
Proceedings of the Seventh International Conference on
Parallel, Distributed, GPU and Cloud Computing for Engineering
Edited by: P. Iványi, F. Magoulès and B.H.V. Topping
Civil-Comp Conferences, Volume 4, Paper 3.3
Civil-Comp Press, Edinburgh, United Kingdom, 2023
doi: 10.4203/ccc.4.3.3


# Avoiding Communication in Two-Sided Krylov Subspace Methods

**H. Liu[1,2] and F. Magoulès[3,4] and Q. Zou[1,2]**

[1]**School of Science, Beijing University of Posts and Telecommunications, Beijing, China**
[2]**Key Laboratory of Mathematics and Information Networks (Beijing University of Posts and Telecommunications), Ministry of Education, China**
[3]**Université Paris-Saclay, CentraleSupélec, MICS, Gif-sur-Yvette, France**
[4]**Faculty of Engineering and Information Technology, University of Pécs, Pécs, Hungary**

## Abstract

Krylov subspace methods play an important role in solving large, sparse linear systems in varieties of scientific fields. Specifically, their parallel variants are widely used in engineering. Classical Krylov subspace methods in parallel scenarios usually require many data communication between different processors per iteration which increases runtime of the algorithms and create a performance bottleneck. Besides, the cost of communication is much more expensive than the cost of computation on modern computer architecture. Therefore, we present two communication-avoiding Krylov subspace methods, namely communication-avoiding quasi-minimal residual algorithm (CA-QMR) and communication-avoiding transpose free quasi-minimal residual algorithm (CA-TFQMR). In practical applications, we reduce the communication demand to one data movement every $O$(s) iterations in the classical two-sided iterated algorithms. Moreover, we have incorporated restart strategy into our algorithms which significantly reduces storage requirements. Experimental results are presented to show

that our algorithms reduce communication consumption and the data storage within the allowable range of convergence accuracy.

**Keywords:** quasi-minimal residual algorithm, communication-avoiding strategy, iterative methods, parallel algorithms

# 1 Introduction

In the era of big data, the data amount increases sharply and the data structure becomes more and more complicated. Confront with difficulty in data storage and processing, parallel algorithms play a more important role in modern scientific and engineering applications. The runtime of a parallel algorithm is decided by both computation and communication. On the modern computer architectures, the cost of communication is much more expensive than the cost of floating point operation [4]. Therefore, many strategies have emerged to reduce the communication of parallel algorithms, such as communication-avoiding (CA) [10], communication-hiding (Pipelined) [8] and low-synchronization (LS) [9] algorithms. In what follows we mainly discuss the CA approaches.

In many situations, we need to calculate the solution of linear systems of the form:

$$Ax = b \tag{1}$$

where $A$ is an $n \times n$ nonsymmetric matrix and $b \in \mathbb{R}^n$. Since the matrix is usually large and sparse, Krylov subspaces methods (KSMs) is the most general and flexible choice. In the majority of cases, the generalized minimal residual algorithm (GMRES) [11] is the most successful solver. However, due to the work and storage growing, we have to use restarts, which leads to slow convergence. The biconjugate gradient algorithm (BCG) [12] and two modifications of BCG, namely CGS [6] and BiCGSTAB [13] have been widely used. To converge smoothly, the quasi-minimal residual algorithm (QMR) [2] was proposed. Later, a modification of QMR, namely TFQMR [7], have been proposed.

All the algorithms we mentioned above require at least one sparse matrix-vector multiplication (SpMV) and some vector operations in one iteration. These operations pose a huge demand for communication in the parallel algorithm. It limits the performance of the algorithm. Therefore, the communication-avoiding generalized minimal residual method (CA-GMRES) [10] was proposed. Then, Carson et al. [1] proposed communication-avoiding biconjugate gradient (CA-BICG) and communication-avoiding biconjugate gradient stabilized (CA-BICGSTAB) algorithms.

In this paper, we mainly focus on two-sided KSMs, which compute two Krylov

subspaces. Two-sided KSMs have better performance on sparse nonsymmetric linear systems [2]. Here, we focus on communication-avoiding quasi-minimal residual (CA-QMR) and tranpose-free QMR (CA-TFQMR). These variants are mathematically equivalent to their original versions. However, their numerical behavior can be much different.

# 2 Communication-avoiding QMR-like Algorithms

In this section we introduce the communication-avoiding variants of two-sided Krylov Subspace Methods. As mentioned above, CA-BICG and CA-BICGSTAB have be successfully formulated by Carson et al. [1]. Here we mainly focus on the QMR and CGS variants.

## 2.1 Communication-avoiding QMR Algorithm

Our core idea towards communication-avoiding QMR algorithm is to convert QMR (Algorithm 1) to an s-step method.

**Algorithm 1** QMR

1: Compute $r_0 = b - Ax_0$ and $\gamma_0 := \|r_0\|_2$, $w_0 := v_0 := r_0/\gamma_0$
2: **for** $m = 0, \ldots,$ until convergence **do**
3: Compute $\alpha_m, \delta_{m+1}$ and $v_{m+1}, w_{m+1}$ as in Lanczos Biorthogonalization
4: Update the QR factorization of $\overline{T}_m$
5: Apply rotation $\Omega_m$, to last column of $\overline{T}_m$ and to $\bar{g}_m$
6: $p_m = (v_m - \sum_{i=m-2}^{m-1} t_{im} p_i)/t_{mm}$
7: $x_m = x_{m-1} + \gamma_m p_m$
8: **end for**

Consider the biorthogonal bases for the Krylov subspaces

$$\mathcal{K}_s(A, v_0) := span\{v_0, Av_0, \ldots, A^{s-1}v_0\}$$
$$\mathcal{K}_s(A^T, w_0) := span\{w_0, A^T w_0, \ldots, (A^T)^{s-1}w_0\}$$

where $v_0, w_0$ are arbitrary vectors satisfying $(v_0, w_0) \neq 0$. Nothen, we rewrite the above Krylov subspaces into

$$\mathcal{K}_s(A, v_0) = span(V_s), \quad V_s := [\rho_0(A)v_0, \rho_1(A)v_0, \ldots, \rho_{s-1}(A)v_0] \tag{2}$$
$$\mathcal{K}_s(A^T, w_0) = span(W_s), \quad W_s := [\rho_0(A^T)w_0, \rho_1(A^T)w_0, \ldots, \rho_{s-1}(A^T)w_0] \tag{3}$$

where $\rho_j(z)$ is a polynomial of degree $j$, satisfying three-term recurrence

$$\rho_0(z) := 1, \rho_1(z) := (z - \alpha_0)\rho_0(z)/\gamma_0 \tag{4}$$
$$\rho_j(z) := ((z - \alpha_{j-1})\rho_{j-1}(z) - \beta_{j-2}\rho_{j-2}(z))/\gamma_{j-1} \tag{5}$$

Besides, there exist $(s+1) \times s$ matrices $T^1_{s+1}, T^2_{s+1}$:

$$T^1_{s+1} := \begin{pmatrix} \alpha^1_0 & \beta^1_0 & & \\ \gamma^1_0 & \alpha^1_1 & \ddots & \\ & \gamma^1_1 & \ddots & \beta^1_{s-2} \\ & & \ddots & \alpha^1_{s-1} \\ & & & \gamma^1_{s-1} \end{pmatrix}, \quad T^2_{s+1} := \begin{pmatrix} \alpha^2_0 & \beta^2_0 & & \\ \gamma^2_0 & \alpha^2_1 & \ddots & \\ & \gamma^2_1 & \ddots & \beta^2_{s-2} \\ & & \ddots & \alpha^2_{s-1} \\ & & & \gamma^2_{s-1} \end{pmatrix}$$

which satisfy:

$$AV_s = V_{s+1}T^1_{s+1}, A^T W_s = W_{s+1}T^2_{s+1} \tag{6}$$

Using the Krylov matrices (2)–(3), we represent components of the QMR iterates in the Krylov bases. We define $\widetilde{v}_j, \widetilde{w}_j$ to represent the vectors generated by QMR, and introduce vectors $\{a_j, c_j\}$ each of length $s$ to represent vectors $\{\widetilde{v}_j, \widetilde{w}_j\}$ such that

$$\widetilde{v}_j := V_s a_j, \quad \widetilde{w}_j := W_s c_j \tag{7}$$

where

$$a_0 := [1, 0_{1,s-1}]^T, c_0 := [1, 0_{1,s-1}]^T \tag{8}$$

Then, for $0 \leq j \leq s$, the update process (line 3) in Algorithm 1 becomes

$$\alpha_j := (A\widetilde{v}_j, \widetilde{w}_j) := (V_{s+1}T^1_{s+1}a_j, W_s c_j) \tag{9}$$
$$\widetilde{v}_{j+1} := A\widetilde{v}_j - \alpha_j\widetilde{v}_j - \beta_j\widetilde{v}_{j-1} := V_{s+1}T^1_{s+1}a_j - \alpha_j V_s a_j - \beta_j V_s a_{j-1} \tag{10}$$
$$\widetilde{w}_{j+1} := A^T\widetilde{w}_j - \alpha_j\widetilde{w}_j - \beta_j\widetilde{w}_{j-1} := W_{s+1}T^2_{s+1}c_j - \alpha_j W_s c_j - \beta_j W_s c_{j-1} \tag{11}$$
$$\beta_{j+1} := (\widetilde{v}_{j+1}, \widetilde{w}_{j+1}) \tag{12}$$

Combining with (6), we can see that (8)–(10) can be rewritten as

$$\alpha_j := (W_s^T V_{s+1}T^1_{s+1}a_j, c_j) \tag{13}$$
$$[a^T_{j+1}, 0] := T^1_{s+1}a_j - \alpha_j[a^T_j, 0] - \beta_j[a^T_{j-1}, 0] \tag{14}$$
$$[c^T_{j+1}, 0] := T^2_{s+1}c_j - \alpha_j[c^T_j, 0] - \beta_j[c^T_{j-1}, 0] \tag{15}$$
$$\beta_{j+1} := (W^T_{s+1}V_{s+1}[a^T_{j+1}, 0], [c^T_{j+1}, 0]) \tag{16}$$

We compute the dot products in a new iteration, using the Gram-like matrix

$$G := W^T_{s+1}V_{s+1} \tag{17}$$

where $G$ is an $(s+1) \times s+1$ matrix. Moreover, due to the the properties of $V_{s+1}$ and $W_{s+1}$, the product $W_s^T V_{s+1}$ used in (12) is equivalent to $G_{1:s,1:s+1}$. Here, $G_{1:s,1:s+1}$ represents the first $s$ rows and first $s+1$ columns of the matrix $G$.

Now, we can assemble the CA-QMR components from (3)–(5) and (12)–(16) into Algorithm 2.

**Algorithm 2** CA-QMR

Compute $r_0 = b - Ax_0$ and $\gamma_0 := \|r_0\|_2$, $w_0 := v_0 := r_0/\gamma_0$
**for** $m = 0, s, 2s, \ldots$ until convergence **do**
  Compute $V_{s+1}, W_{s+1}$ according to (3)-(4)
  Compute $T^1_{s+1}, T^2_{s+1}$ according to (5)
  Compute $G$ accoding to (16)
  Initialize $a_0, c_0$ according to (7)
  **for** $j = 0, \ldots, s-1$ **do**
    $\alpha_j := (G_{1:s,1:s+1} T^1_{s+1} a_j, c_j)$
    $[a^T_{j+1}, 0] := T^1_{s+1} a_j - \alpha_j [a^T_j, 0] - \beta_j [a^T_{j-1}, 0]$
    $[c^T_{j+1}, 0] := T^2_{s+1} c_j - \alpha_j [c^T_j, 0] - \beta_j [c^T_{j-1}, 0]$
    $\beta_{j+1} := (G[a^T_{j+1}, 0], [c^T_{j+1}, 0])$
    $a_{j+1} = a_{j+1}/\beta_{j+1}$
    $c_{j+1} = c_{j+1}/\beta_{j+1}$
    $$H_{j+1} := \begin{pmatrix} \alpha_0 & \beta_1 & & \\ \beta_1 & \alpha_1 & \ddots & \\ & \beta_2 & \ddots & \beta_j \\ & & \ddots & \alpha_j \\ & & & \beta_{j+1} \end{pmatrix}$$
    Apply $\Omega_i, i = j-2, j-1$ to $j$-th column of $H_{j+1}$
    Compute $c'_j = \frac{h_{j,j}}{\sqrt{h^2_{j,j} + h^2_{j+1,j}}}$, $s'_j = \frac{h_{j+1,j}}{\sqrt{h^2_{j,j} + h^2_{j+1,j}}}$
    $\gamma_{j+1} := -s'_j \gamma_j$
    $\gamma_j := c'_j \gamma_k$
    $\alpha_j := c'_j \alpha_j + \beta_{j+1} s'_j$
    $p_j = (V_s a_j - \sum_{i=j-2}^{j-1} h_{i,j} p_i)/h_{j,j}$
    $x_j = x_{j-1} + \gamma_j p_j$
  **end for**
  Recover iterates $x_{m+s}, v_{m+s}, w_{m+s}$
**end for**

## 2.2 Communication-avoiding CGS Algorithm

Based on the strategy above, we can also convert the CGS algorithm [6] to a communication-avoiding version. It is well known that CGS is the basis for many Krylov subspace methods. Here we can see that using the above techniques we can give a similar formulation for avoiding communications.

**Algorithm 3** CGS

Compute $r_0 := b - Ax_0$;$r_0^*$ arbitrary
$p_0 := u_0 := r_0$
**for** $j = 0, 1, 2, \ldots,$ until convergence **do**
    $\alpha_j = (r_j, r_0^*)/(Ap_j, r_0^*)$
    $q_j = u_j - \alpha_j A p_j$
    $x_{j+1} = x_j + \alpha_j(u_j + q_j)$
    $r_{j+1} = r_j - \alpha_j A(u_j + q_j)$
    $\beta_j = (r_{j+1}, r_0^*)/(r_j, r_0^*)$
    $u_{j+1} = r_{j+1} + \beta_j q_j$
    $p_{j+1} = u_{j+1} + \beta_j(q_j + \beta_j p_j)$
**end for**

Consider the CGS algorithm, shown in Algorithm 3. We can represent the components of the CGS iterates by the Krylov bases. Define

$$\mathcal{K}_{2s+1}(A, p_0) = span(P_{2s+1}), \quad \mathcal{K}_{2s}(A, r_0) = span(R_{2s}) \tag{18}$$

Let $T_{j+1}$ be the form of (6). We have

$$AP_{2s} = P_{2s+1}T_{2s+1}, AR_{2s-1} = R_{2s}T_{2s} \tag{19}$$

Using the Krylov matrices, we represent components of the CGS iterates in the Krylov bases. We introduce vectors $\{a_j, c_j, e_j, m_j, n_j\}$ each of length $4s + 1$ to represent vectors $\{p_j, r_j, x_j, q_j, u_j\}$

$$p_j := [P_{2s+1}, R_{2s}]a_j, r_j := [P_{2s+1}, R_{2s}]c_j \tag{20}$$
$$q_j := [P_{2s+1}, R_{2s}]m_j, u_j := [P_{2s+1}, R_{2s}]n_j \tag{21}$$
$$x_j := [P_{2s+1}, R_{2s}]e_j \tag{22}$$

where

$$a_0 := [1, 0_{1,4s}]^T, c_0 := [0_{1,2s+1}, 1, 0_{1,2s-1}]^T \tag{23}$$
$$n_0 := [0_{1,2s+1}, 1, 0_{1,2s-1}]^T, e_0 := [0_{1,4s+1}]^T \tag{24}$$

On the other hand, we have

$$A[p_j, r_j] = [P_{2s+1}, R_{2s}]T'[a_j, c_j] \tag{25}$$

where

$$T' = \begin{bmatrix} [T_{2s+1}0_{2s+1,1}] & \\ & [T_{2s}0_{2s,1}] \end{bmatrix} \tag{26}$$

Moreover, we use again the Gram-like matrix

$$g := [P_{2s+1}, R_{2s}]^T r_0^* \tag{27}$$

Similarly to the CA-QMR algorithm, we can assemble the CA-CGS components into Algorithm 4.

**Algorithm 4** CA-CGS

Compute $r_0 = b - Ax_0$;$r_0^*$ arbitrary
**for** $m = 0, s, 2s, \ldots$ until convergence **do**
    Compute $P_{2s+1}, R_{2s}$ according to (18)
    Compute $T'$ according to (10) and (26)
    Compute $G$ and $g$ accoding to (27)
    Initialize $a_0, c_0, m_0, e_0$ according to (24)
    **for** $j = 0, \ldots, s-1$ **do**
        $\alpha_j := \beta_j/(T'a_j, g)$
        $m_j := n_j - \alpha_j T' a_j$
        $e_{j+1} := e_j + \alpha_j(m_j + n_j)$
        $c_{j+1} := c_j - \alpha_j T'(m_j + n_j)$
        $\beta_j := (c_{j+1}, g)/(c_j, g)$
        $n_{j+1} = c_{j+1} + \beta_j m_j$
        $a_{j+1} = c_{j+1}/\beta_j(m_j + \beta_j a_j)$
    **end for**
    Recover iterates $p_{m+s}, r_{m+s}, x_{m+s}$
**end for**

We can see that CA-QMR and CA-CGS are mathematically equivalent to their original counterparts. Using similar techniques, we can further develop communication-avoiding transpose free quasi-minimal residual (CA-TFQMR) method and other generalizations. Note that theoretically these algorithms can be somewhat less stable than the traditional version. In practice we expect that the gain in communication cost can be greater than their stability drawback.

# 3 Numerical Experiments

In this section, we use 4-CAQMR to represent CA-QMR (Algorithm 2) where the step size $s$ is chosen as 4. All the initial guess is a zero vector and the matrices are extracted from the Matrix Market repository (https://math.nist.gov/MatrixMarket/).

Figure 1 illustrates the convergence behavior of CA-QMR with different step sizes. We find that the CA-QMR results are competitive with the classical algorithm. The accuracy of CA-QMR with different step sizes is close to that of QMR when the condition number is not large. From this result, we can almost sure that in parallel environment the new algorithm can be much more efficient than the traditional version.

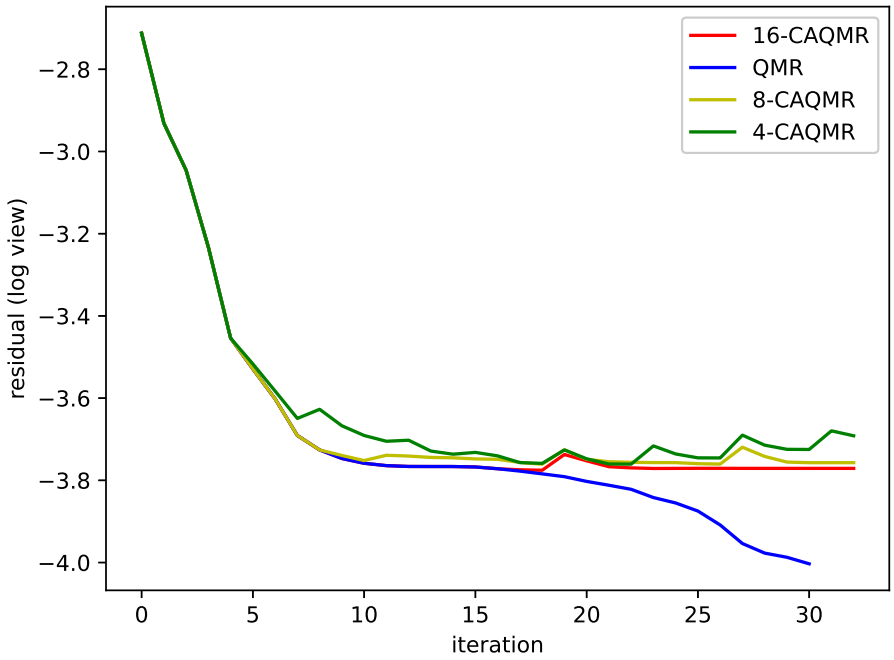


Figure 1: The size of the test matrix if $216 \times 216$ with $4374$ nonzero entries. The estimated condition number is $6.5 \times 10^4$.

| Condition Number | QMR | 4-CAQMR | 8-CAQMR | 16-CAQMR |
|---|---|---|---|---|
| 1.7e+04 | 2.3989e-12 | 4.3632e-11 | 2.3758e-11 | 1.4994e-11 |
| 1.4e+07 | 2.9339e-09 | 1.2507e-08 | 5.7247e-09 | 4.0081e-09 |
| 1.2e+08 | 2.5459e-08 | 2.5357e-07 | 1.0359e-07 | 5.8758e-08 |

Table 1: Algorithm convergence accuracy under different condition numbers.

Table 1 illustrates more results about QMR and their communication-avoiding variants, from which we can find that the step size of CA-QMR does affect the convergence accuracy.

# 4 Future work and conclusions

In this paper, we investigate communication-avoiding two-sided Krylov subspace methods. In particular, CA-QMR and CA-CGS have been discussed in details. Numerical experiments confirm the effectiveness our communication-avoiding variants. We can see from both theoretical and experimental points that CA-QMR is promising, which

has similar stability behavior to QMR while reducing the communication cost to one data movement per $O(s)$ iterations. Practically we have to carefully design the parallel implementation of the communication-avoiding algorithms, including the SpMV operations, data movement, and some low-level optimizations. In theory, however, such strategy does improve numerical performance of Krylov methods. Future research may focus on communication-avoiding transpose-free QMR algorithm and the restart strategy. Other acceleration strategies, such as mixed precision and randomization, as well as efficient parallel implementation are also interesting, and thus could be embedded into this work as future plans.

## Acknowledgements

This work was partly funded by National Natural Science Foundation of China under grant numbers 12101071, 12171051 and 12171052, and partly funded by the French National Research Agency as part of project ADOM, under grant number ANR-18-CE46-0008.

# References

[1] E. Carson, N. Knight, J. Demmel, Avoiding communication in nonsymmetric lanczos-based Krylov subspace methods, SIAM Journal on Scientific Computing 35 (5) (2013) S42–S61.

[2] R. W. Freund, N. M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, Numerische Mathematik 60 (1) (1991) 315–339.

[3] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd Edition, SIAM, Philadelphia, PA, 2003.

[4] E. C. Carson, Z. Strakoš, On the cost of iterative computations, Philos. Trans. R. Soc. A 378 (2166) (2020) 20190050.

[5] Y. Saad, Numerical methods for large eigenvalue problems: revised edition, SIAM, 2011.

[6] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, SIAM journal on scientific and statistical computing 10 (1) (1989) 36–52.

[7] R. W. Freund, A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems, SIAM journal on scientific computing 14 (2) (1993) 470–482.

[8] P. Ghysels, T. J. Ashby, K. Meerbergen, W. Vanroose, Hiding global communication latency in the GMRES algorithm on massively parallel machines, SIAM journal on scientific computing 35 (1) (2013) C48–C71.

[9] K. Świrydowicz, J. Langou, S. Ananthan, U. Yang, S. Thomas, Low synchronization Gram–Schmidt and generalized minimal residual algorithms, Numerical Linear Algebra with Applications 28 (2) (2021) e2343.

[10] M. Hoemmen, Communication-avoiding Krylov subspace methods, University of California, Berkeley, 2010.

[11] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on scientific and statistical computing 7 (3) (1986) 856–869.
[12] C. Lanczos, Solution of systems of linear equations by minimized iterations, J. Res. Nat. Bur. Standards 49 (1) (1952) 33–53.
[13] H. A. Van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM Journal on scientific and Statistical Computing 13 (2) (1992) 631–644.