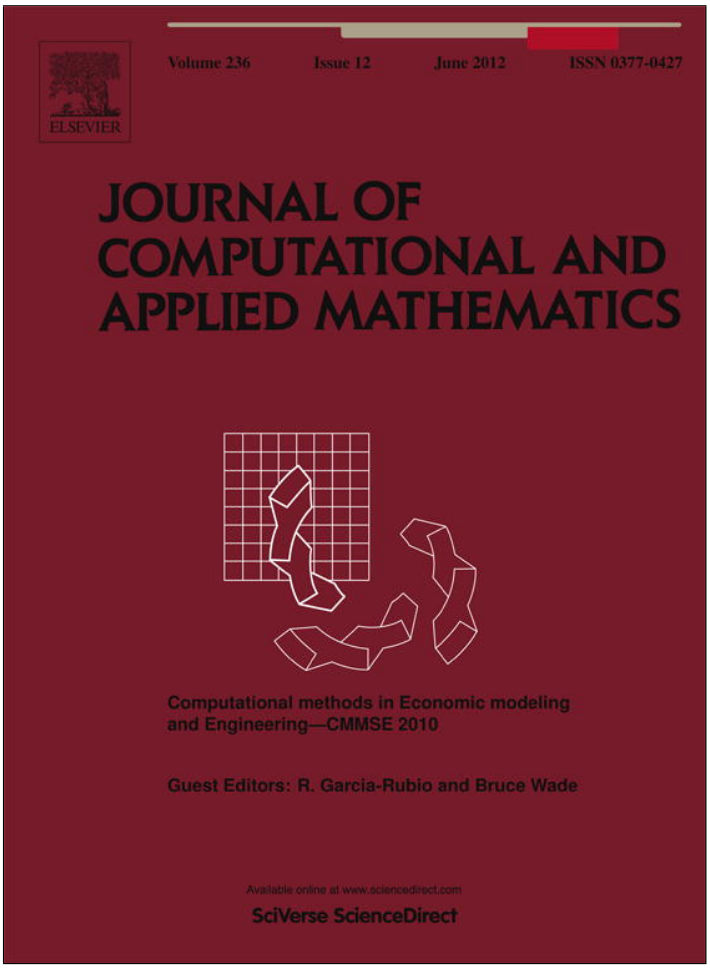

Journal of Computational and Applied Mathematics 236 (2012) 2938–2953

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A generalization of *s*-step variants of gradient methods

J.A. Alvarez-Dios [a], J.C. Cabaleiro [b], G. Casal [c,*]

[a] *Departamento de Matemática Aplicada, Facultade de Matemáticas, Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain*

[b] *Departamento de Electrónica e Computación, Escola Técnica Superior de Enxeñaría, Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain*

[c] *Departamento de Matemática Aplicada, Escola Politécnica Superior, Universidade de Santiago de Compostela, 27002 Lugo, Spain*

ARTICLE INFO

Article history:
Received 30 August 2010
Received in revised form 16 January 2012

Keywords:
Iterative methods
s-step
Large linear systems
Krylov subspace

ABSTRACT

The *s*-step methods were proposed by Chronopoulos to gain efficiency in parallel programming of iterative methods for linear systems. They are variants of classical iterative methods based on the construction of a Krylov subspace basis on each iteration. These s-step methods were inferred from algorithms like the Conjugate Gradient, Generalized Conjugate Residual or the Minimal Residual. They converge for all symmetric, nonsymmetric definite and some nonsymmetric indefinite matrices. In this paper, we introduce an *s*-step variant of a General Orthogonalization Algorithm, that is, a generalization of s-step variants of gradient methods. We prove convergence and obtain error estimates. We also describe an Orthomin variant, together with a convergence theorem. From this we derive the well known *s*-step methods as particular cases, and some which are newfound to our knowledge. This provides a unified framework to derive and study *s*-step methods. Some of the methods obtained are convergent for every nonsingular matrix. Finally, we give some numerical results for the new proposed methods, showing that the parallel implementation of these overcomes the original ones.

1. Introduction

The computational cost of an algorithm is evaluated in terms of arithmetic operations and communication time. Communication costs are much higher than arithmetic costs. Thus, the recent research on parallel programming efficiency focuses on minimizing the number of communications [1]. An important research subject is the study of iterative method solvers for large linear systems [2]. In these numerical methods, most required computations are vector–vector and matrix–vector operations. In the language of the Basic Linear Algebra Subprograms (BLAS) [3] or the Parallel Basic Linear Algebra Subprograms (PBLAS) [4], they primarily translate as SDOTS (inner products) and SAXPYS (vector updates as a linear combination of two vectors) i.e., level 1 BLAS operations. On the other hand, BLAS 2 and BLAS 3 operations, based on submatrix blocks, are much more efficient than BLAS 1 operations on parallel computers with optimized BLAS kernels. This is because the ratio between the number of operations performed and computer memory accesses increases as we raise the BLAS level and, with multiprocessors systems, the number of communications between nodes is reduced.

In order to improve the BLAS 2–3/BLAS 1 ratio, the *s*-step methods proposed in [5,6] are an alternative approach for using BLAS 3 operations in some iterative methods for linear systems. The efficiency of these methods on parallel computers is corroborated in [7–9].

The aim is to generalize these *s*-step variants to other Conjugate Gradient type methods in order to obtain iterative algorithms for the resolution of large linear systems, this also being valid even in the case of nonsymmetric and/or positive

* Corresponding author. Tel.: +34 982 823227; fax: +34 982 285926.
E-mail addresses: joseantonio.alvarez.dios@usc.es (J.A. Alvarez-Dios), jc.cabaleiro@usc.es (J.C. Cabaleiro), xerardo.casal@usc.es (G. Casal).

0377-0427/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2012.01.028

nondefinite matrices, with better performance in parallel programming. For such purpose we present an s-step variant of a General Orthogonalization Algorithm that can be seen, for example in [10] or [11], and we obtain different s-step variants of this method by previously fixing two parameter matrices.

This paper is organized as follows. In Section 2 the notation is established and the s-step variant of the methods proposed by Chronopoulos, which precedes this work, is presented. Section 3 describes a General Orthogonalization Algorithm which we call GOA for short. It is a generalization of the gradient type methods in [10] or [11]. Then an s-step variant of this method (s-GOA) is proposed and a theorem of convergence with some previous lemmas are proved. Section 4 proposes an Orthomin(m) variant of the s-GOA which generalizes the s-step Orthomin(m) method given in [6]. In this section, a convergence theorem for the s-step Orthomin(m) method proposed is also proved. Section 5 describes the well known s-step methods obtained as particular cases of the s-GOA and two new methods are proposed, namely the s-Minimal Error Algorithm and the s-Biconjugate Gradient. In Section 6 we present some numerical results for the new proposed methods. Finally, Section 7 contains the conclusions of this work.

2. Background

It is assumed throughout this paper that A is a general square nonsingular matrix of order n, $b \in \mathbb{R}^n$ a column vector, $\|\cdot\|$ is the 2-norm in $\mathbb{R}^n$ and $\mathcal{M}_{n\times s}(\mathbb{R})$ the set of real matrices of order $n \times s$. The matrix norm of A induced by the 2-norm is

$$\|A\| = \max_{v\neq 0} \frac{\|Av\|}{\|v\|} \tag{1}$$

and the condition number of A with respect to the 2-norm

$$\mathrm{cond}(A) = \|A\| \left\|A^{-1}\right\|. \tag{2}$$

Denote the symmetric and the antisymmetric part of A by:

$$A^S = \frac{1}{2}(A + A^t) \tag{3}$$

$$A^{aS} = \frac{1}{2}(A - A^t). \tag{4}$$

Let M be a positive definite symmetric matrix of order n, with maximum and minimum eigenvalues denoted by $\lambda_{\max}(M)$ and $\lambda_{\min}(M)$, respectively. We recall that, for every $v \in \mathbb{R}^n$

$$\langle v, Mv\rangle = v^t M v \geq \lambda_{\min}(M)\,\|v\|^2. \tag{5}$$

If $v_1, \ldots, v_s$ are column vectors, we denote by $(v_1\,|v_2\ldots|\,v_s)$ the matrix they form and $\mathcal{L}\{v_1, \ldots, v_s\}$ will stand for the vector subspace they span. In an analogous way, if $A_1, \ldots, A_s$ are real matrices, we denote by $(A_1\,|A_2\ldots|\,A_s)$ the matrix they form and $\mathcal{L}\{A_1, \ldots, A_s\}$ will stand for the vector subspace spanned by all columns of all matrices. The aim of the iterative methods, the object of this paper, is the numerical resolution of the linear system

$$Ax = b \tag{6}$$

whose exact solution will be denoted by c.

Therefore, it is worthwhile seeking techniques for accelerating the execution of these methods, such as by including preconditioners [12], or by using wavelets to modify the linear system into another more sparse one [13], or by modifying the methods to obtain better performance in parallel processing. Among these last there would be the s-step variants treated in [5–8], which is the object of generalization in this article.

We shall now recall some elementary key definitions. For each $v \in \mathbb{R}^n$, $v \neq 0$ and $s \in \mathbb{N}, s < n$, we call the vector subspace $\mathcal{L}\{v, Av, A^2v, \ldots, A^{s-1}v\}$ a Krylov subspace of order s, and we denote it by $\mathcal{K}_s(A, v)$.

If $\dim(\mathcal{K}_s(A, v)) < s$, and therefore the dimension of $\mathcal{K}_s(A, v)$ were not maximum, the inverse of A would be a polynomial in A of degree $s-1$ at most and we could easily construct the exact solution of the system (see, for example, [12, page 149]). We often refer to this circumstance as *lucky breakdown*, something which is highly unlikely in practice.

The s-step variant of the *Conjugated Gradient* algorithm (s-CG) was introduced by Chronopoulos and Gear in [5]. Subsequently, in [6], Chronopoulos proposes s-step variants of other methods which are convergent for nonsymmetric definite and some nonsymmetric indefinite coefficient matrices. More specifically, article [6] deals with the s-step variants of the *Generalized Conjugated Residual method* (s-GCR), of the *Minimal Residual* (s-MR) and of the Orthomin(m) (s-Orthomin(m)), and particularly for the case s-Orthomin(1) known as *s-Conjugate Residual method* (s-CR).

In each iteration of the s-step variant, these algorithms compute a base of a Krylov subspace $\mathcal{K}_s(A, v)$ of dimension s and then, by a convenient projection method over the subspace, they calculate the next iterate that minimizes the error $e_i = \|c - x_i\|$, or the residual norm, $r_i = b - Ax_i$ according to the respective method (s-CG, s-GCR, s-Orthomin (m), etc.), $x_i \in \mathbb{R}^n$ being the i-th iterate.

The convergence of these methods in at most $[n/s]$ iterations is proved in [5,6] for symmetric positive definite matrices in s-CG, for nonsymmetric definite matrices in s-CGR and for indefinite matrices, with definite symmetric part, in s-Orthomin(m) and s-GCR. Therefore these methods are not convergent for every nonsingular matrix.

The s-step variants of the GMRES and the Double Orthogonal Series can be seen in [14,15], which converge for every nonsingular matrix. Basing on these, we shall try to construct valid methods for a general nonsingular matrix.

3. The s-step variant of the General Orthogonalization Algorithm

If matrix A is neither necessarily symmetric nor positive definite, there is a more general algorithm than the Conjugate Gradient method, which we call the General Orthogonalization Algorithm (GOA). In what follows we describe this method in a summarized way (see [10] or [11], for example).

Let $Ax = b$ be the linear system of order n with nonsingular matrix A. Let H, K be square matrices of order n with positive definite symmetric part. We set:

$$N = A^t H^S A \quad \text{and} \quad M = L^t N L \tag{7}$$

where $L^t L$ is the Cholesky factorization of the symmetric part of K, and then $K^S = LL^t$. For all $r \in \mathbb{R}^n$ let us define $E(r) = \langle r, Hr\rangle$. From the equality $E(r) = \langle H^t r, r\rangle = \langle r, Hr\rangle$ we get the alternative definition:

$$E(r) = \left\langle r, \frac{1}{2}(H + H^t)r \right\rangle = \langle r, H^S r\rangle. \tag{8}$$

Then $E(r)$ must be a convex function. Next we write the GOA, which is presented in [10]:

Algorithm 3.1 (*GOA*)**.**
Let $x_0 \in \mathbb{R}^n$,
$r_0 = b - Ax_0 = A(x - x_0)$
$g_0 = A^t H^S r_0 = A^t H^S A(x - x_0) = N(x - x_0)$
$p_0 = Kg_0$
For $i = 0, 1, \ldots$ until convergence **Do**:

$$\alpha_i = \frac{\langle g_i, p_i\rangle}{\langle p_i, Np_i\rangle} \tag{9}$$

$$x_{i+1} = x_i + \alpha_i p_i \tag{10}$$

$$g_{i+1} = g_i - \alpha_i N p_i = A^t H^S r_{i+1} \tag{11}$$

$$\beta_{i+1}^l = -\frac{\langle Kg_{i+1}, Np_l\rangle}{\langle p_l, Np_l\rangle}, \quad l = 0, \ldots, i \tag{12}$$

$$p_{i+1} = Kg_{i+1} + \sum_{l=0}^{i} \beta_{i+1}^l p_l \tag{13}$$

EndFor.

We denote as vectors g_i the general residues, and as vectors p_i the general descent directions. The following results are proved in [10]:

(i) $r_i = b - Ax_i$, $i = 0, 1, 2, \ldots$.
(ii) $\langle p_i, Np_j\rangle = 0$ for all $i \neq j$.
(iii) $\langle g_i, p_j\rangle = 0$ for all $0 \leq j \leq i$.
(iv) $\langle g_i, Kg_j\rangle = 0$ for all $0 \leq j \leq i$.
(v) If $g_0 \cdots g_{n-1} \neq 0$ then $g_n = 0$.
(vi) $\mathcal{L}\{p_0, \ldots, p_{i-1}\} = \mathcal{L}\{Kg_0, Kg_1, \ldots, Kg_{i-1}\} = \mathcal{K}_i(KN, Kg_0)$.
(vii) The residual r_{i+1} minimizes $E(r)$ over the affine subspace $x_0 + \mathcal{L}\{p_0, \ldots, p_{i-1}\}$.

Consequently the GOA converges in at most n iterations. Moreover, if we denote by $E_i = E(r_i) = \langle r_i, H^S r_i\rangle$, then the following error estimate is proved in [10]:

$$E_i \leq E_0 \left(1 - \frac{\lambda_{\min}(L^t (K^{-1})^S L)}{\operatorname{cond}(M)}\right)^i \tag{14}$$

and, if matrix K is symmetric:

$$E_i \leq E_0 \left(\frac{\operatorname{cond}(M) - 1}{\operatorname{cond}(M) + 1}\right)^{2i}. \tag{15}$$

Observe that, if K is a symmetric matrix, then (iv) is valid for every $i \neq j$ and taking the value of Np_l from (11) to (12), we get $\beta_{i+1}^l = 0$ for all $0 \leq l < i$ and the sum in (13) reduces to the last term. In this case, storage of the preceding directions p_l, $l = 0, \ldots, i-1$, is not necessary to compute p_{i+1}. On the contrary, if K is a nonsymmetric matrix, and more than a few iterations are needed, then the storage requirements become prohibitive. To circumvent this, the general Orthomin(m) method computes p_{i+1} by N-orthogonalizing to the m preceding directions only. The parameter m is previously chosen, usually between 3 and 10, depending on the order of the matrix. Giving matrix H and K particular values in the GOA, we obtain some known methods like the Conjugate Gradient, Preconditioned Conjugate Gradient, Normal Equation, Minimal Error, Generalized Conjugate Residual and Axelsson's Minimal Residual [10].

In order to unburden the notation, we define:

Definition 3.1. Let $n, s \in \mathbb{N}$ $(s < n)$, $M \in \mathcal{M}_{n\times n}(\mathbb{R})$. Then we define the application $\Delta_M : \mathbb{R}^n \longrightarrow \mathcal{M}_{n\times s}(\mathbb{R})$ by:

$$\Delta_M(v) = (v \,|Mv|\, M^2v \,|\ldots|\, M^{s-1}v) \quad \text{for all } v \in \mathbb{R}^n. \tag{16}$$

We shall often use the following elementary but important properties:

1. $\Delta_M(\alpha u + \beta v) = \alpha\Delta_M(u) + \beta\Delta_M(v)$ for all $\alpha, \beta \in \mathbb{R}$ and $u, v \in \mathbb{R}^n$, then Δ_M is a linear application.
2. $\Delta_M(M^k v) = M^k\Delta_M(v)$ for all $v \in \mathbb{R}^n$ and $k \in \mathbb{N}$.

Then $\mathcal{K}_s(KN, Kg_0)$ is the vector subspace generated by the column vectors of the matrix $\Delta_{KN}(Kg_0)$. We define the s-step variant of the GOA (s-GOA):

Algorithm 3.2 (*s-GOA*)**.**
Let $x_0 \in \mathbb{R}^n$
$r_0 = b - Ax_0$
$g_0 = A^tH^Sr_0$
$P_0 = \Delta_{KN}(Kg_0) = Q_0$
For $i = 0, 1, 2, \ldots$ until convergence **Do**

$$W_i = (P_i)^tNP_i \tag{17}$$

$$z_i = (P_i)^tg_i \tag{18}$$

$$y_i = (W_i)^{-1}z_i \tag{19}$$

$$x_{i+1} = x_i + P_iy_i \tag{20}$$

$$g_{i+1} = g_i - NP_iy_i = A^tH^Sr_{i+1} \tag{21}$$

$$Q_{i+1} = \Delta_{KN}(Kg_{i+1}) \tag{22}$$

For $j = 0, \ldots, i$ **Do**:

$$B_{i+1}^j = -W_j^{-1}(P_j)^tNQ_{i+1} \tag{23}$$

EndFor

$$P_{i+1} = Q_{i+1} + \sum_{j=0}^{i} P_jB_{i+1}^j \tag{24}$$

EndFor.

By induction on i we can obtain in (21) the following equation for the residual:

$$r_{i+1} = r_i - AP_iy_i. \tag{25}$$

Comparing GOA (Algorithm 3.1) with s-GOA (Algorithm 3.2) it is easy to verify that BLAS 1 and BLAS 2 become BLAS 2 and BLAS 3 operations, respectively.

We establish the following lemma relating direction and general residual vectors of s-GOA:

Lemma 3.1. *It holds that:*

(a) $(P_i)^tNP_j = 0$ *for all* $i \neq j$.
(b) $(P_j)^tg_i = 0$ *for all* $i > j$.
(c) $(P_i)^tg_i = (Q_i)^tg_i$.
(d) $(Q_j)^tg_i = 0$ *for all* $i > j$.
(e) $(P_i)^tNQ_j = 0$ *for all* $i > j$.
(f) $(P_i)^tNP_i = (P_i)^tNQ_i$.
(g) $(P_i)^tg_j = (P_i)^tg_0$ *for all* $i \geq j$.

Proof. (a) We have:

$$(P_j)^t NP_i = (P_j)^t NQ_i + (P_j)^t N \sum_{k=0}^{i-1} P_k B_i^k \tag{26}$$

and

$$(P_j)^t NP_j B_i^j = -(P_j)^t NQ_i. \tag{27}$$

Therefore from Eqs. (17) and (23) in Algorithm 3.2, we can easily prove by induction on $i > j$ that:

$$(P_j)^t NP_i = 0. \tag{28}$$

(b) For a fixed $j \in \mathbb{N}$ we obtain by induction on i that g_i is orthogonal to P_j for all $i > j$:
If $i = j + 1$, since

$$g_{j+1} = g_j - NP_j y_j \tag{29}$$

we conclude that g_{j+1} is orthogonal to P_j by definition of y_j, W_j and z_j.

Now suppose that g_i is orthogonal to $\mathcal{P}_j$, with $i > j + 1$. Then, the orthogonality between g_{i+1} and P_j is a consequence of the induction hypothesis and N-orthogonality between P_i and P_j.

(c) This equality follows from definition of P_i and (b).

(d) Since $Q_j = P_j - \sum_k^{j-1} P_k B_j^k$ then (b) implies (d).

(e) This is also from $Q_j = P_j - \sum_k^{j-1} P_k B_j^k$ and (a).

(f) This equality follows from definition of P_i and (a).

(g) The identity $g_j = g_{j-1} - NP_{j-1} y_{j-1}$ and induction give (g). □

Remember that $\{p_0, p_1, \ldots, p_{(i+1)s-1}\}$ are the direction vectors computed in GOA and we denote by $p_i^1, \ldots, p_i^s$ the direction vectors of s-GOA in each iteration, and then $P_i = (p_i^1 | \cdots | p_i^s)$. Now, we can establish the following lemma relating Krylov and direction subspaces generated in both algorithms:

Lemma 3.2. *Let $i, s \in \mathbb{N}$ be such that $s(i + 1) \leq n$. Suppose that $g_i \neq 0$. If $\dim \mathcal{K}_{s(i+1)}(KN, Kg_0) = s(i + 1)$ then:*

$$\text{£}\{P_0, \ldots, P_i\} = \bigoplus_{j=0}^{i} \mathcal{K}_s(KN, Kg_j) = \mathcal{K}_{s(i+1)}(KN, Kg_0) = \text{£}\{p_0, p_1, \ldots, p_{(i+1)s-1}\}, \tag{30}$$

where $\bigoplus$ denotes the direct sum of vectorial subspaces.

Moreover r_{i+1} minimizes $E(r) = \langle r_i, H^S r_i \rangle$ over $x_0 + \text{£}\{P_0, \ldots, P_i\}$.

Proof. It is obvious that $\bigoplus_{j=0}^{i} \mathcal{K}_s(KN, Kg_j) = \text{£}\{Q_0, \ldots, Q_i\}$. Then, the equality

$$\text{£}\{P_0, \ldots, P_i\} = \bigoplus_{j=0}^{i} \mathcal{K}_s(KN, Kg_j) \tag{31}$$

is proved by induction since $P_0 = Q_0$ and the definition of P_i, from which we can also obtain that $Q_i = P_i - \sum_{j=0}^{i-1} P_j B_i^j$.

The equality

$$\bigoplus_{j=0}^{i} \mathcal{K}_s(KN, Kg_j) = \mathcal{K}_{s(i+1)}(KN, Kg_0) \tag{32}$$

is trivial for $i = 0$. The inclusion

$$\bigoplus_{j=0}^{i} \mathcal{K}_s(KN, Kg_j) \subset \mathcal{K}_{s(i+1)}(KN, Kg_0) \tag{33}$$

is proved by induction since:

$$(KN)^k Kg_i = (KN)^k (Kg_{i-1} - KNP_{i-1} y_{i-1}) \tag{34}$$

for $k \in \{0, \ldots, s - 1\}$ and, because of equality (31) and the induction hypothesis

$$(KN)^{k+1} p_{i-1}^1, \ldots, (KN)^{k+1} p_{i-1}^s \in \mathcal{K}_{s(i+1)}(KN, Kg_0) \tag{35}$$

so $(k + 1) + (s \cdot i - 1) = k + s \cdot i \leq s(i + 1) - 1$, and then $(KN)^k Kg_i \in \mathcal{K}_{s(i+1)}(KN, Kg_0)$.

The other inclusion

$$\mathcal{K}_{s(i+1)}(KN, Kg_0) \subset \bigoplus_{j=0}^{i} \mathcal{K}_s(KN, Kg_j) \tag{36}$$

is also proved by induction. Suppose that the inclusion verifies for $i-1$, $i>1$ fixed. Since

$$g_i = g_0 + \sum_{j=0}^{is-1} \lambda_j N(KN)^j Kg_0, \tag{37}$$

which stems from (31), (33) and by induction in $g_i = g_{i-1} - NP_{i-1}y_{i-1}$, we have, for $k \in \{0, \ldots, s-1\}$

$$(KN)^k Kg_i = (KN)^k Kg_0 + \sum_{j=0}^{is-1} \lambda_j (KN)^{j+k+1} Kg_0. \tag{38}$$

Now, if we prove that $\lambda_{s\cdot i-1} \neq 0$ then $\mathcal{K}_{s(i+1)}(KN, Kg_0) \subset \bigoplus_{j=0}^{i} \mathcal{K}_s(KN, Kg_j)$. But $\lambda_{s\cdot i-1} \neq 0$ because if $\lambda_{s\cdot i-1} = 0$ in (37), then $Kg_i \in \mathcal{K}_{s\cdot i}(KN, Kg_0)$. From the induction hypothesis and (31), $Kg_i \in \langle P_0, \ldots, P_{i-1}\rangle$, which implies, by part (b) of Lemma 3.1, that $\langle g_i, Kg_i\rangle = 0$. This is a contradiction if $g_i \neq 0$ because the symmetric part of K is positive definite.

The last equality $\mathcal{K}_{s(i+1)}(KN, Kg_0) = \mathcal{L}\{p_0, p_1, \ldots, p_{(i+1)s-1}\}$ is result (vi) of GOA's properties previously cited.

Finally, let r_{i+1} be the residual which corresponds to iterate x_{i+1}. From definition of r_{i+1} and by induction we have

$$r_{i+1} = r_0 - \sum_{j=0}^{i} (AP_j y_j). \tag{39}$$

Since $(AP_j)^t H^s r_0 = (P_j)^t g_0 = (P_j)^t g_j = z_j$ for all $j = 0, \ldots, i$, using (39) and part (a) of Lemma 3.1 we have that

$$E(r_{i+1}) = \langle r_0, H^S r_0\rangle - 2\sum_{j=0}^{i} y_j^t z_j + \sum_{j=0}^{i} y_j^t W_j y_j. \tag{40}$$

Since $E(r)$ is convex, r_{i+1} is the minimal of $E(r)$ over $x_0 + \mathcal{L}\{P_0, \ldots, P_i\}$ if the coefficient vectors y_j, with $j = 0, \ldots, i$, are the solutions of the linear systems $W_j y_j = z_j$, but this is true by the definition of y_j in s-GOA. □

Observation.

- As a result of the Lemma 3.2, for all $i = 0, 1, 2, \ldots$, matrices P_i have rank s. From definition of W_i, we have that, for all $v \in \mathbb{R}^s$, $v \neq 0$ then $v^t W_i v = \langle P_i v_i, NP_i v_i\rangle$ is strictly positive, and W_i is positive definite and consequently nonsingular.
- Let $\tilde{r}_i$ and r_i be the residual vectors in the ith iteration of the GOA and s-GOA, respectively. Since $E(r)$ is a convex function and from Lemma 3.2, if x_0 is the same for GOA and s-GOA then $\tilde{r}_{s\cdot i} = r_i$ in exact arithmetic.

From Lemmas 3.1 and 3.2 we obtain the following convergence theorem:

Theorem 3.1. *If all previous hypothesis hold, the s-GOA converges in at most* $[n/s]$ *iterations.*

Proof. Let $i \in \mathbb{N}$. Since Lemmas 3.1 and 3.2, if $g_i \neq 0$ then g_i is orthogonal to $\mathcal{K}_{s\cdot i}(KN, Kg_0)$. But $\dim \mathcal{K}_{s\cdot i}(KN, Kg_0) = s\cdot i$, and then, if $s\cdot i \geq n$ it is necessarily $g_i = 0$. This implies that $r_i = 0$, because $g_i = A^t H^S r_i$ and $A^t H^S$ is nonsingular. □

Moreover, thanks to the previous observation, we can establish the error estimate:

Theorem 3.2. *Under the hypothesis of Lemma 3.1, if* r_i *is the residual vector in the ith iteration of the s-GOA and* $E_i = E(r_i)$*, it verifies:*

$$E_i \leq E_0 \left(1 - \frac{\lambda_{\min}(L^t (K^{-1})^S L)}{\operatorname{cond}(M)}\right)^{s\cdot i}. \tag{41}$$

Moreover, if the matrix K is symmetric, we have:

$$E_i \leq E_0 \left(\frac{\operatorname{cond}(M) - 1}{\operatorname{cond}(M) + 1}\right)^{2s\cdot i}. \tag{42}$$

Proof. The proof is obvious from the error estimate (14) and (15) in GOA and since $\tilde{r}_{s\cdot i} = r_i$. □

As seen in GOA, we need all the previous matrices P_j, $j = 0, \ldots, i$ in s-GOA, for the computation of B^j_{i+1}. If more than a few iterations are needed, then the storage requirements become prohibitive. Thanks to the following lemma, when matrix K is symmetric it will only be necessary to store the last of the series of all previous matrices P_j for the computation of matrices B^j_{i+1}.

Lemma 3.3. *If matrix K is symmetric then, for $j = 0, \ldots, i-1$*

$$(P_j)^t N Q_{i+1} = 0. \tag{43}$$

Proof. Let $j \in \{0, \ldots, i-1\}$ fixed. Then $(P_j)^t N Q_{i+1}$ is a square matrix of order s whose kl element is $\langle (KN)^{k-1} K g_{i+1}, N p^l_j \rangle$, with $k, l \in \{1, \ldots, s\}$. If K is symmetric then

$$\langle (KN)^{k-1} K g_{i+1}, N p^l_j \rangle = \langle g_{i+1}, (KN)^k p^l_j \rangle. \tag{44}$$

From Lemma 3.2 we get $p^l_j \in \mathcal{K}_{s(j+1)}(KN, Kg_0)$ for $l = 1, \ldots, s$. So, if $j \leq i-1$ and $k \in \{1, \ldots, s\}$ then $(KN)^k p^l_j \in \mathcal{K}_{s(i+1)}(KN, Kg_0)$ because $k + (j+1)s - 1 \leq s + i \cdot s - 1 = s(i+1) - 1$. But again, from (b) of Lemmas 3.1 and 3.2, g_{i+1} is orthogonal to $P_0, \ldots, P_i$ whose columns span $\mathcal{K}_{s(i+1)}(KN, Kg_0)$, then g_{i+1} is orthogonal to $\mathcal{K}_{s(i+1)}(KN, Kg_0)$ and thus we conclude that if $0 \leq j \leq i-1$ then the right side of (44) is zero. □

In this way, if K is symmetric, Eqs. (23) and (24) of s-GOA becomes:

$$B_{i+1} = -W_i^{-1} (P_i)^t N Q_{i+1} \tag{45}$$

and

$$P_{i+1} = Q_{i+1} + P_i B_{i+1}. \tag{46}$$

4. The s-step variant of the Orthomin(m) method

If matrix K is nonsymmetric the GOA and its s-step variant require the storage of all the previous directions in each iteration. To avoid this, the Orthomin(m) method computes only the m previous directions, in the GOA, or the previous m Krylov subspaces in the s-GOA, $m \in \mathbb{N}$. Nevertheless, in exact arithmetic, it will not be possible to assure the convergence of the Orthomin(m) method in a finite number of iterations. The Orthomin(m) method is written in a similar way as Algorithm 3.1 replacing $l = 0, \ldots, i$ by $l = i - m + 1, \ldots, i$ in (12) and (13). Analogously the s-step variant of the Orthomin(m) is obtained replacing $j = 0, \ldots, i$ by $j = i - m + 1, \ldots, i$ in Eqs. (23) and (24) of the s-GOA Algorithm 3.2.

The following lemma for s-step Orthomin(m) method has a proof completely analogous to the corresponding parts of Lemmas 3.1 and 3.2:

Lemma 4.1. *In the s-step* Orthomin(m) *method, if $g_i \neq 0$ and* $\dim \mathcal{K}_{s(m+1)}(KN, Kg_{i-m}) = s(m+1)$ *then:*

(a) *P_i is N-orthogonal to P_j for all $i - m < j < i$.*
(b) *$(P_j)^t g_i = 0$ for all $i - m < j < i$.*
(c) *r_{i+1} minimizes $E(r)$ over $x_{i-m} + \langle P_{i-m}, \ldots, P_i \rangle$.*

Next we prove that s-step Orthomin(m) converges but, in exact arithmetic, it may require an infinite number of steps. First we tackle the following lemmas, also true for the s-GOA:

Lemma 4.2. *Let $E_i = E(r_i)$ where r_i, for $i = 0, 1, \ldots,$ is the residual vector generated in each iteration of the s-step* Orthomin(m) *then:*

$$E_{i+1} = E_i - (P_i^t g_i)^t W_i^{-1} P_i^t g_i \tag{47}$$

and thus:

$$\lim_{i \to \infty} (P_i^t g_i)^t W_i^{-1} P_i^t g_i = 0. \tag{48}$$

Proof. We have that

$$E_{i+1} = \langle r_{i+1}, H^S r_{i+1} \rangle = \langle r_i - A P_i y_i, H^S (r_i - A P_i y_i) \rangle \tag{49}$$

since $N = A^t H^S A$, we obtain

$$E_{i+1} = E_i - 2 \langle A P_i y_i, H^S r_i \rangle + \langle P_i y_i, N P_i y_i \rangle. \tag{50}$$

But, using that $W_i = P_i^t N P_i$, and substituting $y_i = W_i^{-1} P_i^t g_i$ in the last summand of (50)

$$\langle P_i y_i, N P_i y_i\rangle = y_i^t W_i y_i = (W_i^{-1} P_i^t g_i)^t W_i (W_i^{-1} P_i^t g_i) = g_i^t P_i W_i^{-1} P_i^t g_i. \tag{51}$$

Moreover, since $g_i = A^t H^S r_i$ we obtain that

$$\langle A P_i y_i, H^S r_i\rangle = (P_y W_i^{-1} P_i^t g_i)^t g_i = g_i^t P_i W_i^{-1} P_i^t g_i. \tag{52}$$

Substituting now in (50) we conclude that $E_{i+1} = E_i - (P_i^t g_i)^t W_i^{-1} P_i^t g_i$.

The second part of the lemma is a direct consequence that E_i is a nonnegative, monotonously decreasing succession and therefore convergent. In this way $\lim_{i\to\infty} E_i = \lim_{i\to\infty} E_{i+1}$, and thus

$$\lim_{i\to\infty} (P_i^t g_i)^t W_i^{-1} P_i^t g_i = \lim_{i\to\infty} E_i - \lim_{i\to\infty} E_{i+1} = 0. \quad \square \tag{53}$$

Lemma 4.3. *Under the preceding notations we have*

$$\left\| P_i^t g_i \right\| \geq \lambda_{\min}(K^S) \left\| g_i \right\|^2. \tag{54}$$

Proof. From definition of P_i from (b) in Lemma 4.1 we get $P_i^t g_i = Q_i^t g_i$. Then

$$\left\| P_i^t g_i \right\|^2 = \left\| Q_i^t g_i \right\|^2 = \sum_{k=0}^{s-1} \left(\langle (KN)^k K g_i, g_i\rangle^2 \right). \tag{55}$$

Thus:

$$\left\| P_i^t g_i \right\|^2 \geq \langle K g_i, g_i\rangle^2 = \langle K^S g_i, g_i\rangle^2 \tag{56}$$

and, since K^S is symmetric positive definite, from (5) we conclude that

$$\left\| P_i^t g_i \right\|^2 \geq \left(\lambda_{\min}(K^S) \left\| g_i \right\|^2 \right)^2. \quad \square \tag{57}$$

Lemma 4.4. *For each $i \in \mathbb{N}$ we have*

$$\lambda_{\max}(W_i) \leq \lambda_{\max}(N) \left(\|K\| \frac{\|KN\|^s - 1}{\|KN\| - 1} \right)^2 \left\| g_i \right\|^2. \tag{58}$$

Proof. Let $v \in \mathbb{R}^s$ be the eigenvector of W_i associated to the eigenvalue $\lambda_{\max}(W_i)$ and so that $\|v\| = 1$, and thus:

$$\lambda_{\max}(W_i) = v^t W_i v = v^t P_i^t N P_i v. \tag{59}$$

On the other hand, from definition of P_i

$$P_i^t N P_i = \left(Q_i + \sum_{j=i-m}^{i-1} P_j B_i^j \right)^t N \left(Q_i + \sum_{j=i-m}^{i-1} P_j B_i^j \right) \tag{60}$$

using the orthogonality properties of the Lemma 4.1 we obtain that

$$P_i^t N P_i = Q_i^t N Q_i + \sum_{j=i-m}^{i-1} Q_i^t N P_j B_i^j + \sum_{j=i-m}^{i-1} (P_j B_i^j)^t N Q_i + \sum_{j=i-m}^{i-1} (P_j B_i^j)^t N P_j B_i^j. \tag{61}$$

From definition of B_i^j and because of the symmetry of N and W_j:

$$Q_i^t N P_j B_i^j = -Q_i^t N P_j W_j^{-1} P_j^t N Q_i, \tag{62}$$

$$(P_j B_i^j)^t N Q_i = -(W_j^{-1} P_j^t N Q_i)^t P_j^t N Q_i = -Q_i^t N P_j W_j^{-1} P_j^t N Q_i \tag{63}$$

and, since $P_j^t N P_j W_j^{-1} = W_j W_j^{-1} = I$

$$(P_j B_i^j)^t N P_j B_i^j = (W_j^{-1} P_j^t N Q_i)^t P_j^t N P_j (W_j^{-1} P_j^t N Q_i) = Q_i^t N P_j W_j^{-1} P_j^t N Q_i. \tag{64}$$

Then we can substitute (62)–(64) in (61) and obtain

$$P_i^t NP_i = Q_i^t NQ_i - \sum_{j=i-m}^{i-1} Q_i^t NP_j W_j^{-1} P_j^t NQ_i. \tag{65}$$

Since $Q_i^t NP_j W_j^{-1} P_j^t NQ_i$ are symmetric positive definite we can assure that

$$v^t P_i^t NP_i v \le v^t Q_i^t NQ_i v \le \lambda_{\max}(N) \left\| Q_i v \right\|^2. \tag{66}$$

On the other hand, if $(v_1, \ldots, v_s)$ are the coordinates of v respect to the canonic base of $\mathbb{R}^s$,

$$\left\| Q_i v \right\| = \left\| v_1 Kg_i + v_2 (KN) Kg_i + \cdots + v_s (KN)^{s-1} Kg_i \right\|. \tag{67}$$

Using the triangular inequality and the fact that $\|v\| = 1$, we have

$$\left\| v_1 Kg_i + \cdots + v_s (KN)^{s-1} g_i \right\| \le \left(\|K\| + \cdots + \left\| (KN)^{s-1} K \right\| \right) \|g_i\| \tag{68}$$

and

$$\|K\| + \cdots + \left\| (KN)^{s-1} K \right\| \le \|K\| \left(1 + \|KN\| + \cdots + \left\| (KN)^{s-1} \right\| \right) = \|K\| \frac{\|KN\|^s - 1}{\|KN\| - 1}. \tag{69}$$

We can then conclude that

$$\lambda_{\max}(W_i) = v^t P_i^t NP_i v \le \lambda_{\max}(N) \left(\|K\| \frac{\|KN\|^s - 1}{\|KN\| - 1} \right)^2 \|g_i\|^2. \quad \square \tag{70}$$

Finally we enunciate the following convergence theorem for the s-step Orthomin(m) method:

Theorem 4.1. *Under the hypotheses of previous lemmas, we have*

$$\lim_{i \to \infty} r_i = 0. \tag{71}$$

Proof. Since Lemma 4.2

$$E_{i+1} = E_i - (P_i^t g_i)^t W_i^{-1} P_i^t g_i. \tag{72}$$

Sequence E_i is nonnegative and monotonously decreasing. Then there are two possibilities:

(a) There exists $i \in \mathbb{N}$ so that $E_i = E_{i+1}$. Then $(P_i^t g_i)^t W_i^{-1} P_i^t g_i = 0$, which implies that $P_i^t g_i = 0$ since W_i^{-1} is positive definite. Then, from (b) of Lemma 4.1 and definition of P_i, $Q_i^t g_i = 0$. In particular $\langle Kg_i, g_i \rangle = 0$, then $g_i = 0$ and the method converges.

(b) For all $i \in \mathbb{N}$ it verifies that $E_i < E_{i+1}$. Then E_i converges since it is strictly decreasing and $E_i \ge 0$. Thus:

$$\lim_{i \to \infty} (E_i - E_{i+1}) = 0 = \lim_{i \to \infty} (P_i^t g_i)^t W_i^{-1} P_i^t g_i. \tag{73}$$

In this way, for each $\varepsilon > 0$, there exists $k \in \mathbb{N}$ so that $(P_k^t g_k)^t W_k^{-1} P_k^t g_k < \varepsilon$. On the other hand

$$(P_k^t g_k)^t W_k^{-1} P_k^t g_k \ge \lambda_{\min}(W_k^{-1}) \left\| P_k^t g_k \right\|^2 = \frac{1}{\lambda_{\max}(W_k)} \left\| P_k^t g_k \right\|^2 \tag{74}$$

and applying previous Lemmas 4.3 and 4.4, we obtain that

$$\frac{1}{\lambda_{\max}(W_k)} \left\| P_k^t g_k \right\|^2 \ge \frac{\lambda_{\min}(K^S)^2 \left(\|KN\| - 1 \right)^2}{\lambda_{\max}(N) \|K\|^2 \left(\|KN\|^s - 1 \right)^2} \|g_k\|^2. \tag{75}$$

We have proved that, for each $\varepsilon > 0$, there exists $k \in \mathbb{N}$ so that

$$\|g_k\|^2 < \varepsilon \frac{\lambda_{\max}(N) \|K\|^2 \left(\|KN\|^s - 1 \right)^2}{\lambda_{\min}(K^S)^2 \left(\|KN\| - 1 \right)^2} \tag{76}$$

which implies that $\lim_{i \to \infty} g_i = 0$, and then $\lim_{i \to \infty} r_i = 0$. $\square$

5. Particular cases of the *s*-GOA

From particular choices of matrices H and K, first we shall proceed to obtain the known s-step methods (Preconditioned s-Conjugate Gradient, s-Generalized Conjugate Residual and s-Normal Equation) and then we shall propose new s-step variants.

5.1. Preconditioned s-Conjugate Gradient Algorithm

Suppose that matrix A is symmetric positive definite. Let $H = A^{-1}$ and K any symmetric positive definite matrix. Then $N = A$ and the s-GOA becomes the s-step variant of the Preconditioned Conjugate Gradient Algorithm proposed in [7]. In the particular case of $K = I$ we have the s-step variant of the Conjugate Gradient Method [5].

From Lemmas 3.1 and 3.2 we obtain the following orthogonality properties:

(i) $P_i^t A P_j = 0$ for all $i \neq j$.
(ii) $P_j^t r_i = 0$ for all $i > j$. In particular $\langle r_i, Kr_j\rangle = 0$ for all $i > j$.
(iii) r_{i+1} minimizes $E(r) = \langle r, A^{-1}r\rangle = \langle c - x, A(c - x)\rangle$ over $x_0 + \pounds\{P_0, \ldots, P_i\}$.

5.2. s-Generalized Conjugate Residual Algorithm

Now choose $H = I$ and $K = A^{-1}$. Then $N = A^2$ and we obtain the s-step variant of the Generalized Conjugate Residual Algorithm proposed in [6]. If matrix A is symmetric positive definite, then K is symmetric and this method is the s-step variant of the Conjugate Residual Algorithm [5].

The properties for these methods from Lemmas 3.1 and 3.2 are:

(i) $(AP_i)^t (AP_j) = 0$ for all $i \neq j$.
(ii) $AP_j^t r_i = 0$ for all $i > j$. In particular $\langle r_i, Ar_j\rangle = 0$ for all $i > j$.
(iii) r_{i+1} minimizes $E(r) = \langle r, r\rangle$ over $x_0 + \pounds\{P_0, \ldots, P_i\}$.

Since matrix K is not symmetric in general we can consider the Orthomin(m) method for this algorithm. The Orthomin(0) is the s-step variant of the Minimal Residual Algorithm proposed in [6], and the Orthomin(1) is the s-step variant of the well known Axelsson's Minimal Residual [16]. In these cases it holds, from (c) in Lemma 4.1, that r_{i+1} minimizes $E(r) = \langle r, r\rangle$ over $x_i + P_i$.

5.3. s-Normal Equation algorithm

Now A is a nonsingular matrix, $H = I$ and $K = I$. Then $N = A^t A$ and the resulting algorithm is the s-step variant of the Normal Equation which appears in [17].

The properties from Lemmas 3.1 and 3.2 are:

(i) $(AP_i)^t (AP_j) = 0$ for all $i \neq j$.
(ii) $AP_j^t r_i = 0$ for all $i > j$. In particular $\langle Ar_i, Ar_j\rangle = 0$ for all $i > j$.
(iii) r_{i+1} minimizes $E(r) = \langle r, r\rangle$ over $x_0 + \pounds\{P_0, \ldots, P_i\}$.

5.4. s-Minimal Error Algorithm

Let A be a nonsingular matrix, $H = (AA^t)^{-1}$ and $K = A^t A$. Then K is symmetric and $N = I$. In this case we have that $g_i = A^{-1} r_i$ and $z_i = P_i^t g_i$. Then vector z_i depends on A^{-1} whose calculation would render the algorithm useless in practice. To avoid computing g_i we introduce the following matrices:

- $R_i = \Delta_{AA^t}(r_i)$,
- $Q_0 = R_0$,
- $Q_i = R_i + Q_{i-1} B_{i-1}$ for $i > 0$.

It is obvious that $P_i = A^t Q_i$ and, since $P_i^t P_{i-1} = 0$, we deduce

$$Q_i^t (AA^t) Q_{i-1} = 0. \tag{77}$$

Thus we propose in this paper the s-step variant of the Minimal Error Algorithm.

Algorithm 5.1 (*s-Minimal Error*).
Let $x_0 \in \mathbb{R}^n$
$r_0 = b - Ax_0$
$Q_0 = \Delta_{AA^t}(r_0)$.
For $i = 0, 1, 2, \ldots$ until convergence **Do**

$$P_i = A^t Q_i$$
$$W_i = P_i^t P_i$$
$$z_i = Q_i^t r_i$$
$$y_i = W_i^{-1} z_i$$
$$x_{i+1} = x_i + P_i y_i$$
$$r_{i+1} = r_i - AP_i y_i$$
$$R_{i+1} = \Delta_{AA^t}(r_{i+1})$$
$$B_{i+1} = -W_i^{-1}(AP_i)^t R_{i+1}$$
$$Q_{i+1} = R_{i+1} + Q_i B_{i+1}$$

EndFor.

The orthogonalization properties from Lemmas 3.1 and 3.2 are:

(i) $(P_i)^t(P_j) = 0$ for all $i \neq j$.
(ii) $P_j^t A^{-1} r_i = 0$ for all $i > j$. In particular $\langle r_i, r_j \rangle = 0$ for all $i > j$.
(iii) r_{i+1} minimizes $E(r) = \langle A^{-1}r, A^{-1}r \rangle = \langle x - c, x - c \rangle$ over $x_0 + \mathcal{L}\{P_0, \ldots, P_i\}$.

5.5. s-Biconjugate Gradient

If A is a nonsingular matrix, the Biconjugate Gradient method, [18], generates two CG-like sequences of vectors, one based on a system with the original coefficient matrix A, and another one with A^t. In this subsection we propose an s-step variant of the Biconjugate Gradient method. First, we define the following matrices:

$$\mathbf{A} = \begin{pmatrix} 0 & A \\ A^t & 0 \end{pmatrix}, \qquad X = \begin{pmatrix} x^* \\ x \end{pmatrix}, \qquad B = \begin{pmatrix} b \\ b^* \end{pmatrix}. \tag{78}$$

Let

$$\mathbf{H} = (\mathbf{A}^{-1})^t = \begin{pmatrix} 0 & (A^t)^{-1} \\ A^{-1} & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{K} = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}, \tag{79}$$

then $\mathbf{N} = \mathbf{A}^t\mathbf{H}\mathbf{A} = \mathbf{A}$ is a symmetric matrix. Superscript $*$ denotes the array part which is associated to the sequence based on A^t. Now, the s-step variant of the Biconjugate Gradient method can be derived from the s-GOA method:

Algorithm 5.2 (*s-Biconjugate Gradient*).
Let $X_0 \in \mathbb{R}^{2n}$
$R_0 = B - \mathbf{A}X_0$
$\mathbf{P}_0 = \Delta_{\mathbf{KA}}(\mathbf{K}R_0)$
For $i = 0, 1, 2, \ldots$ until convergence **Do**

$$W_i = \mathbf{P}_i^t \mathbf{A} \mathbf{P}_i \tag{80}$$
$$y_i = W_i^{-1} \mathbf{P}_i^t R_i \tag{81}$$
$$X_{i+1} = X_i + \mathbf{P}_i y_i \tag{82}$$
$$R_{i+1} = R_i - \mathbf{A}\mathbf{P}_i y_i \tag{83}$$
$$\mathbf{Q}_{i+1} = \Delta_{\mathbf{KA}}(\mathbf{K}R_{i+1}) \tag{84}$$
$$B_{i+1} = -(W_i)^{-1} \mathbf{P}_i^t \mathbf{A} \mathbf{Q}_{i+1} \tag{85}$$
$$\mathbf{P}_{i+1} = \mathbf{Q}_{i+1} + \mathbf{P}_i B_{i+1} \tag{86}$$

EndFor.

Next we will write the s-Biconjugate Gradient method in terms of n-dimensional vectors. For this purpose we denote by

$$\mathbf{P}_i = \begin{pmatrix} P_i^* \\ P_i \end{pmatrix}, \qquad \mathbf{Q}_i = \begin{pmatrix} Q_i^* \\ Q_i \end{pmatrix} \quad \text{and} \quad R_i = \begin{pmatrix} r_i \\ r_i^* \end{pmatrix} \tag{87}$$

and enunciate the following lemma:

Lemma 5.1. *In the s-Biconjugate Gradient method, and for all $k \in \{0, 1, 2, \ldots\}$, it holds that:*

(a) $\left(Q_i^*\right)^t A^k r_i = Q_i^t \left(A^t\right)^k r_i^*$.
(b) $\left(Q_i^*\right)^t A^k Q_i = Q_i^t \left(A^t\right)^k Q_i^*$.

Proof. It is obvious from the fact that, for all $k \in \{0, 1, 2, \ldots\}$,

$$r_i^t \left(A^t\right)^k r_i^* = \left(r_i^*\right)^t A^k r_i. \quad \square \tag{88}$$

Now, the following lemma can be enunciated as a consequence of Lemmas 3.1 and 5.1:

Lemma 5.2. *In the s-Biconjugate Gradient method, and for all $k \in \{0, 1, 2, \ldots\}$, it holds that:*

(a) $(P_i^*)^t A^k P_i = P_i^t \left(A^t\right)^k P_i^*$.
(b) $(P_i^*)^t A^k r_{i+1} = P_i^t \left(A^t\right)^k r_{i+1}^*$.
(c) $(P_i^*)^t r_{i+1} = P_i^t r_{i+1}^* = 0$.
(d) $(P_i^*)^t r_i = P_i^t r_i^* = (Q_i^*)^t r_i = Q_i^t r_i^*$.

Proof. We will prove statements (a) and (b) by induction on i. For $i = 0$ statements (a) and (b) are true from Lemma 5.1 and since $R_1 = R_0 - \mathbf{A}\mathbf{P}_0 y_0$. Suppose that (a) and (b) are true for $i - 1$ with $i \geq 1$, then, using (86),

$$(P_i^*)^t A^k P_i = (Q_i^*)^t A^k Q_i + (Q_i^*)^t A^k P_{i-1} B_{i-1} + B_{i-1}^t (P_{i-1}^*)^t A^k Q_i + B_{i-1}^t (P_{i-1}^*)^t A^k P_{i-1} B_{i-1} \tag{89}$$

and

$$P_i^t \left(A^t\right)^k P_i^* = Q_i^t (A^t)^k Q_i^* + Q_i^t (A^t)^k P_{i-1}^* B_{i-1} + B_{i-1}^t P_{i-1}^t (A^t)^k Q_i^* + B_{i-1}^t P_{i-1}^t (A^t)^k P_{i-1}^* B_{i-1}. \tag{90}$$

The first summands of the second right hand side of (89) and (90) are equal as a consequence of Lemma 5.1. So are the other corresponding summands by the induction hypothesis on (a) and (b), which proves (a). On the other hand, using Eq. (83),

$$(P_i^*)^t A^k r_{i+1} = (P_i^*)^t A^k r_i - (P_i^*)^t A^{k+1} P_i y_i \tag{91}$$

and

$$P_i^t \left(A^t\right)^k r_{i+1}^* = P_i^t \left(A^t\right)^k r_i^* - P_i^t \left(A^t\right)^{k+1} (P_i)^* y_i. \tag{92}$$

Then equality (b) is derived from (a), Lemma 5.1 and induction hypothesis since $\mathbf{P}_i = \mathbf{Q}_i + \mathbf{P}_{i-1} B_i$.

Section (c) follows from statement (b) of Lemma 3.1, and statement (d) from Eq. (86) and previous (c). $\square$

Now, by the previous Lemma 5.2, we can write

$$W_i = (P_i^*)^t A P_i + P_i^t A^t P_i^* = 2(P_i^*)^t A P_i \tag{93}$$

and (81) can be written as

$$W_i y_i = 2(P_i^*)^t r_i. \tag{94}$$

As consequence of section (b) of Lemma 5.2, we have that

$$(P_i^*)^t A Q_{i+1} = P_i^t A^t Q_{i+1}^*, \tag{95}$$

and then from (85) we get

$$B_{i+1} = -(W_i)^{-1} \left(P_i^t A^t Q_{i+1}^* + (P_i^*)^t A Q_{i+1}\right) = -2(W_i)^{-1} \left(P_i^t A^t Q_{i+1}^*\right). \tag{96}$$

Finally, we can write again the s-Biconjugate Gradient method in the following way:

Table 1
s-Step methods.

Algorithm	Convergence condition	H	K
s-Conjugate Gradient	A s. p. d.	A^{-1}	I
s-Preconditioned Conjugate Gradient	A s. p. d.	A^{-1}	Any
s-Conjugate Residual	A s. p. d.	I	A^{-1}
s-Normal Equation	A nonsingular	I	I
s-Minimal Error	A nonsingular	$(AA^t)^{-1}$	A^tA
s-Generalized Conjugate Residual	A^S s. p. d.	I	$(A^{-1})^t$
s-Minimal Residual	A^S s. d. p.	I	$(A^{-1})^t$
s-Biconjugate Gradient	–	$(\mathbf{A}^{-1})^t$	$\mathbf{K}$

Algorithm 5.3 (*s-Biconjugate Gradient*)**.**
Let $x_0, x_0^* \in \mathbb{R}^n$
$r_0 = b - Ax_0$
$r_0^* = b^* - A^t x_0^*$
$P_0 = \Delta_A(r_0)$
$P_0^* = \Delta_{A^t}(r_0^*)$
For $i = 0, 1, 2, \ldots$ until convergence **Do**

$$W_i = (P_i^*)^t A P_i$$
$$y_i = W_i^{-1}(P_i^*)^t r_i$$
$$x_{i+1} = x_i + P_i y_i$$
$$r_{i+1} = r_i - AP_i y_i$$
$$r_{i+1}^* = r_i^* - A^t P_i^* y_i$$
$$Q_{i+1} = \Delta_A(r_{i+1})$$
$$Q_{i+1}^* = \Delta_{A^t}(r_{i+1}^*)$$
$$B_{i+1} = -(W_i)^{-1}(P_i^*)^t A Q_{i+1}$$
$$P_{i+1} = Q_{i+1} + P_i B_{i+1}$$
$$P_{i+1}^* = Q_{i+1}^* + P_i^* B_{i+1}$$

EndFor.

Remark. Since matrices **N** and **K** are not positive definite in general, Theorem 3.1 cannot be used to assure the convergence of the *s*-Biconjugate Gradient. In practice, we expect convergence to occur in similar conditions to the usual Biconjugate Gradient method.

5.6. Other s-step methods

Suppose that matrix A is symmetric positive definite. We can obtain two new *s*-step methods taking $H = A^{-1}$ and $K = A$, and then $N = A$, for the first, and $H = (AA^t)^{-1}$ and $K = A$, and then $N = I$, for the second. Matrix K is symmetric positive definite in both algorithms.

The first *s*-step method minimizes $E(r) = \langle r, A^{-1}r\rangle$ and its orthogonalization properties are:

(i) $(P_i)^t A(P_j) = 0$ for all $i \neq j$.
(ii) $P_j^t r_i = 0$ for all $i > j$. In particular $\langle r_i, Ar_j\rangle = 0$ for all $i > j$.

The second one minimizes $E(r) = \langle x - c, x - c\rangle$ and its orthogonalization properties are:

(i) $(P_i)^t(P_j) = 0$ for all $i \neq j$.
(ii) $P_j^t A^{-1} r_i = 0$ for all $i > j$. In particular $\langle r_i, r_j\rangle = 0$ for all $i > j$.

Table 1 summarizes the *s*-step methods obtained as particular cases of *s*-GOA and the convergence condition on the coefficient matrix A. To our knowledge, the two methods in italic are so far unpublished.

6. Numerical results

The performance gains of parallel implementations of the *s*-steps methods have been shown in some of the cited references, alongside the numerical results presented. We can find numerical examples of *s*-Conjugate Gradient in [5,7, 17], of the *s*-Preconditioned Conjugate Gradient in [7,17], and of the *s*-Conjugate Residual in [5,17]. Numerical results for the *s*-step variant of Orthomin(m) are obtained in [8,19]. In this section we present some numerical results for the two new

Table 2
Iterations s-Minimal Error.

	$s=1$	$s=2$	$s=4$	$s=6$	$s=8$
Iterations	15 526	8305	3897	2590	1570

Table 3
Iterations s-BiCG.

	$s=1$	$s=2$	$s=4$	$s=6$
Iterations	277	140	94	49

Table 4
Execution times in seconds s-Minimal Error.

	$s=1$	$s=2$	$s=4$	$s=6$	$s=8$
$Nth=1$	190.22	198.99	188.77	197.04	186.59
$Nth=2$	157.40	118.91	107.92	107.36	87.14
$Nth=4$	90.99	83.72	72.28	68.45	55.56
$Nth=8$	83.64	66.84	56.24	54.32	44.06
$Nth=16$	83.49	61.53	54.01	49.85	40.24

Table 5
Execution times in seconds s-BiCG.

	$s=1$	$s=2$	$s=4$	$s=6$
$Nth=1$	3.23	2.87	2.71	2.72
$Nth=2$	1.42	1.22	1.18	1.08
$Nth=4$	0.74	0.65	0.64	0.53
$Nth=8$	0.38	0.35	0.34	0.31
$Nth=16$	0.41	0.41	0.37	0.28

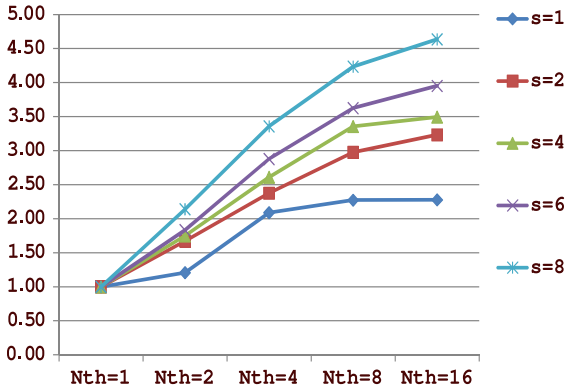

Fig. 1. Speedup for s-Minimal Error.

methods proposed in this work, the s-Minimal Error and the s-Biconjugate Gradient. These examples were executed on the Finis Terrae at the *Centro de Supercomputación de Galicia* (CESGA). This supercomputer is an integrated system with shared memory nodes with a NUMA SMP architecture. It is composed of 142 HP Integrity rx7640 nodes with 16 Itanium Montvale cores. For the parallel implementation we use OpenMp and execute 1, 2, 4, 8 and 16 cores (Nth) with shared memory. We have used matrices from the Matrix Market Collection [20]. The s-Minimal Error was executed over the matrix fidapm11, non symmetric matrix of order 22 294, and the s-BiCG over fidapm37, non symmetric matrix of order 9152. The termination criterion used was $\frac{\|r_i\|}{\|r_0\|} < \epsilon$, where $\epsilon = 5 \cdot 10^{-3}$ for s-Minimal Error and $\epsilon = 5 \cdot 10^{-4}$ for s-BiCG.

First we can see, in Tables 2 and 3, how the number of iterations decreases when the value of s increases. In the s-Minimal Error s take values of 1, 2, 4, 6 and 8 and in the s-BiCG 1, 2, 4 and 6. Methods with $s = 1$ are equivalent to the originals. We have found stability problems in s-BiCG with values of s greater than 6.

Execution times are shown in Tables 4 and 5.

We have measured a relative speedup with respect to the original method ($s = 1$) in sequential ($Nth = 1$). Figs. 1 and 3 show the speedup for both methods, and Figs. 2 and 4 show their relative speedup. It can be seen that the greater the value of s is, the higher the relative speedup is. They corroborate a major efficiency of these methods in parallel programming.

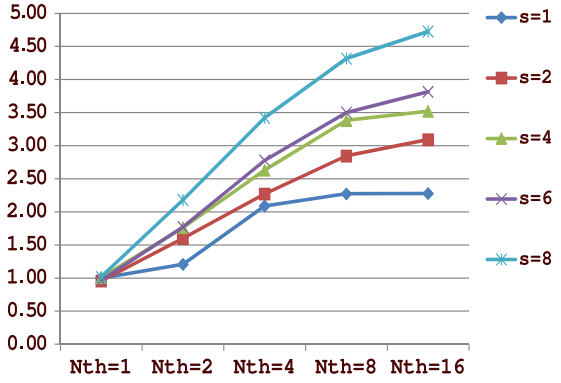

Fig. 2. Relative speedup for *s*-Minimal Error.

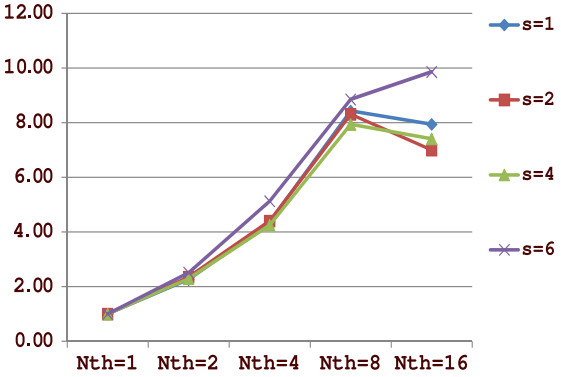

Fig. 3. Speedup for *s*-BiCG.

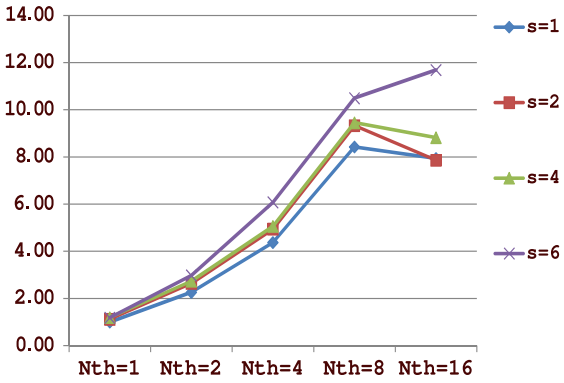

Fig. 4. Relative speedup for *s*-BiCG.

7. Conclusions and future work

In this work, an *s*-step variant of the General Orthogonalization Algorithm which generalizes Conjugate Gradient methods has been presented. The *s*-step variants of known iterative methods are derived as particular cases (some of which converging for every nonsingular matrix) and two are unpublished to our knowledge. It has been verified that the convergence of these methods is supported in their *s*-step variants. This was done by proving some prerequisite lemmas and convergence and error estimate theorems. An Orthomin variant together with a convergence theorem has also been described.

Using *s*-step variants, the transformation of *s* vector operations in one matrix operation has been achieved. Therefore, these methods make use of greater level BLAS operations and gain efficiency in parallel computers with optimized BLAS kernels. This is because the ratio between the number of operations performed and computer memory accesses increases and the number of communications between nodes is reduced in multiprocessors systems.

The authors prove that the number of iterations required for convergence in an *s*-step method is that of the original method divided by *s*. Broadly speaking, this translates into corresponding time improvements, provided numerical instability does not show up in the process. We have to bear in mind that with the advent of mainstream 64-bit workstations, instability will be less of an issue in solving large systems. Numerical examples of the *s*-step variant of the double orthogonal

series algorithm were dealt with in [15] and s-step variant of the GMRES in [14]. Also, we can see more numerical examples of other s-steps variants in the cited references. Throughout these papers, the benefits of s-step variant of the methods have been assessed as outweighing the costs, which justifies attempting to generalize them to a larger class. Finally, some numerical results are presented for the two new proposed methods. These results show that the parallel implementation of s-step methods have better performance than the original ones ($s = 1$).

An exhaustive numerical and stability analysis of those and other methods in Table 1 is at present under study by the authors.

Acknowledgments

This work was partially supported by Xunta de Galicia (Project PGIDIT06RMA23501PR-2) and Spain Government (Project TIN2010-17541). The authors would like to thank the *Centro de Supercomputación de Galicia* (CESGA) for its support and for putting its advanced computing facilities at our disposal.

References

[1] G. Ballard, J. Demmel, O. Holtz, O. Schwartz, Minimizing communication in numerical linear algebra, Technical Report No. UCB/EECS-2011-15, EECS Department University of California, Berkeley, 2011.
[2] J. Demmel, M. Hoemmen, M. Mohiyuddin, K. Yelick, Communication-optimal iterative methods, Journal of Physics Conference Series 180 (2009) SciDAC 2009. Article Number: UNSP 012040.
[3] C. Lawson, R. Hanson, D. Kincaid, F. Krogh, Basic linear algebra subprograms for FORTRAN usage, ACM Transactions on Mathematical Software 5 (1986) 308–325.
[4] PBLAS home page. http://www.netlib.org/scalapack/pblas_qref.html.
[5] A.T. Chronopoulos, C.W. Gear, s-step iterative methods for symmetric linear systems, Journal of Computational and Applied Mathematics 25 (1989) 153–168.
[6] A.T. Chronopoulos, s-step iterative methods for (non)symmetric (in)definite linear systems, SIAM Journal on Numerical Analysis 28 (6) (1991) 1776–1789.
[7] A.T. Chronopoulos, C.W. Gear, On the efficient implementation of preconditioned s-step conjugate gradient methods on multiprocessors with memory hierarchy, Parallel Computing 11 (1989) 37–53.
[8] A.T. Chronopoulos, C.D. Swanson, Parallel iterative s-step methods for unsymmetric linear systems, Parallel Computing 22 (1996) 623–641.
[9] A.T. Chronopoulos, A. Kucherov, Block S-step Krylov iterative methods, Numerical Linear Algebra with Applications 17 (1) (2010) 3–15.
[10] P. Joly, Présentation de synthèse des méthodes de gradient conjugué, RAIRO. Modélisation Mathématique et Analyse Numérique 20 (4) (1986) 639–665.
[11] P. Joly, M. Vidrascu, Quelques méthodes classiques de résolution de systèmes linéaires: exemples d'utilisation dans la biliothèque MODULEF, INRIA, 1994.
[12] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, 1996.
[13] P. González, J.C. Cabaleiro, T.F. Pena, Iterative solution of large linear systems with non-smooth submatrices using partial wavelet transforms and split-matrix matrix-vector multiplication, International Journal for Numerical Methods in Engineering 59 (2004) 457–473.
[14] Guangye Li, A block variant of the GMRES method on massively parallel processors, Parallel Computing 23 (1997) 1005–1019.
[15] J.A. Alvarez-Dios, J.C. Cabaleiro, G. Casal, A s-step variant of the double orthogonal series algorithm, in: Numerical Mathematics and Advanced Applications, ENUMATH 2005, Springer-Verlag, 2006, pp. 937–944.
[16] O. Axelsson, Conjugate gradient type methods for unsymmetric and inconsistent systems of equations, Linear Algebra and its Applications 29 (1980) 1–16.
[17] A.T. Chronopoulos, A class of parallel iterative methods implemented on multiprocessors, Ph.D. Thesis, University of Illinois at Urbana-Champaing, 1987.
[18] R. Fletcher, Conjugate gradient methods for indefinite systems, in: Lecture Notes in Mathematics, vol. 506, Springer-Verlag, 1976, pp. 73–89.
[19] F. Toutounian, The stable A^tA-orthogonal s-step Orthomin(k) algorithm with the CADNA library, Numerical Algorithms 17 (1–2) (1986) 105–119.
[20] Matrix market home page. http://math.nist.gov/MatrixMarket/.