# New implementation of QMR-type algorithms

M.D. García, E. Flórez, A. Suárez, L. González, G. Montero *

*University Institute of Intelligent Systems and Numerical Applications in Engineering, University of Las Palmas de Gran Canaria, Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, Spain.*

**Abstract**

Quasi-minimal residual algorithms, these are QMR, TFQMR and QMRCGSTAB, are biorthogonalisation methods for solving nonsymmetric linear systems of equations which improve the irregular behaviour of BiCG, CGS and BiCGSTAB algorithms, respectively. They are based on the quasi-minimisation of the residual using the standard Givens rotations that lead to iterations with short term recurrences. In this paper, these quasi-minimisation problems are solved using a different direct solver which provides new versions of QMR-type methods, the modified QMR methods (MQMR). MQMR algorithms have different convergence behaviour in finite arithmetic although are equivalent to the standard ones in exact arithmetic. The new implementations may reduce the number of iterations in some cases.

In addition, we study the effect of reordering and preconditioning with Jacobi, ILU, SSOR or sparse approximate inverse preconditioners on the performance of these algorithms.

Some numerical experiments are solved in order to compare the results obtained by standard and modified algorithms.

*Key words:* Nonsymmetric linear systems, sparse matrices, Krylov subspace methods, quasi-minimal residual methods, preconditioning, reordering.

## 1 Introduction

The application of discretization techniques to obtain approximate solutions of partial differential equations generally leads to large and sparse linear systems of equations,

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

* Corresponding author.

*Preprint submitted to Elsevier Science* *21 March 2005*

Direct solvers have the disadvantage of producing the fill-in effect which affects the memory requirements and the computational cost. However, iterative methods based on Krylov subspaces present some advantages with respect to direct ones and other iterative solver.

For systems with symmetric positive definite matrix, the Conjugate Gradient algorithm [20] is in general the best choice. Nevertheless, for nonsymmetric systems, there exist different families of methods [23], each of them with its own characteristics of robustness and efficiency. Orthogonalisation methods such as GMRES [25] are constructed using a minimisation procedure in a Krylov subspace generated by $A$, what produces a smooth monotonic convergence but at the expense of increasing cost and memory requirements per iteration. The Biconjugate Gradient method (BiCG) [9], reference of all biorthogonalisation methods, does not increase the computational cost and memory requirements along the iterations. The procedure is defined by a Galerkin condition instead of a minimisation as GMRES. This leads to an erratic convergence behaviour with strong oscillation of the residual norm. In addition, this algorithm includes a matrix-vector product with $A^T$ per iteration and there exists a double possibility of break-down. Sonneveld [26] proposes a transpose free algorithm, the conjugate gradient squared (CGS), a faster converging alternative to BiCG when the latter converges, but with the same convergence problems. In order to improve and smooth the convergence of the previous biorthogonalisation methods, Van der Vorst [28] presents the BiCGSTAB which has a better performance in most of the cases but does not eliminate the break-downs.

Freund and Nachtigal [12] propose another biorthogonalisation approach, the quasi-minimal residual method (QMR), which solves the rest of the BiCG problems, although it is not transpose free. Each iteration is characterised by a quasi minimisation of the residual norm, leading to a smoother convergence without strong oscillations. The break-down in BiCG due to nonexistent iterates is avoided. On the other hand, this method uses a look-ahead variant of the nonsymmetric Lanczos algorithm [13,14] for generating the basis of the Krylov subspace, which eliminates the other case of possible break-down of BiCG. However, in some applications $A$ is only accessible by approximations and not explicitly. In such cases, $A^T$ is not readily available. The Transpose-Free QMR algorithm (TFQMR) [11] is a quasi-minimal residual version of the CGS algorithm that smoothes its convergence without involving $A^T$-vector products. Following the same procedure, Chan et al [5] propose a QMR variant of the BiCGSTAB algorithm (QMRCGSTAB), which simultaneously takes advantage of the quasi-minimisation of the residual and the transpose free characteristic of BiCGSTAB. Nevertheless, the differences between TFQMR and CGS is more appreciable than those between QMRCGSTAB and BiCGSTAB due to the smoother behaviour of the latter compared to CGS. The relation between both families of algorithms is well illustrated in [29], where the quasi-minimal residual methods are derived by using residual smoothing techniques in BiCG, CGS and BiCGSTAB algorithms, respectively.

The behaviour of these methods improves considerably when preconditioning is used [1,24,4,27]. These techniques consist of transforming the original system (1) into another $\overline{A}\overline{x} = \overline{b}$, which provides the same solution, where $\overline{A}$ has a lower condition number. Implicit preconditioners construct approximations of matrix $A$ that are easily reversible or suitable to factorise, for example, Jacobi, SSOR and ILU. More recently, the possibilities of parallel computing have led to explicit preconditioners that directly approximate the inverse of $A$. In [19,22] it is obtained such approximate inverse $M$ by minimising the Frobenius norm of matrix $AM - I$. Also a factorized approximate inverse is proposed in [2].

The effect of reordering techniques on the convergence of preconditioned Krylov methods has been studied by several authors. In [7,**?**] it is observed that reordering has not a beneficial effect in the convergence behaviour of the Conjugate Gradient method with incomplete factorisation preconditioning. However, these techniques considerably improve the convergence of other Krylov subspace methods for solving nonsymmetric linear systems [8,3,10].

In section 2 we summarise the formulation of the standard QMR algorithm and introduce its modified version. Next, in sections 3 and 4, respectively, the modified TFQMR and QMRCGSTAB methods are developed. Section 5 is devoted to some numerical experiments in order to compare the proposed algorithms with other Krylov subspace methods, including the standard QMR-type algorithms. Finally, in section 6 we present the concluding remarks of this paper.

## 2 Modified QMR method

The approximate solution using the standard QMR method for the Krylov subspace of order $k$ is,

$$x_k = x_0 + V_k\, u \tag{2}$$

where $u$ minimises the norm,

$$\left\| \gamma e_1 - \overline{T}_k\, u \right\|_2 \tag{3}$$

which is a simplification of the residual norm,

$$\|\, r\|_2 = \left\| V_{k+1} \left( \gamma e_1 - \overline{T}_k\, u \right) \right\|_2 \tag{4}$$

where $V_k$ is the matrix which columns are the vectors $v_i$, $i = 1, ..., k$, obtained by Lanczos biorthogonalisation procedure, $\gamma = \|r_0\|_2$, and matrix $\overline{T}_k$ is,

$$\overline{T}_k = \begin{pmatrix} T_k \\ \delta_{k+1} e_k^t \end{pmatrix} \tag{5}$$

with,

$$T_k = \begin{pmatrix} \alpha_1 & \beta_2 & & . & & & \\ \delta_2 & \alpha_2 & \beta_3 & . & & & \\ & \delta_3 & \alpha_3 & . & & & \\ . & . & . & . & . & . & . \\ & & & . & \alpha_{k-2} & \beta_{k-1} & \\ & & & . & \delta_{k-1} & \alpha_{k-1} & \beta_k \\ & & & . & & \delta_k & \alpha_k \end{pmatrix} \tag{6}$$

$\alpha_i$, $i = 1, ..., k$; $\beta_j$, $j = 2, ..., k$; $\delta_l$, $l = 2, ..., k+1$, are the parameters obtained during Lanczos process (see, e.g., [24]).

In this paper, the quasi-minimisation problem is solved using a similar procedure to that developed in [15] for the minimisation problem arising in GMRES. We will directly solve the minimum square problem related to the quadratic functional (3), instead of using the QR factorisation of matrix $\overline{T}_k$; see e.g. [16]. Consider the orthogonal projection on the subspace of solutions of the quasi-minimisation problem (3) multiplying by $\overline{T}_k^T$ we obtain,

$$\overline{T}_k^T \, \overline{T}_k \, u = \overline{T}_k^T \gamma e_1 \tag{7}$$

where the structure of the $(k+1) \times k$ matrix $\overline{T}_k$ is,

$$\overline{T}_k = \begin{pmatrix} \boxed{\qquad d_k^T \qquad} \\ \boxed{\begin{matrix} & U_k \\ & \\ \mathbf{0} & \end{matrix}} \end{pmatrix}$$

the first row of $\overline{T}_k$ is a $k$ dimension vector $d_k^t$, and the rest is an upper triangular matrix $U_k$,

$$d_k = \begin{pmatrix} \alpha_1 & \beta_2 & 0 & . & . & . & 0 \end{pmatrix}$$

$$U_k = \begin{pmatrix} \delta_2 & \alpha_2 & \beta_3 & . & & & \\ & \delta_3 & \alpha_3 & \beta_4 & & & \\ & & & . & & & \\ . & . & . & . & . & . & . \\ & & & . & \delta_{k-1} & \alpha_{k-1} & \beta_k \\ & (0) & & . & & \delta_k & \alpha_k \\ & & & . & & & \delta_{k+1} \end{pmatrix}$$

where,

$$\{d_k\}_i = d_i = \left\{\overline{T}\right\}_{1i} \qquad i = 1, ..., k \tag{8}$$

$$\{U_k\}_{ij} = u_{ij} = \begin{cases} \left\{\overline{T}\right\}_{i+1,\,j} & 1 \le i \le j \le k \\ 0 & \text{in the rest} \end{cases} \tag{9}$$

then, the decomposition of the product $\overline{T}_k^T\,\overline{T}_k$ in (7) becomes in a sum,

$$\left\{\overline{T}_k^T\,\overline{T}_k\right\}_{ij} = d_i d_j + \sum_{m=1}^{k} u_{mi} u_{mj} \tag{10}$$

Taking into account the decomposition of $\overline{T}_k^T\,\overline{T}_k$, the equation (7), can be written as,

$$\left(d_k d_k^T + U_k^T U_k\right) u = \overline{T}_k^T\,\gamma e_1 \tag{11}$$

and, from $\overline{T}_k^T e_1 = d_k$, we obtain,

$$\left(d_k d_k^T + U_k^T U_k\right) u {=} \gamma d_k \tag{12}$$

Using the associative and distributive properties of matrix product, the equation above can be written as,

$$U_k^T U_k u {=} d_k \left(\gamma - \langle d_k, u\rangle\right) \tag{13}$$

from,

$$\lambda_i = \gamma - \langle d_k, u\rangle \tag{14}$$
$$u = \lambda_i p_k \tag{15}$$

we obtain,

$$U_k^T U_k p_k {=} d_k \tag{16}$$

Which is a double triangular system, where $U_k^T$ y $U_k$ are triangular matrices and only two substitution process are required for the solution.

Once we solve (16), we compute $\lambda_i$ to obtain $u$ from equation (15),

$$\lambda_i = \gamma - \langle d_k, u\rangle = \gamma - \lambda_i \langle d_k, p_k\rangle \tag{17}$$

thus,

$$\lambda_i = \frac{\gamma}{1 + \langle d_k,\, p_k\rangle} \tag{18}$$

Note that $1 + \langle d_k, p_k\rangle \neq 0$, because,

$$\langle d_k, p_k\rangle = \left\langle U_k^T U_k p_k, p_k \right\rangle = \|U_k p_k\|_2^2 \geq 0 \tag{19}$$

therefore $\lambda_i$ never degenerates.

The proposed method requires:

1. Given $d_k$ and $U_k$ defined in (8) and (9), solve in a double triangular system given in (16) ,

$$U_k^T \bar{p}_k = d_k \tag{20}$$
$$U_k p_k = \bar{p}_k \tag{21}$$

2. Compute $\lambda_i$ in equation (18).

3. Obtain $u$ solving equation (15)

The residual vector whose norm is given in (4) can be obtained from,

$$r_i = V_{k+1}\widehat{r}_i \tag{22}$$

where $\widehat{r}_i$ is the $(k+1)$-vector,

$$\widehat{r}_i = \gamma e_1 - \overline{T}_k\, u \tag{23}$$

and its entries can be computed as follow,

$$\{\widehat{r}_i\}_j = \begin{cases} \lambda_i & \text{if} \quad j = 1 \\ -\lambda_i \bar{p}_k & \text{if} \quad j = 2, ..., k+1 \end{cases} \tag{24}$$

Since, from partition of $\overline{T}_k$, the first entry from $(k+1)$-vector $(\overline{T}_k\, u)$ is $\langle d_k, u\rangle$, and the rest of the entries are given by $k$-vector $(U_k u)$. Then the first entry of $\hat{r}_i$ is $\lambda_i$, and the rest are,

$$-U_k u = -\lambda_i U_k p_k = -\lambda_i \bar{p}_k \tag{25}$$

where $\bar{p}_k$ can be kept in the resolution of the first triangular system given in (20).

Note that the residuals are not equivalent (as in GMRES), because vectors $v_i$ are not orthonormal, $\|r_i\|_2 \neq \|\widehat{r}_i\|_2$. The MQMR algorithm obtained with direct solving of the quasi-minimisation problem results as follows,

*MQMR algorithm*

Initial guess $x_0$. $r_0 = b - Ax_0$

$\beta_1 = \delta_1 = 0$

$v_0 = w_0 = 0$

$\gamma = \|r_0\|$

$v_1 = w_1 = \dfrac{1}{\gamma} r_0$

Do while $\sqrt{k+1} \parallel \widehat{r}_{k-1} \parallel / \parallel r_0 \parallel \geq \varepsilon \quad (k = 1, 2, 3, ...)$,

$\alpha_k = \langle Av_k, w_k \rangle$

$\widehat{v}_{k+1} = Av_k - \alpha_k v_k - \beta_k v_{k-1}$

$\widehat{w}_{k+1} = A^T w_k - \alpha_k w_k - \delta_k w_{k-1}$

$\delta_{k+1} = |\langle \widehat{v}_{k+1}, \widehat{w}_{k+1} \rangle|^{1/2}$

$\beta_{k+1} = \langle \widehat{v}_{k+1}, \widehat{w}_{k+1} \rangle / \delta_{k+1}$

$v_{k+1} = \widehat{v}_{k+1} / \delta_{k+1}$

$w_{k+1} = \widehat{w}_{k+1} / \beta_{k+1}$

Solve $U_k^T \bar{p} = d_k$ and $U_k p = \bar{p}$

$$\text{where} \begin{cases} \{d_k\}_m = \left\{\overline{T}\right\}_{1m} \\ \{U_k\}_{lm} = \left\{\overline{T}\right\}_{l+1m} \end{cases} \quad l, m = 1, ..., k$$

$\lambda_k = \dfrac{\gamma}{1 + \langle d_k, \, p \rangle}$

$u_k = \lambda_k \, p$

$x_k = x_0 + V_k u_k$ ; being $V_k = [v_1, v_2, ..., v_k]$

$r_k = V_{k+1} \widehat{r}_k$ ; being $V_{k+1} = [v_1, v_2, ..., v_{k+1}]$,

$$\begin{cases} \{\widehat{r}_k\}_1 = \lambda_k \\ \{\widehat{r}_k\}_{l+1} = -\lambda_k \{\bar{p}\}_l \end{cases} \quad l = 1, ..., k$$

End

We must take into account that the convergence criterion depends on $\widehat{r}_k$, which is the residual computed from Modified QMR.

## 3 Modified TFQMR Method

The approximation obtained using TFQMR method in a Krylov subspace of dimension $k$, is,

$$x_0 + Y_k u_k \tag{26}$$

where $Y_k = [y_1, y_2, ..., y_k]$, $y_k = t_{i-1}$ if $k = 2i - 1$ is odd, and $y_k = q_i$ if $k = 2i$ is even, and $u_k$ minimises the norm $\left\| \left( \delta_1 e_1 - \bar{T}_k\, u \right) \right\|_2$, which represents a quasi-minimum of the residual la norm (see Saad [24]),

$$\| r_k \|_2 = \left\| W_{k+1} \Delta_{k+1}^{-1} \left( \delta_1 e_1 - \Delta_{k+1} \overline{B}_k u_k \right) \right\|_2 \tag{27}$$

being,

$$\overline{\bar{T}}_k = \mathbf{\Delta}_{k+1} \overline{B}_k \tag{28}$$

Where $W_{k+1}$ is the matrix whose columns are the vectors,

$$W_{k+1} = [w_1, w_2, ..., w_{k+1}] \tag{29}$$

and $\mathbf{\Delta}_{k+1}$ is a diagonal matrix, such that $W_{k+1}$ is scaled up ($\delta_k = \| r_i \|$, if $k = 2i+1$ is odd, or $\delta_k = \sqrt{\| r_{i-1} \| \ \ \| r_i \|}$, if $k = 2i$ is even),

$$\mathbf{\Delta}_{k+1} = \begin{pmatrix} \delta_1 & & . & & \\ & \delta_2 & . & & \\ . & . & . \ . & & . \\ & & . \ \delta_k & & \\ & & . & & \delta_{k+1} \end{pmatrix} \tag{30}$$

and $\overline{B}_k$ is the $(k+1)\times k$ matrix,

$$\overline{B}_k = \begin{pmatrix} \alpha_0^{-1} & & & . & & \\ -\alpha_0^{-1} & \alpha_0^{-1} & & . & & \\ & -\alpha_0^{-1} & \alpha_1^{-1} & . & & \\ . & . & . & . \; . & & . \\ & & & . \; \alpha_{(k-1)/2}^{-1} & & \\ & & & . \; -\alpha_{(k-1)/2}^{-1} & \alpha_{(k-1)/2}^{-1} & \\ & & & . & -\alpha_{(k-1)/2}^{-1} & \end{pmatrix} \tag{31}$$

The MTFQMR algorithm obtained with direct solving of the quasi-minimisation problem is as follows.

*MTFQMR algorithm*

Initial guess $x_0$. $r_0 = b - Ax_0$

$r_0^*$ is arbitrary, such that $\langle r_0, r_0^* \rangle \neq 0$

$s_0 = t_0 = r_0$

$v_0 = As_0$

$\rho_0 = \langle r_0, r_0^* \rangle$

$\delta_1 = \| r_0 \|$

Do while $\sqrt{i+1} \parallel \widehat{r}_{i-1} \parallel / \parallel r_0 \parallel \geq \varepsilon \quad (i = 1, 2, 3, ...)$

$\sigma_{i-1} = \langle v_{i-1}, r_0^* \rangle$

$\alpha_{i-1} = \rho_{i-1} / \sigma_{i-1}$

$q_i = t_{i-1} - \alpha_{i-1} v_{i-1}$

$r_i = r_{i-1} - \alpha_{i-1} A \left( t_{i-1} + q_i \right)$

From $k = 2i-1, 2i$ do

If $k$ is odd do

$\delta_{k+1} = \sqrt{\| r_{i-1} \| \; \| r_i \|}$; $y_k = t_{i-1}$

Else

$$\delta_{k+1} = \| \, r_i \|; \, y_k = q_i$$

End

End

Solve $U_k^T \bar{p} = d_k$ and $U_k p = \bar{p}$

$$\text{where} \begin{cases} \{d_k\}_m = \left\{ \overline{\overline{T}} \right\}_{1m} \\ \{U_k\}_{lm} = \left\{ \overline{\overline{T}} \right\}_{l+1m} \end{cases} \quad l, m = 1, ..., k$$

$$\lambda_k = \frac{\delta_1}{1 + \langle d_k, \, p \rangle}$$

$$u_k = \lambda_k \, p$$

$x_k = x_0 + Y_k u_k$ ; with $Y_k = [y_1, y_2, ..., y_k]$

$$\begin{cases} \{\widehat{r}_i\}_1 = \lambda_{2i} \\ \{\widehat{r}_i\}_{l+1} = -\lambda_{2i} \, \{\bar{p}\}_l \end{cases} \quad l = 1, ..., 2i$$

$$\rho_i = \langle r_i, r_0^* \rangle$$

$$\beta_i = \rho_i / \rho_{i-1}$$

$$t_i = r_i + \beta_i q_i$$

$$s_i = t_i + \beta_i (q_i + \beta_i s_{i-1})$$

$$v_i = \; A s_i$$

End

Now the convergence criterion depends on $\widehat{r}_k$, which represents the residual, computed from Modified TFQMR.

## 4 Modified QMRCGSTAB Method

The QMRCGSTAB algorithm proposed by Chan et al [5], makes two quasi-minimisations per iterations. If we define $Y_k = [y_1, y_2, ..., y_k]$, being $y_{2l-1} = g_l$ for $l = 1, ..., [(k+1)/2]$ ($[(k+1)/2]$ the integer part of $(k+1)/2$) and $y_{2l} = s_l$ for $l = 1, ..., [k/2]$($[k/2]$ the integer part of $k/2$). The approximate solution of the system $Ax = b$, starting from the $k$-th Krylov subspace, is built as $x_0 + Y_k u_k$, where $u_k$ minimises the norm $\left\| \left( \delta_1 e_1 - \bar{T}_k\, u \right) \right\|_2$, which is again a quasi-minimum of the residual norm,

$$\| r_k \|_2 = \left\| W_{k+1} \Delta_{k+1}^{-1} \left( \delta_1 e_1 - \Delta_{k+1} \overline{B}_k\, u_k \right) \right\|_2 \tag{32}$$

being,

$$\overline{\overline{T}}_k = \mathbf{\Delta}_{k+1} \overline{B}_k \tag{33}$$

$W_{k+1}$ is the matrix whose columns are the residual vectors,

$$W_{k+1} = [w_1, w_2, ..., w_{k+1}] \tag{34}$$

with $w_{2l-1} = s_l$ for $l = 1, ..., [(k+1)/2]$ and $w_{2l} = r_l$ for $l = 1, ..., [k/2]$; and $\mathbf{\Delta}_{k+1}$ is a diagonal matrix, such that $W_{k+1}$ is scaled up ($\delta_i = \| w_i \|$),

$$\mathbf{\Delta}_{k+1} = \begin{pmatrix} \delta_1 & & . & & \\ & \delta_2 & . & & \\ . & . & . & . & . \\ & & . & \delta_k & \\ & & . & & \delta_{k+1} \end{pmatrix} \tag{35}$$

$\overline{B}_k$ is the $(k+1) \times k$ matrix,

$$\overline{B}_k = \begin{pmatrix} \sigma_1^{-1} & & & . & & \\ -\sigma_1^{-1} & \sigma_2^{-1} & & . & & \\ & -\sigma_2^{-1} & \sigma_3^{-1} & . & & \\ . & . & . & . & . & . \\ & & & . & \sigma_{k-1}^{-1} & \\ & & & . & -\sigma_{k-1}^{-1} & \sigma_k^{-1} \\ & & & . & & -\sigma_k^{-1} \end{pmatrix} \tag{36}$$

with $\sigma_{2l} = \omega_l$ for $l = 1, ..., [(k+1)/2]$, and $\sigma_{2l-1} = \alpha_l$ for $l = 1, ..., [(k+1)/2]$.

The MQMRCGSTAB algorithm obtained with direct solving of the quasi-minimisation problem is written below.

*MQMRCGSTAB algorithm*

Initial guess $x_0$, $r_0 = b - Ax_0$

$r_0^*$ is arbitrary, such that $\langle r_0, r_0^* \rangle \neq 0$

$\rho_0 = \alpha_0 = \omega_0 = 1$

$g_0 = v_0 = 0$

Do while $\sqrt{2i+1} \parallel \widehat{r}_{i-1} \parallel / \parallel r_0 \parallel \geq \varepsilon \quad (i = 1, 2, 3, ...)$

$\rho_i = \langle r_0^*, r_{i-1} \rangle$

$\beta_i = (\rho_i / \rho_{i-1})(\alpha_{i-1} / \omega_{i-1})$

$g_i = r_{i-1} + \beta_i (g_{i-1} - \omega_{i-1} v_{i-1})$

$v_i = A g_i$

$\alpha_i = \dfrac{\rho_i}{\langle v_i, r_0^* \rangle}$

$s_i = r_{i-1} - \alpha_i v_i$

$\delta_{2i-1} = \|s_i\|$ , $y_{2i-1} = g_i$

$t_i = A s_i$

$\omega_i = \dfrac{\langle t_i, s_i \rangle}{\langle t_i, t_i \rangle}$

$r_i = s_i - \omega_i t_i$

$\delta_{2i} = \|r_i\|$ , $y_{2i} = s_i$

Solve $U_{2i}^t \bar{p} = d_{2i}$ and $U_{2i} p = \bar{p}$

$$\text{where} \begin{cases} \{d_{2i}\}_m = \left\{\overline{T}\right\}_{1m} \\ \{U_{2i}\}_{lm} = \left\{\overline{T}\right\}_{l+1m} \end{cases} \quad l, m = 1, ..., 2i$$

$\lambda_{2i} = \dfrac{\delta_1}{1 + \langle d_{2i},\, p \rangle}$

$u_{2i} = \lambda_{2i}\, p$

$x_i = x_0 + Y_{2i} u_{2i}$; with $Y_{2i} = [y_1, y_2, ..., y_{2i}]$

$$\begin{cases} \{\widehat{r}_i\}_1 = \lambda_{2i} \\ \{\widehat{r}_i\}_{l+1} = -\lambda_{2i}\{\bar{p}\}_l \end{cases} \quad l = 1, ..., 2i$$

End

The stopping criterion depends on $\widehat{r}_k$, the residual computed from MQMRCGSTAB.

## 5 Numerical experiments

We next compare the performance of the modified QMR-type methods with that of the standard QMR-type, BiCGSTAB and VGMRES [15] methods. In addition, we illustrate the effect of preconditioning and reordering in the convergence of the proposed algorithms. For our test runs, we always chose $x_0 = 0$ as starting vector. The stopping criterion used in the iterations was $||r_k||/||b|| < 10^{-10}$, with $r_k = b - Ax_k$ being the true residual. In all experiments, we have chosen the best preconditioning version (left, right or both sides) of each algorithm in each example. Thus, although the MQMR-type algorithms are equivalent to the standard ones in exact arithmetic, this was only appreciable when non preconditioned algorithms were applied. In these cases, the convergence paths were similar at the beginning of the iterations, but soon they became different.

All experiments were run on a XEON Precision 530 with Fortran Double Precision.

### *5.1 Example 1*

This example has been taken from the *Harwell-Boeing Sparse Matrix Collection*. It is one of the OILGEN collection matrices, called *orsreg1*. It comes from an oil reservoir problem on a $21 \times 21 \times 5$ grid, which yields a system of $2205$ equations with $14133$ non zero entries in the matrix. The convergence behaviour of non preconditioned BiCGSTAB, QMRCGSTAB and MQMRCGSTAB algorithms is represented in figure 1(a). We can see the smoother convergence of QMR type methods compared to that of BiCGSTAB. Although the standard version of QMRCGSTAB is faster than the modified one, we have observed in our experiments that this behaviour is inverted when any preconditioning is used (see table 1). For example, figure 1(b) represents the convergence plots of these algorithms with Jacobi preconditioner. Note again the smoother convergence of QMR type methods, with MQMRCGSTAB showing the best performance by a significant margin.

Table 1
Number of iterations of BiCGSTAB, QMRCGSTAB and MQMRCGSTAB with several preconditioners

| *orsreg1* | NP | Jacobi | ILU | SSOR | Diagopt |
|---|---|---|---|---|---|
| BiCGSTAB | 1090 | 601 | 50 | 131 | 400 |
| QMRCGSTAB | 648 | 397 | 52 | 162 | 610 |
| MQMRCGSTAB | 826 | 306 | 50 | 156 | 300 |

### *5.2 Example 2*

The second example, called *watt1*, has been selected from the *Harwell-Boeing Sparse Matrix Collection* too. It also arises from an oil reservoir engineering problem and the linear system has $1856$ equations with $11360$ non zero entries in the matrix.

Figures 2(a) and 2(b) show the performance of standard and modified QMR type methods, respectively, using an approximate inverse preconditioner with diagonal pattern. In this case, although the modified versions of QMR and TFQMR reach convergence before the standard ones, here the QMRCGSTAB is faster than MQMRCGSTAB.

### *5.3 Example 3*

The third numerical experiment (*cuaref*) is related to the convection-diffusion equation in a square $\Omega = (0,1) \times (0,1)$

$$v_1 \frac{\partial u}{\partial x} + v_2 \frac{\partial u}{\partial y} - K \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = 0 \qquad \text{in } \Omega$$

where $K = 1$ and the velocity field is,

$$v_1 = C\,(y - 1/2)\left(x - x^2\right), \quad v_2 = C\,(1/2 - x)\left(y - y^2\right)$$

being $C = 10^5$. We impose Dirichlet boundary condition $u = 0$ on $x = 1$ and $u = 1$ on $x = 0$, and Neumann condition $\frac{\partial u}{\partial y} = 0$ on $y = 0$ and $y = 1$. An adaptive finite element discretization leads to a nonsymmetric linear system of $7520$ equations with a matrix having $52120$ non zero entries.

In figure 3 we represent the convergence of some Krylov subspace methods with SSOR preconditioning. Note that MQMRCGSTAB reaches convergence at a lower number of iterations than BiCGSTAB, QMRCGSTAB and VGMRES. At first, MQMRCGSTAB curve is close to VGMRES one, while at the end it has the same behaviour than QMRCGSTAB. This phenomenon has been repeated in many others experiments not included here.

In this problem the values of $K$ and $C$ lead to an ill-conditioned system of equations. This fact may be understood by observing the spectrum of the original matrix from cuaref in Figure 4. Indeed, the convergence of all the algorithms without preconditioning was very slowly in this example.

### *5.4 Example 4*

The last linear system arise from a two-dimensional convection-diffusion problem (*convdifhor*) defined in a square $\Omega$, see figure 5,

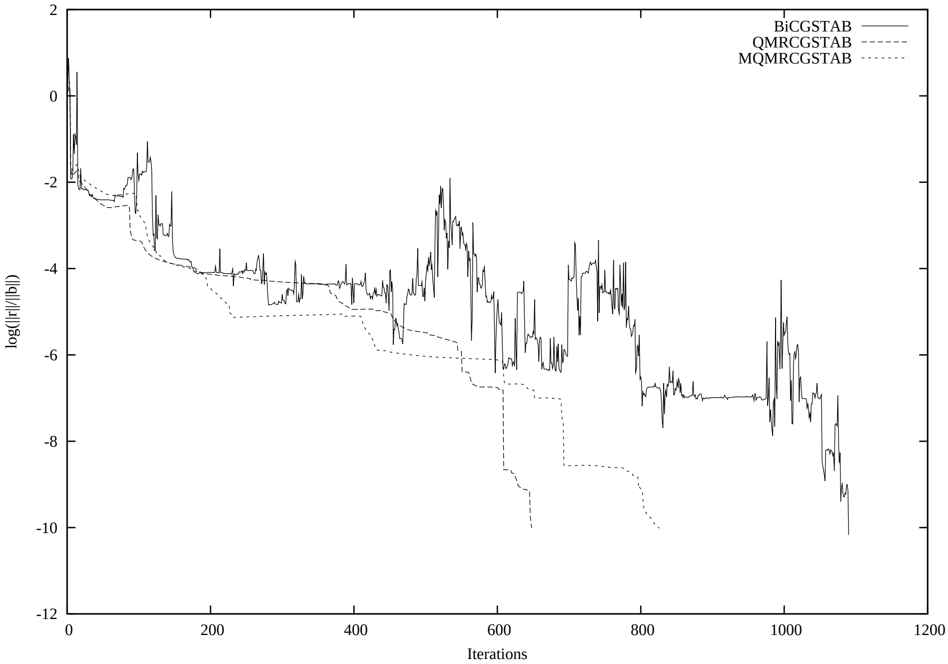
$$v_1 \frac{\partial u}{\partial x} - K \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = F \qquad \text{in } \Omega$$

with a velocity field given by,

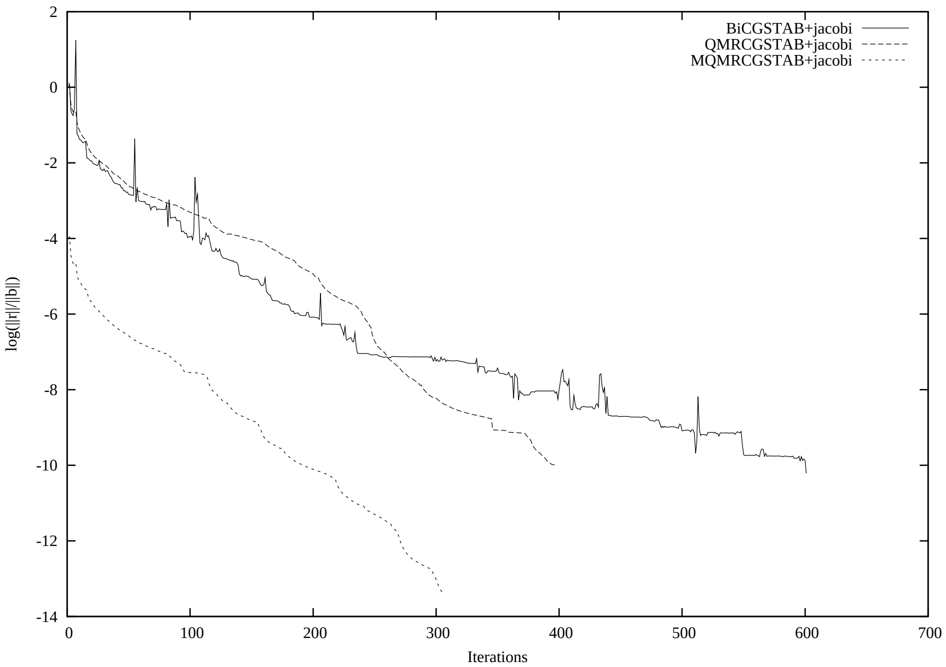
$$v_1 = 10^4 \left(y - 1/2\right) \left(x - x^2\right) \left(1/2 - x\right)$$

Here $K = 10^{-5}$ in $\Omega_2$ and $K = 10^2$ elsewhere, and $F = 10^3$ in $\Omega_3$ and $F = 1$ elsewhere. The boundary conditions are the same as in example 3. Again, an adaptive finite element discretization yields a nonsymmetric system of $3423$ equations where the matrix contains $23579$ non zero entries.

Figure 6 illustrates the effect of ordering on the convergence of Preconditioned MTFQMR when we use ILU(0). In this example, Minimum Degree [18], Minimum Neighbouring [21] and Reverse Cuthill-McKee [6,17] reordering algorithms have been applied (see also [10] for the effect on sparse approximate inverse preconditioners). The results show that a suitable reordering technique may reduce about 50% the number of iterations when we use an incomplete factorisation as preconditioner.

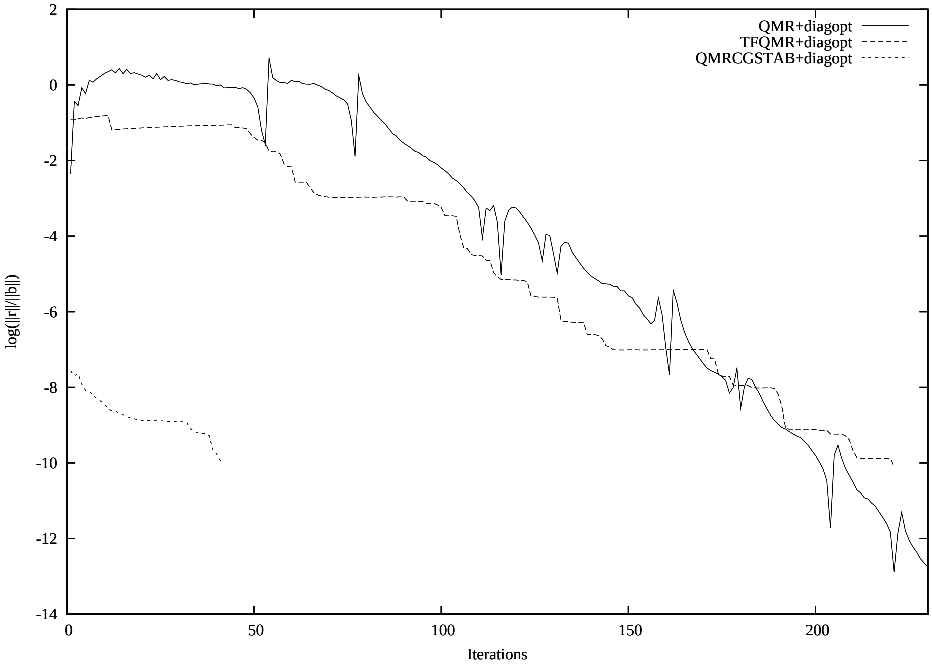



(a) Unpreconditioned

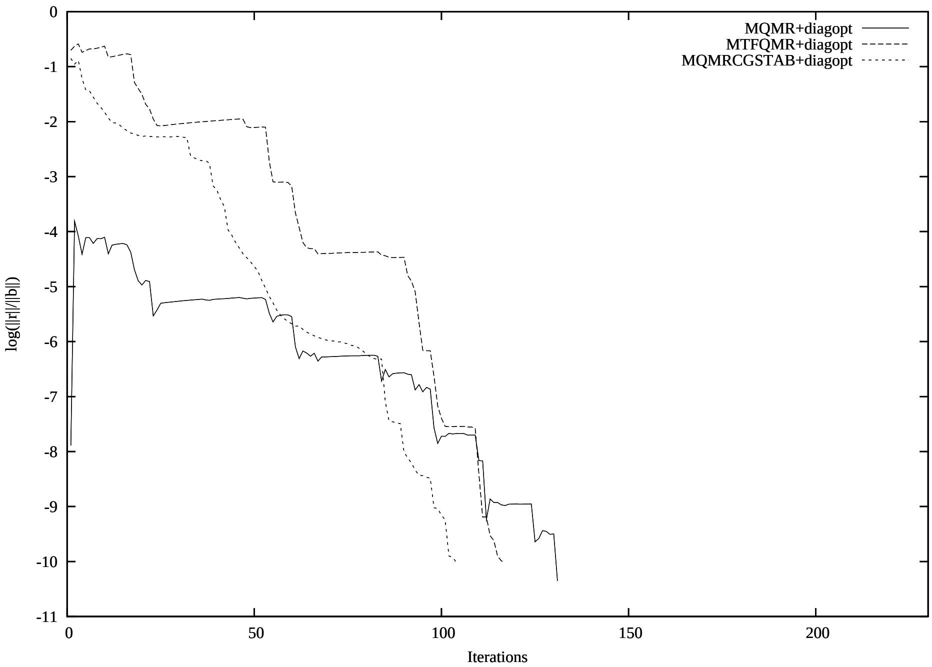



(b) Preconditioned with Jacobi

Fig. 1. Convergence of stabilised biorthogonalisation methods for *orsreg1*




(a) Standard QMR-type algorithms




(b) Modified QMR-type algorithms

Fig. 2. Convergence of standard and modified QMR algorithms with diagonal approximate inverse preconditioning for *watt1*

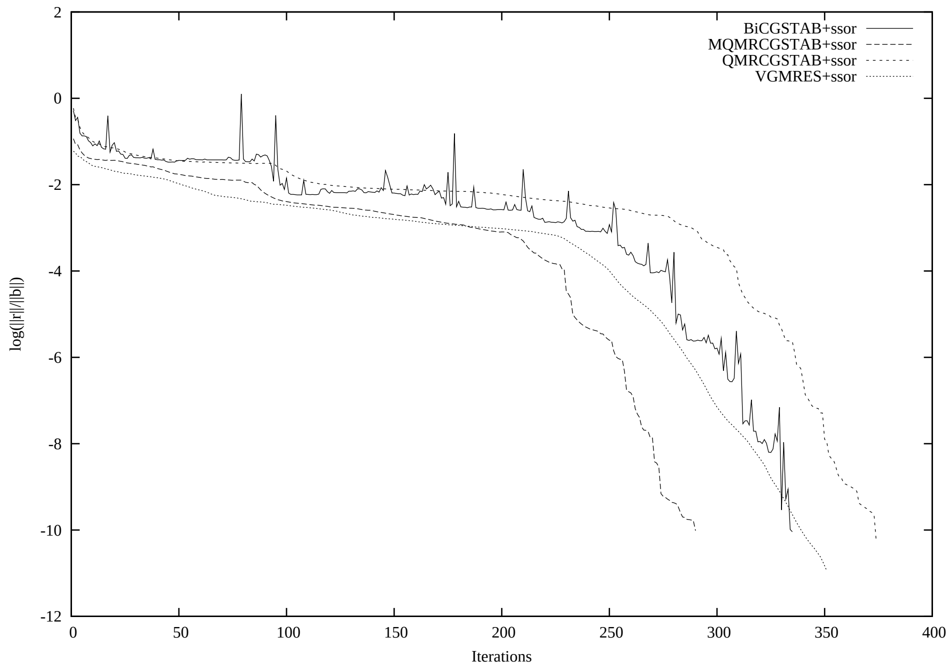



Fig. 3. Performance of several Krylov subspace methods with SSOR preconditioning for *cuaref* (7520 equations)

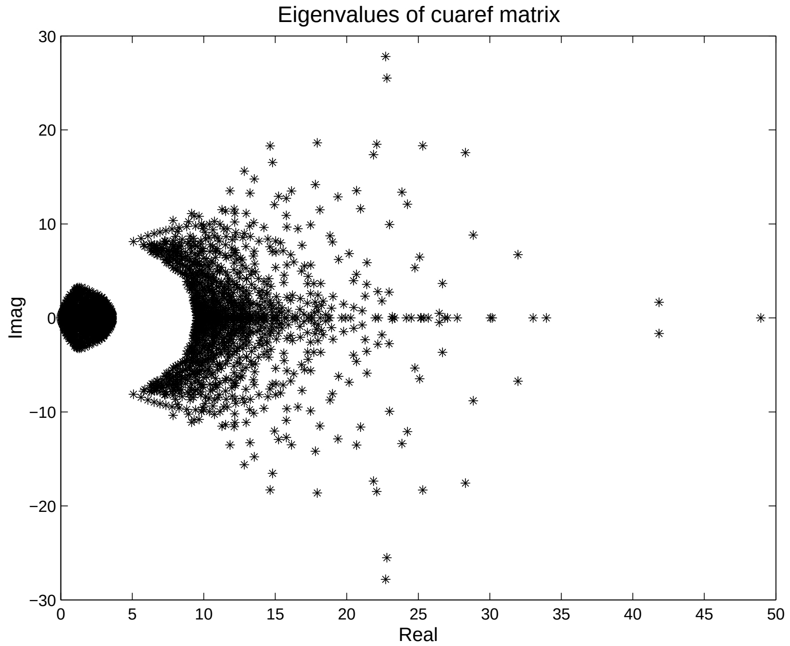



Fig. 4. Eigenvalues of matrix *cuaref*. Only the highest 1300 and the lowest 1000 eigenvalues are plotted.

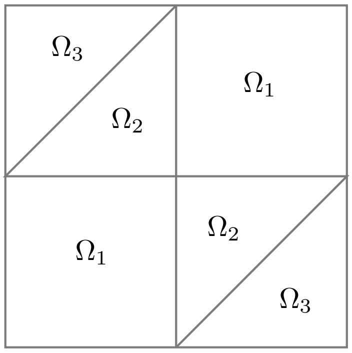



Fig. 5. Domain of example 4

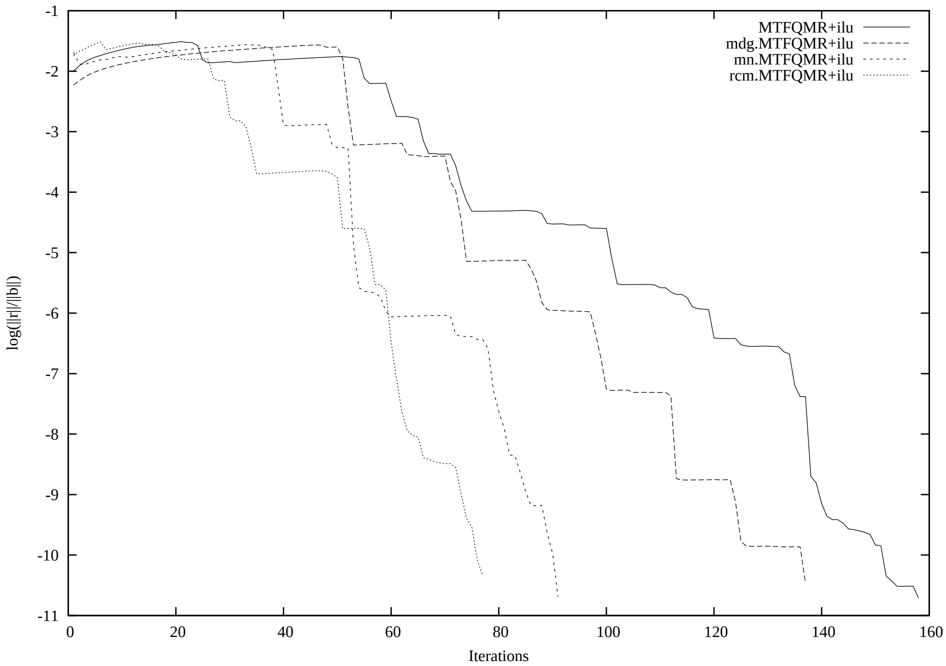



Fig. 6. Effect of ordering on the convergence of MTFQMR with ILU(0) preconditioning *convdifhor* (3423 equations)

## 6 Conclusion

The modified versions of QMR methods may lead to faster convergence than the standard ones. This effect is remarked if preconditioning is used. The studied numerical experiments shows that the modified algorithms are closer to GMRES at the beginning of the convergence process but at lower computational cost, and work like the standard QMR methods at the last iterations. We have verified that ordering techniques improve the rate of convergence and the computational cost of the modified algorithms, specially with ILU and SSOR preconditioning.

## Acknowledgements

This work has been partially supported by the MCYT of Spanish Government and FEDER, grant contract REN2001-0925-C03-02/CLI.

## References

[1] O. Axelsson, *"Iterative Solution Methods"*, Cambridge University Press, 1996.

[2] M. Benzi y M. Tuma, *"A sparse approximate inverse preconditioner for nonsymmetric linear systems"*, SIAM J. Sci. Comput.19, 3, 968-994, 1998.

[3] M. Benzi, D.B. Szyld y A. van Duin, *"Orderings for incomplete factorization preconditioning of nonsymmetric problems"*, SIAM J. Sci. Comput., 20, 5, 1652-1670, 1999.

[4] M. Benzi, *"Preconditioning techniques for large linear systems: a survey"*, J. Comput. Physics, 182, 418-477, 2002.

[5] T.F. Chan, E. Gallopoulos, V. Simoncini, T. Szeto, C.H. Tong, *"A Quasi-Minimal Residual Variant of the Bi-CGSTAB Algorithm for Nonsymmetric Systems"*, SIAM J. Sci. Statist. Comput., 15, 338-247, 1994.

[6] E.H. Cuthill y J.M. Mckee, *"Reducing the Bandwidth of Sparse Symmetric Matrices"*, Proc. 24th National Conference of the Association for Computing Machinery, Brondon Press, New Jersey, 157-172, 1969.

[7] I.S. Duff, G. A. Meurant, *"The effect of ordering on preconditioned conjugate gradients"*, BIT, 29, 635-657, 1989.

[8] L.C. Dutto, *"The effect of ordering on preconditioned GMRES algorithm, for solving the compressible Navier-Stokes equations"*, Int. J. Num. Meth. Eng., 36, 457-497, 1993.

[9] R. Fletcher, *"Conjugate Gradient Methods for Indefinite Systems"*, Lectures Notes in Math., 506, 73-89, 1976.

[10] E. Flórez, D. García, L. González, G. Montero, *"The effect of ordering on Sparse Approximate Inverse Preconditioners for Nonsymmetric problems"*, Advances in Engineering Software, 33, 611-619, 2002.

[11] R.W. Freund, *"A Transpose-Free Quasi-Minimal Residual Algorithm for non-Hermitian Linear Systems"*, SIAM J. Sci. Comput., 14, 470-482, 1993.

[12] R.W. Freund, N.M. Nachtigal, *"QMR: a Quasi-Minimal Residual Method for non-Hermitian Linear Systems"*, Numerische Math., 60, 315-339, 1991.

[13] R.W. Freund, M.H. Gutknecht y N.M. Nachtigal, *"An Implementation of the Look-ahead Lanczos Algorithm for non-Hermitian Matrices"*, SIAM J. Sci. Comput., 14, 1, 137-158, 1993.

[14] R.W. Freund y N.M. Nachtigal, *"An Implementation of the QMR Method Based on Coupled Two-Term Recurrences"*,SIAM J. Sci. Comput., 15, 2, 313-337, 1994.

[15] M. Galán, G. Montero, G. Winter, *"A Direct Solver for the Least Square Problem Arising From GMRES(k)"*, Com. Num. Meth. Eng., 10, 743-749, 1994.

[16] D. García, *"Estrategias para la resolución de grandes sistemas de ecuaciones lineales. Métodos de Cuasi-Mínimo Residuo Modificados"*, PhD., University of Las Palmas de

Gran Canaria, 2003 (in Spanish).

[17] A. George, *“Computer Implementation of the Finite Element Method”*, Report Stan CS-71-208, 1971.

[18] A. George y J.W. Liu, *“The Evolution of the Minimum Degree Ordering Algorithms”*, SIAM Rev., 31, 1-19, 1989.

[19] M. Grote y H.D. Simon, *“Parallel Preconditioning and Approximate Inverses on the Connection Machine”*, Sixth SIAM Conference on Parallel Processing for Scientific Computing, 2, 519-523, Philadelphia, 1992.

[20] M.R. Hestenes y E. Stiefel, *“Methods of Conjugate Gradients for Solving Linear Systems”*, Jour. Res. Nat. Bur. Sta., 49, 6, 409-436, 1952.

[21] G. Martin, *“Methodes de Preconditionnemen par Factorisation Incomplete”*, Memoire de Maitrise, Universite Laval, Quebec, Canada, 1991.

[22] G. Montero, L. González, E. Flórez, M.D. García, A. Suárez, *“Approximate Inverse Computation Using Frobenius Inner Product”*, Num. Lin. Alg. Appl., 9, 239-247, 2002.

[23] N.M. Nachtigal, S.C. Reddy y L.N. Trefethen, *“How Fast Are Nonsymmetric Matrix Iterations?”*, SIAM J. Matr. Anal. Appl., 13, 3, 796-825, 1992.

[24] Y.Saad, *“Iterative Methods for Sparse Linear Systems”*, PWE Publishing Company, Boston, 1996.

[25] Y. Saad y M. Schultz, *“GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems”*, SIAM J. Sci. Statist. Comput., 7, 856-869, 1986.

[26] P. Sonneveld, *“CGS: a Fast Lanczos-Type Solver for Nonsymmetric Linear Systems”*, SIAM J.Sci. Statist. Comput., 10, 36-52, 1989.

[27] A. Suárez, D. García, E. Flórez y G. Montero, *“Preconditioning Krylov Methods”*, en Nato Book Series, Algorithms for Sparse Large Scale Linear Algebraic Systems: State of the Art and Applications in Science and Engineering, 1997.

[28] H.A. van der Vorst,*”Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems”*, SIAM J. Sci. Comput., 13, 631-644, 1992.

[29] L. Zhou y H.F. Walker, *“Residual Smoothing Techniques for Iterative Methods”*, SIAM J. Sci. Comput., 15, 2, 297-312, 1994.