 0, 1, \dots$ 直至收敛

(1) 选取 a_i , 使得 $r_{i+1} = r_i - AP_i a_i$ 正交于 $A^T P_i^*$; 选取 a_i^* , 使得 $r_{i+1}^* = r_i^* - A^T P_i^* a_i^*$ 正交于 AP_i ;

(2) $x_{i+1} = x_i + P_i a_i$; 令 $R_{i+1} = [r_{i+1}, Ar_{i+1}, \dots, A^{s-1}r_{i+1}]$; 令 $R_{i+1}^* = [r_{i+1}^*, A^T r_{i+1}^*, \dots, (A^T)^{s-1}r_{i+1}^*]$;

(3) 选取 B_i , 使得 $AP_{i+1} \perp A^T P_i^*$, 其中, $P_{i+1} = R_{i+1} + P_i B_i$;

选取 B_i^* , 使得 $A^T P_{i+1}^* \perp AP_i$, 其中, $P_{i+1}^* = R_{i+1}^* + P_i^* B_i^*$;

其中, $a_i = (a_i^1, a_i^2, \dots, a_i^s)^T, a_i^* = (a_i^{1,*}, a_i^{2,*}, \dots, a_i^{s,*})^T; B_i = [\beta_i^1, \beta_i^2, \dots, \beta_i^s], \beta_i^k = (\beta_i^{k,1}, \beta_i^{k,2}, \dots, \beta_i^{k,s})^T (1 \leq k \leq s), B_i^* = [\beta_i^{1,*}, \beta_i^{2,*}, \dots, \beta_i^{s,*}], \beta_i^{k,*} = (\beta_i^{k,1,*}, \beta_i^{k,2,*}, \dots, \beta_i^{k,s,*})^T (1 \leq k \leq s); P_i = [p_i^1, \dots, p_i^s], P_i^* = [p_i^{1,*}, \dots, p_i^{s,*}]$ 。

接下来,讨论算法 2 中参数 a_i, a_i^*, B_i, B_i^* 的求解。首先引入如下定义:

$$\text{定义 1 定义矩阵 } M_i = \begin{bmatrix} \mu_i^1 & \mu_i^2 & \dots & \mu_i^s \\ \mu_i^2 & \mu_i^3 & \dots & \mu_i^{s+1} \\ \dots & \dots & \dots & \dots \\ \mu_i^s & \mu_i^{s+1} & \dots & \mu_i^{2s-1} \end{bmatrix}_{s \times s}, \text{ 其中, } \mu_i^l = (r_i^*, A^{l+1}r_i), 1 \leq l \leq 2s-1.$$

命题 1 $M_i = (A^T R_i^*)^T A R_i$ 。

根据算法 2 中第(1)步 $r_{i+1} = (r_i - AP_i a_i) \perp A^T P_i^*$, 可得 $(A^T P_i^*)^T r_i = (A^T P_i^*)^T AP_i a_i$, 类似地, 有 $(AP_i)^T r_i^* = (AP_i)^T A^T P_i^* a_i^*$, 由此两式便可求出 a_i, a_i^* 。下面通过定义 2 给出其求解公式。

定义2 定义 $W_i = (A^T P_i^*)^T A P_i$, $m_i = (A^T P_i^*)^T r_i$, $m_i^* = (A P_i)^T r_i^*$, 则由 $W_i a_i = m_i$, $W_i^T a_i^* = m_i^*$ 求出 a_i, a_i^* 。

命题2 记 $m_i = (\mu_i^0, \dots, \mu_i^{i-1})^T$, 则 $m_i = m_i = m_i^*$ 成立。

根据算法2中第(3)步知 $(A^T P_i^*)^T A P_{i+1} = 0$, $P_{i+1} = R_{i+1} + P_i B_i$, 可得 $(A^T P_i^*)^T A P_i B_i = - (A^T P_i^*)^T A R_{i+1}$, 类似地有 $(A P_i)^T A^T P_i^* B_i = - (A P_i)^T A^T R_{i+1}^*$, 由此便可求出 B_i, B_i^* 。下面通过定义3给出其求解公式。

定义3 记 $C_i = [c_i^1, \dots, c_i^s]$, $C_i^* = [c_i^{1*}, \dots, c_i^{s*}]$, 其中, $c_i^k = ((A^k r_{i+1}, A^T p_i^{1*}), \dots, (A^k r_{i+1}, A^T p_i^{s*}))^T$, $c_i^{k*} = (((A^T)^k r_{i+1}^*, A p_i^1), \dots, ((A^T)^k r_{i+1}^*, A p_i^s))^T (1 \leq k \leq s)$, 则 $W_i B_i = -C_i$, $W_i^T B_i^* = -C_i^*$ 。

不难看出 W_i 对于 a_i, a_i^*, B_i, B_i^* 的求解具有重要作用, 为此通过命题3、4讨论其性质。

命题3 参数 a_i, a_i^*, B_i, B_i^* 具有唯一解的充要条件是 $|W_i| \neq 0$; 参数 a_i, a_i^* 具有唯一非零解的充要条件是 $|W_i| \neq 0$ 且 r_i 不与 $A^T R_i^*$ 共轭 (r_i^* 不与 $A R_i$ 共轭)。

命题4 $W_i = M_i + (B_{i-1}^*)^T C_{i-1}$ 。

2.2 s-BiCR 算法基本性质

首先, 类似于文献[5]得到如下引理:

引理1 s-BiCR 算法中, 若 $i = 0, 1, \dots, j-1$, 且 $|W_i| \neq 0, a_i^s \neq 0, a_i^{s*} \neq 0$, 则

- (1) $r_j \in K_{j+1}(A, r_0)$, 且若记 $r_j = q_r^j(A) r_0$, 则 $\text{degree}(q_r^j(\lambda)) = js$;
- (2) $p_j^k \in K_{j+s-k}(A, r_0)$, 且若记 $p_j^k = q_p^{j,k}(A) r_0$, 则 $\text{degree}(q_p^{j,k}(\lambda)) = js + k - 1$;
- (3) $r_j^* \in K_{j+1}(A^T, r_0^*)$, 且若记 $r_j^* = q_r^{j*}(A^T) r_0^*$, 则 $\text{degree}(q_r^{j*}(\lambda)) = js$;
- (4) $p_j^{k*} \in K_{j+s-k}(A^T, r_0^*)$, 且若记 $p_j^{k*} = q_p^{j,k*}(A^T) r_0^*$, 则 $\text{degree}(q_p^{j,k*}(\lambda)) = js + k - 1$ 。

下面, 讨论算法2中剩余向量序列 R_i, R_i^* 和方向向量序列 P_i, P_i^* 所满足的性质:

定理1 对于 $j = 1, 2, \dots$, 假设 $m, m^* \geq js$, 其中 m, m^* 分别为 r_0, r_0^* 相对于 A, A^T 的最小多项式的度; 对于 $i = 0, 1, \dots, j-1$, 假设有 $|W_i| \neq 0$ 且 $a_i^s \neq 0, a_i^{s*} \neq 0$, 那么 s-BiCR 满足以下性质:

- (1) 对于 $i < j$, 有 $A P_j \perp A^T P_i^*$; (2) 对于 $i < j-1$, 有 $A R_j \perp A^T R_i^*$; (3) 以 $A^T R_i^*$ ($i = 0, 1, \dots, j-1$) 的列向量张成的子空间, 即 $K_j(A^T, A^T r_0^*)$; 同样, $A^T P_i^*$ ($i = 0, 1, \dots, j-1$) 张成的子空间亦为 $K_j(A^T, A^T r_0^*)$; (4) $r_j \perp K_j(A^T, A^T r_0^*)$; (5) 对于 $i < j$, 有 $r_j \perp A^T P_i^*$; (6) 对于 $i < j$, 有 $r_j \perp A^T R_i^*$ 。

证明 使用归纳法证明。根据算法2中有关 B_i, a_i 的求解规则可知: $j = 1$ 时, (1)、(3)~(6) 成立; $j = 2$ 时, (1)~(6) 均成立。设 $j \leq k$ 时结论成立, 考虑 $j = k+1$ 时的情形:

先证(5)。当 $j = k+1$ 时, 即要证明“对 $i < k+1$ 有 $r_{k+1} \perp A^T P_i^*$ ”成立: 当 $i = k$ 时, 由算法2第(1)步知 $r_{k+1} \perp A^T P_k^*$ 成立, 因而只须证明 $i < k$ 时有 $r_{k+1} \perp A^T P_i^*$; 由(1)、(5)的归纳假设知, 当 $i < k$ 时有 $A P_k \perp A^T P_i^*, r_k \perp A^T P_i^*$ 成立, 故 $r_{k+1} = (r_k - A P_k a_k) \perp A^T P_i^*$ 成立。

证明(3)。欲证明 $A^T R_0^*, A^T R_1^*, \dots, A^T R_k^*$ 张成的子空间为 $K_{(k+1)s}(A^T, A^T r_0^*)$, 只须证明 R_0^*, \dots, R_k^* 的列向量均属于 $K_{(k+1)s}(A^T, A^T r_0^*)$ 且线性无关。结合引理1和题设条件, 类似于文献[5]的证明可知结论(3)成立。

证明(4)。根据已证得的(5)知 $r_{k+1} \perp A^T P_i^* (0 \leq i \leq k)$; 又由(3)知 $A^T P_0^*, \dots, A^T P_k^*$ 的列向量张成的子空间为 $K_{(k+1)s}(A^T, A^T r_0^*)$ 。因此 $r_{k+1} \perp K_{(k+1)s}(A^T, A^T r_0^*)$ 成立。

证明(6)。由(4)知 $r_{k+1} \perp K_{(k+1)s}(A^T, A^T r_0^*)$, 再由 $A^T R_0^*, A^T R_1^*, \dots, A^T R_k^*$ 的列向量张成的子空间为 $K_{(k+1)s}(A^T, A^T r_0^*)$ 可知, $r_{k+1} \perp A^T R_i^* (i = 0, \dots, k)$ 。

证明(2)。欲证明 $i < k$ 时有 $A R_{k+1} \perp A^T P_i^*$, 即要证明 $\{A r_{k+1}, \dots, A^s r_{k+1}\} \perp \{A^T r_i^*, \dots, (A^T)^s r_i^*\}$, 显然只要能够证明 $r_{k+1} \perp \{r_i^*, \dots, (A^T)^{2s} r_i^*\}$ 即可。由引理1知, $r_i^* = q_r^{i*}(A^T) r_0^*$ 且 $\text{degree}(q_r^{i*}(\lambda)) = is$, 因此, $\{r_i^*, \dots, (A^T)^{2s} r_i^*\} \subset K_{(i+2)s}(A^T, A^T r_0^*)$; 又由于当 $i < k$ 时有 $(i+2)s \leq (k+1)s$ 成立, 从而 $K_{(i+2)s}(A^T, A^T r_0^*) \subset K_{(k+1)s}(A^T, A^T r_0^*)$ 。由(4)知 $r_{k+1} \perp K_{(k+1)s}(A^T, A^T r_0^*)$, 从而有 $r_{k+1} \perp \{r_i^*, \dots,$

$(A^T)^{2s} r_i^*$ 。结论得证。

最后证明(1)。当 $i = k$ 时,由算法 2 中(3)可知 $AP_{k+1} \perp A^T P_k^*$ 成立。当 $i < k$ 时,由归纳假设知 $AP_k \perp A^T P_i^*$, 而 $AP_{k+1} = AR_{k+1} + AP_k B_k$, 所以 $AP_{k+1} \perp A^T P_i^*$ 成立的充要条件是 $AR_{k+1} \perp A^T P_i^*$ 。又由 $A^T P_i^* = A^T R_i^* + A^T R_{i-1}^* B_{i-1}^* + \dots + A^T R_0^* (B_{i-1}^* \dots B_0^*)$ 和(2)知 $i < k$ 时, AR_{k+1} 与 $A^T R_i^*, \dots, A^T R_0^*$ 共轭。因此, $AP_{k+1} \perp A^T P_i^*$ 成立。

类似于定理 1, 结合 A 和 A^T 关系, 可以对偶得到如下结论:

定理 2 题设条件同定理 1, 那么 s -BiCR 满足以下性质:

(1) 对于 $i < j$, 有 $A^T P_j^* \perp AP_i$; (2) 对于 $i < j - 1$, 有 $A^T R_j^* \perp AR_i$; (3) 以 $AR_i (i = 0, 1, \dots, j - 1)$ 的列向量张成的子空间, 即 $K_{js}(A, Ar_0)$; 同样, $AP_i (i = 0, 1, \dots, j - 1)$ 张成的子空间亦为 $K_{js}(A, Ar_0)$; (4) $r_j^* \perp K_{js}(A, Ar_0)$; (5) 对于 $i < j$, 有 $r_j^* \perp AP_i$ 且 $r_j^* \perp AR_i$ 。

2.3 s -BiCR 算法的改进

利用定理 1、2 的相关结论来简化算法 2 的计算, 并给出更为高效的 s -BiCR 算法:

命题 5 (1) c_i^1, \dots, c_i^s 有如下表达式:

$$c_i^1 = (0, \dots, 0, (Ar_{i+1}, (A^T)^s r_i^*))^T, \quad c_i^2 = (0, \dots, 0, (Ar_{i+1}, (A^T)^s r_i^*), (Ar_{i+1}, (A^T)^{s+1} r_i^*))^T, \dots, \\ c_i^s = ((Ar_{i+1}, (A^T)^s r_i^*), \dots, (Ar_{i+1}, (A^T)^{2s-1} r_i^*))^T.$$

(2) $c_i^{1*}, \dots, c_i^{s*}$ 有如下表达式:

$$c_i^{1*} = (0, \dots, 0, (A^T r_{i+1}^*, A^s r_i))^T, \quad c_i^{2*} = (0, \dots, 0, (A^T r_{i+1}^*, A^s r_i), (A^T r_{i+1}^*, A^{s+1} r_i))^T, \dots, \\ c_i^{s*} = ((A^T r_{i+1}^*, A^s r_i), \dots, (A^T r_{i+1}^*, A^{2s-1} r_i))^T.$$

命题 6 下述递推公式成立:

$$(1) (r_{i+1}, (A^T)^{s+1} r_i^*) = -\mu_{i+1}^0 / a_i^{s*}; (r_{i+1}, (A^T)^{s+k} r_i^*) = -\frac{1}{a_i^{s*}} [\mu_{i+1}^{k-1} + \sum_{l=s+1}^{s+k-1} a_i^{l-k*} (r_{i+1}, (A^T)^l r_i^*)] (k = 2, \dots, s);$$

$$(2) (r_{i+1}^*, A^{s+1} r_i) = -\mu_{i+1}^0 / a_i^s; (r_{i+1}^*, A^{s+k} r_i) = -\frac{1}{a_i^s} [\mu_{i+1}^{k-1} + \sum_{l=s+1}^{s+k-1} a_i^{l-k} (r_{i+1}^*, A^l r_i)] (k = 2, \dots, s)。$$

由命题 5、6 可以看出, c_i^1, \dots, c_i^s 和 $c_i^{1*}, \dots, c_i^{s*}$ 的求解可以归结为 $\mu_{i+1}^l = (r_{i+1}^*, A^{l+1} r_{i+1}) (0 \leq l \leq s-1)$ 的运算, 这样可以省却大量的计算开销。最后, 给出改进的 s -BiCR 算法。

算法 3 改进的 s -BiCR

令 $r_0 = b - Ax_0, R_0 = P_0 = [r_0, Ar_0, \dots, A^{s-1} r_0]$; 选取 $r_0^* ((r_0, r_0^*) \neq 0)$, 令 $R_0^* = P_0^* = [r_0^*, A^T r_0^*, \dots, (A^T)^{s-1} r_0^*]$;

计算 $\mu_0^0, \dots, \mu_0^{2s-1}$, 令 $m_0 = (\mu_0^0, \dots, \mu_0^{s-1})^T, W_0 = M_0$ 。

For $i = 0, 1, \dots$ 直至收敛

(1) 分解 $W_i = L_i U_i$, 求解 $L_i U_i a_i = m_i, (L_i U_i)^T a_i^* = m_i$;

(2) 计算 $x_{i+1} = x_i + P_i a_i; r_{i+1} = r_i - A P_i a_i; r_{i+1}^* = r_i^* - A^T P_i^* a_i^*$;

(3) 计算 $Ar_{i+1}, \dots, A^s r_{i+1}, A^T r_{i+1}^*, \dots, (A^T)^s r_{i+1}^*$, 令

$$R_{i+1} = [r_{i+1}, \dots, A^{s-1} r_{i+1}], AR_{i+1} = [Ar_{i+1}, \dots, A^s r_{i+1}], R_{i+1}^* = [r_{i+1}^*, \dots, (A^T)^{s-1} r_{i+1}^*], A^T R_{i+1}^* = [A^T r_{i+1}^*, \dots, (A^T)^s r_{i+1}^*];$$

(4) 计算 $\mu_{i+1}^0, \dots, \mu_{i+1}^{2s-1}$, 并令 $M_{i+1} = \{\mu_{i+1}^{k+l-1}\}_{k,l}, m_{i+1} = (\mu_{i+1}^0, \dots, \mu_{i+1}^{s-1})^T$, 其中 $\mu_l^k = (r_i^*, A^{l+1} r_i)$, $0 \leq l \leq 2s-1$;

(5) 计算 $C_i = [c_i^1, \dots, c_i^s], C_i^* = [c_i^{1*}, \dots, c_i^{s*}]$, 其中 $c_i^k = (0, \dots, 0, (r_{i+1}, (A^T)^{s+1} r_i^*), \dots, (r_{i+1}, (A^T)^{s+k} r_i^*))^T, c_i^{k*} = (0, \dots, 0, (r_{i+1}^*, A^{s+1} r_i), \dots, (r_{i+1}^*, A^{s+k} r_i))^T, k = 1, \dots, s$;

(6) 求解 $L_i U_i B_i = -C_i, (L_i U_i)^T B_i^* = -C_i^*$;

- (7) 计算 $P_{i+1} = R_{i+1} + P_i B_i$, $P_{i+1}^* = R_{i+1}^* + P_i^* B_i^*$; $AP_{i+1} = AR_{i+1} + AP_i B_i$, $AP_{i+1}^* = AR_{i+1}^* + AP_i^* B_i^*$;
 (8) 计算 $W_{i+1} = M_{i+1} + (B_i^*)^T C_i$;

3 s -BiCR 算法分析

3.1 正确性分析

命题 7 当 $s = 1$ 时, 算法 2 给出的 s -BiCR 算法即为标准的 BiCR 算法。

证明 当 $s = 1$ 时, a_i, a_i^* 为一维系数, $P_i, P_i^*, R_{i+1}, R_{i+1}^*, B_i, B_i^*$ 均为向量, 为便于与 BiCR 算法作比较, 将其分别改写为 $\alpha_i, \alpha_i^*, p_i, p_i^*, r_{i+1}, r_{i+1}^*, \beta_i, \beta_i^*$ 。由于 1-BiCR 与 BiCR 算法的基本流程相同, 唯一的区别在于参数 $\alpha, \alpha^*, \beta, \beta^*$ 的求解, 所以, 下面讨论两者的参数求解方法。

(1) 参数 α, α^* 的取值。由算法 2 第(1)步 $r_{i+1} \perp A^T p_i^*$ 得 $\alpha_i = (p_i^*, Ar_i) / (A^T p_i^*, Ap_i)$, 由于 $p_i^* = r_i^* + p_{i-1}^* \beta_{i-1}^*$ 且 $Ar_i \perp p_{i-1}^*$, 所以 $(p_i^*, Ar_i) = (r_i^*, Ar_i)$ 。不难看出算法 2 中 α_i 的求解与 BiCR 一样, α_i^* 类似可证。

(2) 参数 β_i, β_i^* 的取值。由算法 2 中第(3)步知 $Ap_{i+1} \perp A^T p_i^*$, 因而 $\beta_i = - (A^T p_i^*, Ar_{i+1}) / (A^T p_i^*, Ap_i)$, 由于 $r_{i+1}^* = r_i^* - A^T P_i^* a_i^*$, 所以 $(A^T p_i^*, Ar_{i+1}) = (r_i^* - r_{i+1}^*, Ar_{i+1}) / \alpha_i$, 进而有 $\beta_i = (r_{i+1}^* - r_i^*, Ar_{i+1}) / (r_i^*, Ar_i)$ 。再由定理 1 的(6)知 $r_{i+1} \perp A^T r_i^*$, 故 $\beta_i = (r_{i+1}^*, Ar_{i+1}) / (r_i^*, Ar_i)$ 。不难看出算法 2 中 β_i 的求解与 BiCR 算法中的相同, β_i^* 类似可证。

命题 8 在定理 1 的假设条件下, 若 BiCR 算法与 s -BiCR 算法初值相同, 则求解线性方程组 $Ax = b$ 时, s -BiCR 算法得到的近似解 x_i 与 BiCR 算法得到的 x_s 相同。

证明 (1) 分析 s -BiCR 与 BiCR 的迭代过程具有一致性。① s -BiCR 迭代目标: 由算法 2 的第(2)步可得 $x_i = x_0 + \sum_{k=0}^{i-1} P_k a_k$, 根据定理 2 的(3)可知 P_0, \dots, P_{i-1} 张成子空间 $K_{is}(A, r_0)$, 根据定理 1 的(4)可知 $r_i \perp K_{is}(A^T, A^T r_0^*)$, 因此算法 2 在第 i 步迭代过程中实质是寻找 $x_i \in x_0 + K_s(A, r_0)$, 使其残量 $r_i \perp K_{is}(A^T, A^T r_0^*)$; ② BiCR 迭代目标: 由算法 1 可得 $x_s = x_0 + \sum_{k=0}^{is-1} p_k \alpha_k$, 根据命题 7 知 1-BiCR 即为 BiCR, 为此令定理 2、4 中 $s = 1$, 那么根据定理 2 的(3)可知 p_0, \dots, p_{is-1} 张成子空间 $K_s(A, r_0)$ 。由定理 1 的(4)知 $r_{is} \perp K_{is}(A^T, A^T r_0^*)$, 易知 BiCR 在第 is 步与 s -BiCR 在第 i 步的迭代目标是一致的。

(2) 证明两个迭代过程得到的解相同。令 $V = [r_0, Ar_0, \dots, A^{s-1} r_0]$, $\Phi = (\varphi_1, \dots, \varphi_s)^T$, 则 BiCR 在第 is 步与 s -BiCR 在第 i 步得到的解可统一表述成 $\hat{x} = x_0 + V\Phi$, 相应地有 $\hat{r} = r_0 - AV\Phi$ 且 $\hat{r} \perp K_s(A^T, A^T r_0^*)$ 。令 $V^* = [A^T r_0^*, \dots, (A^T)^{is} r_0^*]$, 则有 $(V^*)^T AV\Phi = (V^*)^T r_0$; 因 V^* , V 满秩, A 非奇异, 故 $|(V^*)^T AV| \neq 0$, 因此 Φ 被唯一确定, 显然 \hat{x} 唯一。综上, s -BiCR 与 BiCR 得到的近似解 x_i 和 x_s 相同。

3.2 性能分析

表 1 给出标准 BiCG^[4]、标准 BiCR^[4]、 s -BiCG^[5] 与改进的 s -BiCR 之间计算量的比较。可看出, 1 步 s -BiCR (s -BiCG) 迭代较 s 步 BiCR (BiCG) 迭代增加了 $O(4s^2)$ 次向量更新操作, 而向量内积和矩阵向量乘操作次数则保持不变。表面看来, s -BiCR 的运算量增加了, 但实质是以此为代价减少了迭代过程中的同步次数和访存次数:

表 1 计算量对比
 Tab.1 Summary of cost

操作	BiCG	BiCR	s -BiCG	s -BiCR
向量内积	2s	2s	2s	2s
向量更新	5s	6s	$s(4s+3)$	$s(4s+3)$
矩阵向量乘	2s	2s	2s	2s

(1) 同步次数。令 $i = 0, 1, \dots$, 算法 s -BiCR 和 BiCR 如果需要得到相同近似解, 前者需要完成 $2is$ 个向量内积、 $2is$ 个矩阵向量乘和 $is(4s+3)$ 个向量更新操作, 通过数据相关性分析可知, 所需的同步通信次数分别为 i 、 is 和 $2i$ 次; 后者需要完成 $2is$ 个向量内积、 $2is$ 个矩阵向量乘和 $6is$ 个向量更新操作, 所需同步次数分别为 $2is$ 、 is 和 $2is$ 次; 两者同步次数总和之比为: $(s+3)/(5s)$ 。

(2) 访存次数。 s -BiCR 只需访存 1 次读入矩阵 A , 便可连续完成 $2s$ 个矩阵向量乘、 $2s$ 个向量内积, 而后者需要访存 s 次, 才能完成相应操作。

另外, 由于 BiCR 与 BiCG 的基本流程相同, 区别仅在于参数 α_i 、 β_i 的计算上, 因此两者的计算量大致相同, 所以决定了 s -BiCR 与 s -BiCG 的计算量也基本相同。

4 数值试验

首先, 对 BiCG^[4]、BiCR^[4]、 s -BiCG^[5] 和 s -BiCR 的性能进行对比测试, 以检验 s -BiCR 的高效性, 测试内容包括^[4]: 迭代次数 (the number of iterations, Its)、计算时间 (computational time, Time) 和迭代精度 $\lg(\|b - Ax_n\|_2 / \|b\|_2)$ (lg of the true relative residual 2-norm, TTR); 其次, 对 BiCR 和 s -BiCR 的收敛性进行对比分析, 以验证 s -BiCR 的正确性。所有试验程序均采用 Matlab 实现, 试验数据来源于 Harwell-Boeing collection^[6]、NEP collection^[7]、SPARSKIT collection^[8] 和 Tim Davis' collection^[9]。

表 2 BiCG \ BiCR \ s -BiCG \ s -BiCR 的性能对比测试

Tab. 2 Numerical results of BiCG \ BiCR \ s -BiCG \ s -BiCR

Matrix	gr_30_30	sayl3	e20r0000	fs_54l_1	themat	cage9	cage8	poli	add32	add20	
Time	BiCG	0.0780	1.7160	32.0430	0.0150	1.2000	1.1080	0.1060	1.9810	6.1780	6.2430
	BiCR	0.0780	1.4970	28.7040	0.0160	1.2230	1.0600	0.1060	1.9340	5.9900	6.3720
	s -BiCG	0.1090	1.9040	31.6680	0.0470	1.2100	1.1240	0.1460	1.9190	6.2560	6.6650
	s -BiCR	0.0940	1.7000	28.6410	0.0150	1.1690	1.0600	0.1240	1.8720	5.8030	5.3360
Its	BiCG	20	406	447	12	27	23	22	31	68	261
	BiCR	20	357	414	12	27	23	22	30	65	235
	s -BiCG	11	209	224	7	14	12	12	16	35	132
	s -BiCR	12	186	208	7	14	12	12	16	33	120
TTR	BiCG	-7.1114	-7.1139	-7.0385	-7.8701	-7.0648	-7.3738	-7.6003	-8.2376	-7.1782	-7.0758
	BiCR	-7.2154	-7.0025	-7.0418	-7.8452	-7.0632	-7.3178	-7.4531	-7.1321	-7.0434	-7.0002
	s -BiCG	-7.2250	-7.1240	-7.0334	-8.6968	-7.0648	-7.3738	-8.1923	-8.2377	-7.0275	-7.0431
	s -BiCR	-8.4723	-7.0114	-7.0605	-8.6300	-7.0632	-7.3178	-8.2233	-8.1966	-7.0434	-7.0016

首先比较 s 为 2 时四种算法的性能 (试验结果见表 2)。总体看来: (1) 计算时间方面, s -BiCR 最少, 根据 3.2 节的分析可知, 相对于 BiCG \ BiCR 而言, 尽管 s -BiCR 的计算量有所增加, 但其同步等待时间和访存时间却大幅减少, 当同步等待时间占据主导时, 总体计算时间便会显著降低; 另外, (s -) BiCR 的计算时间比 (s -) BiCG 少。(2) 迭代次数方面, (s -) BiCR 比 (s -) BiCG 少, 这是导致 (s -) BiCR 计算时间比 (s -) BiCG 少的原因, 说明前者的收敛速度相对较快; 另外, s 步方法的迭代次数近似为对应的非 s 步方法的一半, 这正好与其设计初衷相一致。(3) 计算精度方面, 四种算法均能达到迭代停止条件 $TTR < 1.0e-007$, 而且 (s -) BiCR 和 (s -) BiCG 的 TTR 均值相差不大; 尽管如此, 仍属 s -BiCR 的 TTR 均值最低, 且达到 $1.0e-008$ 的次数最多。随后, 针对 s 的更多取值进行了测试比较, 发现对于 $s \leq 5$ 时仍能取得较好的效果。综上所述, s -BiCR 的性能相对最优, 而且 (s -) BiCR 的性能优于 (s -) BiCG。

下面比较 BiCR 和 2 \ 3 \ 4 \ 5- BiCR 的收敛情况。试验结果如图 1 所示, (a)、(b) 和 (c) 中的矩阵分别为 jpw_h_991、sherman4 和 dwb512。可以看出, s -BiCR 的收敛速度基本保持在 BiCR 的 s 倍左右。同时, 所有的收敛曲线均在整体上呈现出相同的变化趋势: 图 1(a) 中, 大约在迭代过程的 1/2 和 5/6 处会出

现较为明显的突出, 而其它时候则相对平滑; 图 1(b) 中, 大约在迭代过程的 $3/14$ 、 $5/14$ 、 $7/14$ 和 $9/14$ 处会有明显突出, 其它处较为平滑; 图 1(c) 中, 三条曲线均很平滑, 没有明显突出。这些试验现象共同验证了 s -BiCR 的正确性, 即其第 i 步所完成的求解过程与 BiCR 的第 is 步过程是一致的。

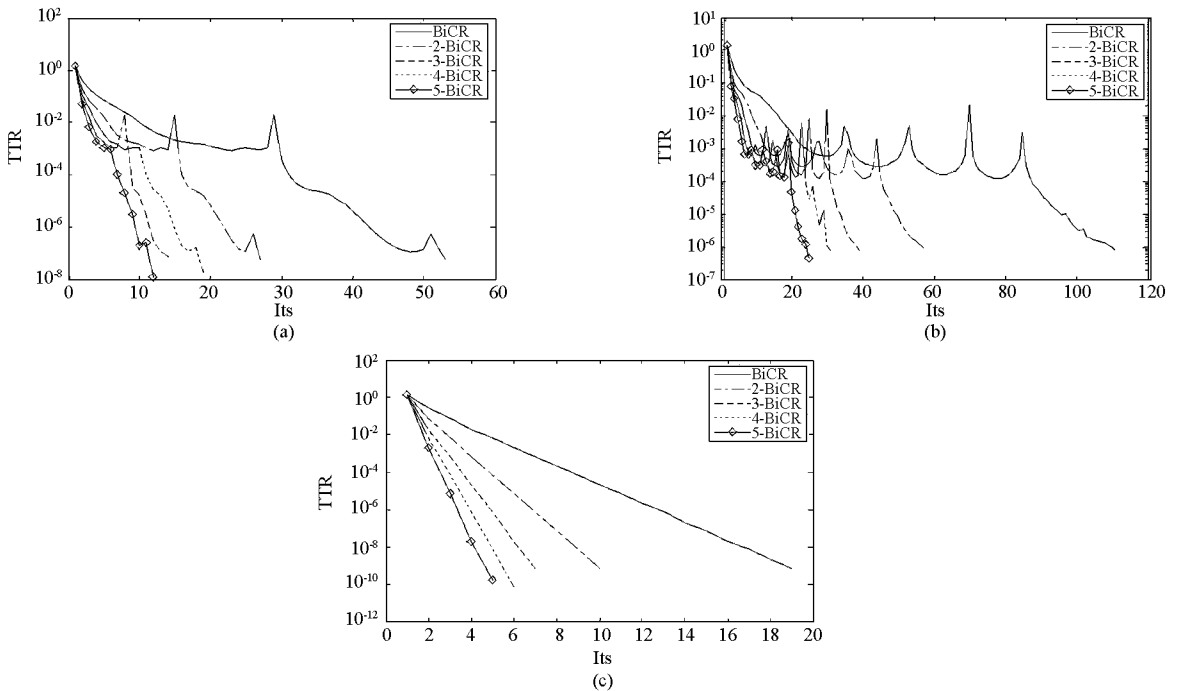


图 1 BiCR 和 s -BiCR 的收敛性对比

Fig. 1 Residual 2-norm histories of BiCR and s -BiCR

5 结束语

本文提出了求解非对称线性方程组的 s -BiCR 算法。首先, 给出了该算法的基本计算框架, 并在此基础上通过分析剩余向量与方向向量的基本性质, 推导出一种更为高效的 s -BiCR 算法; 其次, 讨论了 s -BiCR 算法的正确性, 并通过性能分析发现该算法具有很好的并行特性。实验结果表明, s -BiCR 算法的性能最优且 (s) -BiCR 算法的性能优于 (s) -BiCG 算法, 从而验证了本文算法的高效性; 随后, 通过对比分析 BiCR 算法和 s -BiCR 算法的收敛情况, 验证了 s -BiCR 算法的正确性。在下一步的研究中, 我们将会在分布式环境中实现 s -BiCR 算法。

参考文献:

- [1] 吴建平, 王正华, 李晓梅. 稀疏线性方程组的高效求解与并行计算[M]. 长沙: 湖南科学技术出版社, 2004.
- [2] Sleijpen G L G, Fokkema D R. BiCGStab(1) for Linear Equations Involving Unsymmetric Matrices with Complex Spectrum[J]. Electronic Transactions on Numerical Analysis, 1993, 1(1): 11- 32.
- [3] Chan T F, Gallopoulos E, Simoncini V, et al. A Quasi-minimal Residual Variant of the Bi-CGSTAB Algorithm for Nonsymmetric Systems[J]. SIAM Journal on Scientific Computing, 1994, 2(15): 338- 347.
- [4] Sogabe T, Sugihara M, Zhang S L. An Extension of the Conjugate Residual Method to Nonsymmetric Linear Systems[J]. Journal of Computational and Applied Mathematics, 2009, 1(226): 103- 113.
- [5] 骆志刚. 典型结构大型线性方程组的分布式并行算法研究[D]. 长沙: 国防科技大学, 2000.
- [6] Duff I S, Grimes R G, Lewis J G. User's Guide for the Harwell Boeing Sparse Matrix Collection[R]. Tech. Rep. RAL- 92- 086, Rutherford Appleton Laboratory, Chilton, UK, 1992.
- [7] Bai Z, Day D, Demmel J, et al. A Test Matrix Collection for Non-hermitian Eigenvalue Problems[R]. Tech. Rep. CS- 97- 355, Department of Computer Science, University of Tennessee, Knoxville, TN, March, 1997.
- [8] Saad Y. SPARSKIT: A Basic Tool Kit for Sparse Matrix Computation[R]. Tech. Rep. CSRDTR 1029, CSRD, University of Illinois, Urbana, IL, 1990.
- [9] Davis T. UF Sparse Matrix Collection[EB]. <http://www.cise.ufl.edu/research/sparse/matrices>, 2009.