JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 132, No. 2, pp. 267–285, February 2007 (© 2007)
DOI: 10.1007/s10957-006-9119-3

Lanczos Conjugate-Gradient Method and Pseudoinverse Computation on Indefinite and Singular Systems[1,2]

G. FASANO[3]

Communicated by L. C. W. Dixon

Published Online: 19 January 2007

Abstract. This paper extends some theoretical properties of the conjugate gradient-type method FLR (Ref. 1) for iteratively solving indefinite linear systems of equations. The latter algorithm is a generalization of the conjugate gradient method by Hestenes and Stiefel (CG, Ref. 2). We develop a complete relationship between the FLR algorithm and the Lanczos process, in the case of indefinite and possibly singular matrices. Then, we develop simple theoretical results for the FLR algorithm in order to construct an approximation of the Moore-Penrose pseudoinverse of an indefinite matrix. Our approach supplies the theoretical framework for applications within unconstrained optimization.

Key Words. Unconstrained optimization, Krylov subspace methods, planar conjugate gradient, Moore-Penrose pseudoinverse.

1. Introduction

In this paper, we study iterative methods to provide a solution of a dense linear system

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

where the symmetric matrix $A \in \mathbb{R}^{n\times n}$ is indefinite and possibly singular, $b \in \mathbb{R}^n$, and n is large. Many real large-scale problems require the solution of linear

[1] This work was partially supported by the MIUR, FIRB Research Program on Large-Scale Nonlinear Optimization and by the Ministero delle Infrastrutture e dei Trasporti in the framework of the Research Program on Safety.

[2] The author thanks Stefano Lucidi and Massimo Roma for fruitful discussions plus the Associate Editor for effective comments.

[3] Postdoctoral Fellow, Istituto di Analisi dei Sistemi ed Informatica A. Ruberti, CNR, Rome, Italy, and Istituto Nazionale per Studi ed Esperienze di Architettura Navale (INSEAN), Rome, Italy.

system (1) and they often need the use of efficient solvers, along with easy software packages. Many iterative algorithms for solving the linear system (1) provide us with useful and efficient tools (Ref. 3); nevertheless, the selection of the appropriate method is often a difficult problem for nonspecialists.

When Krylov subspace methods are considered (Refs. 4, 5) and good preconditioners are adopted, the differences among methods become less relevant (Ref. 6). However, this trivially shifts the problem to the identification of a suitable general purpose preconditioner.

When problem (1) becomes ill-conditioned, the numerical treatment is more complicated and some regularization techniques, which use additional information for stabilizing the solution, are often advisable (Ref. 6). Moreover, optimization frameworks provide strong motivations for investigating the solution of the possibly singular system (1).

In particular, consider the solution of the nonlinear least-squares problem

$$\min_{x\in\mathbb{R}^n} (1/2)\,\|r(x)\|^2, \qquad r:\mathbb{R}^n \to \mathbb{R}^m, \tag{2}$$

by means of the damped Gauss-Newton method (Ref. 7). Let $J(x) \in \mathbb{R}^{m\times n}$ be the Jacobian of the vector function $r(x)$ at the current point x. Then, at step k, the latter method considers the linear approximation $r(x_k) + J(x_k)d_k$ of $r(x)$ at x_k and computes d_k as a solution of the unconstrained subproblem

$$\min_{d\in\mathbb{R}^n} \|r(x_k) + J(x_k)d\|. \tag{3}$$

Then, the next iterate is

$$x_{k+1} = x_k + \alpha_k d_k,$$

where the steplength $\alpha_k \in \mathbb{R}$ is selected by a line search procedure (Ref. 8). Let $J^+(x_k)$ be the Moore-Penrose pseudoinverse of the matrix $J(x_k)$ (Ref. 9): the choice

$$d_k = -J^+(x_k)r(x_k)$$

among the solutions of (3) has two remarkable advantages. It is invariant under a linear transformation on x and it is a descent direction for the objective function in (2) (Ref. 7). In particular, the latter property is used in Refs. 10, 11, where the CG method is adopted to compute d_k, i.e. for equivalently giving a solution of the linear system

$$J^T(x_k)J(x_k)d = -J^T(x_k)r(x_k);$$

see also (Ref. 12). Observe that, in general, the matrix $J^T(x_k)J(x_k)$ is rank deficient.

The Newton method for eigenvector computation is another application within nonconvex optimization, where a solution of the possibly singular system (1) is needed. Suppose that $\lambda \in \mathbb{R}$ is an approximate eigenvalue of the symmetric indefinite matrix $H \in \mathbb{R}^{n\times n}$ associated to the eigenvector $\upsilon \in \mathbb{R}^n$. Then, a nontrivial solution x^* of the linear system $(H - \lambda I)x = 0$ yields an approximation to the vector υ. The Newton method is often the method of choice for this purpose and gives the iterate (Ref. 13)

$$x_{k+1} = x_k - (H - \lambda I)^{-1} r_k, \tag{4}$$

where $r_k = (H - \lambda I)x_k,\ x_0 \in \mathbb{R}^n$. Since (4) is not well defined, it is turned into an iteration (Ref. 3)

$$x_{k+1} = x_k - (H - \lambda I)^{+} r_k, \tag{5}$$

by introducing the Moore-Penrose pseudoinverse of $H - \lambda I$. Under suitable assumptions, (5) is convergent to an approximation x^* of the eigenvector υ. Observe that the pseudoinverse $(H - \lambda I)^+$ is also an inner inverse, i.e.,

$$(H - \lambda I)(H - \lambda I)^+(H - \lambda I) = (H - \lambda I)$$

and that

$$r_k = (H - \lambda I)x_k.$$

Therefore, on large-scale problems, the iteration (5) may be solved as the equation

$$(H - \lambda I)(x_{k+1} - x_k) = -r_k, \tag{6}$$

and a Krylov-based method may be adopted. Unfortunately, since the matrix H is indefinite, the CG method may fail. We consider in this paper a generalized CG method, and we prove that, under suitable assumptions, it provides the pseudoinverse solution of equation (6). An iteration similar to (5) is introduced when the Jacobi-Davidson method (Ref. 14) is used, in place of the Newton method, for computing the eigenvector υ.

The above examples, along with the low computational cost and the low memory requirements of CG-like methods, induced us to study and consider the FLR algorithm in Ref. 1 as a possible candidate for solving (1).

We prove also the complete theoretical relationship between the FLR algorithm and the Lanczos process. Equivalently, under few assumptions, the FLR algorithm is proved to generate in exact arithmetic the sequence of Lanczos vectors.

In the following sections, we use the symbol $\|\cdot\|$ to denote the Euclidean norm for both a real n-dimensional vector and a real $n \times n$ matrix. We use the notation $x^T y$ for the inner product between the vectors $x, y \in \mathbb{R}^n$, so that $x \perp y$ is equivalent to $x^T y = 0$. $0_{[m,n]}$ is the $m \times n$ matrix with all entries equal to zero. With $R(A)$ and $N(A)$, we denote the range and the null space of the symmetric matrix $A \in \mathbb{R}^{m\times n}$. With $\mathcal{K}_i(\upsilon, A)$, we indicate the Krylov subspace $\mathrm{span}\{\upsilon,\ A\upsilon, \ldots, A^{i-1}\upsilon\}$

associated with the vector $v \in \mathbb{R}^n$ and the matrix $A \in \mathbb{R}^{m\times n}$. $\Pr_W(\nu)$indicates the projection of vector ν onto the linear vector space W. Finally,

$$\lambda_m = \min_j \left|\lambda_j(A)\right|, \quad \lambda_M = \max_j \left|\lambda_j(A)\right|,$$

where $\lambda_j(A),\ j \geq 1$, are the smallest and largest eigenvalues of the symmetric matrix A.

The paper is organized as follows. Section 2 deals with the description of a few general preliminaries. Sections 3 and 3.1 provide some relevant features of the FLR algorithm (Ref. 1), when used for solving (1) and the coefficient matrix A is indefinite and possibly singular. Here, under mild assumptions, the latter algorithm is used to construct an approximation of the Moore-Penrose pseudoinverse A^+. Sections 4 and 5 provide a noteworthy relation between the FLR algorithm and the Lanczos process. Finally, Section 6 contains conclusions and perspectives related to the treated subject.

2. Some General Results

In this section, we introduce a few general results for the solution of (1) which will be used in the sequel. Consider the CG-based algorithm FLR described in Ref. 1 (see Table 1). The latter algorithm is a general planar method (Refs. 15–19) for solving (1), when the matrix A is indefinite; i.e., it avoids the possible pivot breakdown of the CG in the indefnite case, by introducing 2×2 pivot elements.

Table 1. FLR Algorithm for solving the linear system (1).

Step 1.	Set $k = 1, x_1 \in \mathbb{R}^n, r_1 = b - Ax_1$.
	If $r_1 = 0$, then stop. Else, set $p_1 = r_1$.
Step k.	Compute $d_k = p_k^T A p_k$; set $\epsilon_k > 0$.
	If $\|d_k\| \geq \epsilon_k \|\|p_k\|\|^2$, go to Step k_A.
	If $\|d_k\| < \epsilon_k \|\|p_k\|\|^2$, go to Step k_B.
Step k_A.	Set $a_k = r_k^T p_k / d_k,\ x_{k+1} = x_k + a_k p_k,\ r_{k+1} = r_k - a_k A p_k$.
	If $r_{k+1} = 0$, then stop. Else,
	set $b_k = -p_k^T A r_{k+1}/d_k$ and $p_{k+1} = r_{k+1} + b_k p_k$.
	Set $k = k + 1$ go to Step k.
Step k_B.	If $k = 1$, then set $q_k = Ap_k$.
	If $k > 1$ and the previous Step is $(k-1)_A$, then
	set $\beta_{k-1} = -(Ap_{k-1})^T Ap_k/d_{k-1}$ and $q_k = Ap_k + \beta_{k-1} p_{k-1}$.
	If $k > 1$ and the previous Step is $(k-2)_B$, then
	set $\hat\beta_{k-2} = -(Aq_{k-2})^T Ap_k$ and $q_k = Ap_k + \hat\beta_{k-2}(d_{k-2} q_{k-2} - \delta_{k-2} p_{k-2})/\Delta_{k-2}$.
	Compute $c_k = r_k^T p_k,\ \delta_k = p_k^T Aq_k,\ e_k = q_k^T Aq_k,\ \Delta_k = d_k e_k - \delta_k^2$ and
	$\hat c_k = (c_k e_k - \delta_k q_k^T r_k)/\Delta_k,\ \hat d_k = (d_k q_k^T r_k - \delta_k c_k)/\Delta_k$.
	Set $x_{k+2} = x_k + \hat c_k p_k + \hat d_k q_k,\ r_{k+2} = r_k - \hat c_k Ap_k - \hat d_k Aq_k$.
	If $r_{k+2} = 0$, then stop. Else,
	compute $\hat b_k = -q_k^T Ar_{k+2}$ and set $p_{k+2} = r_{k+2} + \hat b_k (d_k q_k - \delta_k p_k)/\Delta_k$.
	Set $k = k + 2$ go to Step k.

Table 2. The Lanczos process applied to the system (1).

Step 0.	$k = 0,\ v_0 = b \in \mathbb{R}^n$,
	$u_0 = 0,\ \delta_0 = \|\|b\|\|$.
Step k.	If $\delta_k = 0$, then stop. Else, set $u_{k+1} = v_k/\delta_k$.
	Set $k = k + 1,\ \gamma_k = u_k^T A u_k$,
	$v_k = (A - \gamma_k I)u_k - \delta_{k-1}u_{k-1}$
	$\delta_k = \|\|v_k\|\|$, go to Step k.

We are concerned with proposing some new properties of the FLR algorithm in the case where the matrix A in (1) is singular.

We remark that the Krylov based algorithm FLR is a generalization of the CG in the case where the matrix A is indefinite. Indeed from Table 1, as long as the quantity d_k at step k is relatively large, a CG step is performed at step k_A. On the contrary, whenever d_k is relatively small, the vector q_k is generated at step k_B, so that the solution of (1) is detected over the 2-dimensional manifold span $\{p_k, q_k\}$; see also Ref. 20. Furthermore, from Lemma 2.2 of Ref. 1, if $r_k \neq 0$ and at Step k_A we have $d_k = 0$ (i.e., a pivot breakdown occurs), the step k_B cannot fail by a possible division by zero (i.e., $\Delta_k \neq 0$) and the FLR algorithm does not stop.

First, we intend to determine the properties of the FLR algorithm when the matrix A is singular. Then, we study the relationship between the sets of orthogonal directions generated by the Lanczos process and the FLR algorithm, when solving (1).

Consider algorithm FLR and the Lanczos process (Table 2), where without loss of generality we assumed $v_0 = b$ at Step 0 (see Ref. 21 for a more general choice). Recalling the symmetry of the matrix A, let either the first nonzero Lanczos vector u_1 or the first residual r_1 in the FLR algorithm be given by

$$u_1 = y + z, \quad y = \Pr_{R(A)}(u_1), \quad z = \Pr_{N(A)}(u_1), \tag{7a}$$

$$r_1 = y + z, \quad y = \Pr_{R(A)}(r_1), \quad z = \Pr_{N(A)}(r_1). \tag{7b}$$

Then, the following general result holds (see also Refs. 22, 23).

Lemma 2.1. Given the symmetric matrix $A \in \mathbb{R}^{m\times n}$, let $P_i(\bullet)$ be a nonzero real polynomial of finite degree $i \geq 1$. Let $\lambda_1, \ldots \lambda_k$, $k \leq n$, be all the nonzero eigenvalues of the matrix A. Assume that only the nonzero eigenvalues $\lambda_{j_1}, \ldots \lambda_{j_{\hat{k}}}$ are distinct, with $j_h \in \{1, \ldots, k\}$, $h = 1, \ldots, \hat{k}$. For the hth distinct eigenvalue λ_{j_h}, consider the corresponding eigenspace E_{j_h}, i.e., the subspace E_{j_h} which spans the eigenvectors associated with λ_{j_h}.

(i) If the vector $y \in R(A)$ has a nonzero orthogonal projection on only $\hat{l} \leq \hat{k}$ eigenspaces among $E_{j_1}, \ldots, E_{j_{\hat{k}}}$, then we have $P_i(A)y = 0$ only if $i \geq \hat{l}$.

(ii) The sequence $\{P_i(A)y\}$, which is dependent on the index i, contains at most $\hat{l}$ linearly independent vectors.

Proof.

(i) Assume without loss of generality that

$$\Pr_{E_{j_h}}(y) \neq 0, \quad h = 1, \dots \hat{l}.$$

Then, $\hat{l}$ orthonormal eigenvectors $y_{j_1} \in E_{j_1}, \dots, y_{j_{\hat{l}}} \in E_{j_{\hat{l}}}$ and the vector $c \in \mathbb{R}^{\hat{l}}$ exist such that

$$y = \sum_{h=1}^{\hat{l}} c_{j_h} y_{j_h}, \quad c_{j_h} \neq 0. \tag{8}$$

From the symmetry of the matrix A, an orthogonal matrix $V \in \mathbb{R}^{n\times n}$ exists such that

$$A = VDV^T, \quad D = \operatorname{diag}\{\lambda_1, \dots \lambda_k, 0_{[n-k]}\}, \quad V = [y_1 \cdots y_k z_1 \cdots z_{n-k}], \tag{9}$$

where $y_1, \dots, y_k$ are orthonormal eigenvectors associated with the eigenvalues $\lambda_1, \dots, \lambda_k$ and

$$y_{j_h} \in \{y_1, \dots y_k\}, \quad \text{for any } j_h \in \{j_1, \dots j_{\hat{l}}\}. \tag{10}$$

Moreover, $z_1, \dots, z_{n-k}$ are orthonormal eigenvectors associated with the zero eigenvalue. Thus, for any i,

$$P_i(A) = VP_i(D)V^T; \tag{11}$$

consequently $P_i(A)y$ is given by

$$\begin{aligned} P_i(A)y &= V \begin{pmatrix} P_i(\lambda_1) & & & \\ & \ddots & & \\ & & P_i(\lambda_k) & \\ & & & P_i\left(0_{[n-k,n-k]}\right) \end{pmatrix} \sum_{h=1}^{\hat{l}} c_{j_h} V^T y_{j_h}, \\ &= Vu, \end{aligned}$$

where $u \in \mathbb{R}^n$ and from (10), for $P = 1, \dots n$,

$$u_p = \begin{cases} c_p P_i(\lambda_p), & \text{if } p \in \{j_1, \dots j_{\hat{l}}\}, \\ 0, & \text{otherwise.} \end{cases}$$

Since $c_{j_h} \neq 0$, for any $j_h \in \{j_1, \dots, j_{\hat{l}}\}$, and since V is nonsingular, $P_i(A)y = 0$ if and only if $P_i(\lambda_{j_h}) = 0$, $j_h \in \{j_1, \dots, j_{\hat{l}}\}$. In particular, this implies that the $\hat{l}$ distinct eigenvalues $\lambda_{j_1}, \dots, \lambda_{j_{\hat{l}}}$ are roots of the polynomial $P_i(\lambda)$. Consequently, if $P_i(A)y = 0$, then $i \geq \hat{l}$.

(ii) Consider the relation (8). Observe that (λ_{j_h}, y_{j_h}) is an eigenpair of the matrix A and the eigenvectors $y_{j_1}, \ldots, y_{j_{\hat{l}}}$ are orthogonal. Therefore,

$$P_i(A)y \in \text{span}\{y_{j_1}, \ldots, y_{j_{\hat{l}}}\}, \quad \text{for any } i \geq 1.$$

This implies that the sequence $\{P_i(A)y\}$ contains at most $\hat{l}$ linearly independent vectors, regardless of the choice of index $i \geq 1$. □

Remark 2.1. Observe that, according with the definitions used in Ref. 21, the integer $\hat{l}$ of Lemma 2.1 is the grade of y with respect to matrix A, i.e., the lowest degree of the polynomial $P(A)$ such that $P(A)y = 0$. Therefore Lemma 2.1 states a relationship between the grade of y and the eigenpairs of matrix A. Furthermore, connections between the polynomial $P_{\hat{l}}(A)$ and the minimal polynomial of matrix A were highlighted in Ref. 12.

3. FLR algorithm for Singular Linear Systems

Here, we aim at extending the results in Refs. 12, 16 and in the previous section, when considering the FLR algorithm in Table 1 for solving (1), in the case of indefinite and possibly singular matrix A. When the indefinite matrix A is nonsingular and $r_k \neq 0$ in the FLR algorithm, at step k we have either $d_k \neq 0$ or $\Delta_k \neq 0$ (Ref. 1); i.e., we are ensured that either step k_A or step k_B, can be performed. In this section, we are concerned with recasting an analogous result under the hypothesis that the matrix A is singular. Observe that, at step k of the FLR algorithm, $d_k = 0$ implies

$$\Delta_k = \delta_k^2 = -||Ap_k||^4,$$

since

$$\delta_k = p_k^T A q_k = ||Ap_k||^2$$

from Theorem 2.1 of Ref. 1. Hence, if $d_k = 0$ and the matrix A is singular, then Δ_k is nonzero as long as

$$p_k \notin N(A), \quad k < n. \tag{12}$$

The following theorem yields some results in order to satisfy condition (12).

Theorem 3.1. Consider the linear system (1) and let the matrix A be indefinite and possibly singular. In the FLR algorithm Let $r_1 = y + z$, with $y = \text{Pr}_{R(A)}(r_1)$ and

$z = \mathrm{Pr}_{N(A)}(r_1)$. Suppose that the integer $\hat{l}$ and the vector y satisfy the hypothesis of Lemma 2.1. Then, the FLR algorithm generates the sequences

$$\begin{aligned} r_i &= P_{i-1}(A)y + z, && i \le \hat{l}, \\ p_i &= Q_{i-1}(A)y + m_{i-1}z, && i \le \hat{l}, \\ q_i &= R_i(A)y + n_i z, && i \le \hat{l} - 1, \end{aligned} \tag{13}$$

where $P_j(\bullet)$, $Q_j(\bullet)$, $R_j(\bullet)$, are real polynomials of degree j and m_j, n_j, $\in \mathbb{R}$. Moreover, the directions p_i and q_i satisfy the relations:

$$\begin{aligned} &p_i \notin N(A), \quad i \le \hat{l}, \\ &q_i \notin N(A), \quad i \le \hat{l} - 1. \end{aligned} \tag{14}$$

Proof. By complete induction, when $i = 1$, then $r_1 = p_1 = y + z$; if step 1_B is performed, $q_1 = Ay$, according with (13). Moreover, let us distinguish between two cases. On the one hand, assume

$$\begin{aligned} r_{i-1} &= P_{i-2}(A)y + z, \\ p_{i-1} &= Q_{i-2}(A)y + m_{i-2}z, \end{aligned}$$

and let the FLR algorithm perform step $(i-1)_A$. Then,

$$\begin{aligned} r_i &= P_{i-2}(A)y + z - a_{i-1}AQ_{i-2}(A)y \\ &= P_{i-1}(A)y + z, \\ p_i &= P_{i-1}(A)y + z + b_{i-1}[Q_{i-2}(A)y + m_{i-2}z] \\ &= Q_{i-1}(A)y + m_{i-1}z. \end{aligned}$$

Furthermore, if the subsequent step is step i_B, then we obtain

$$\begin{aligned} q_i &= Ap_i + \beta_{i-1}p_{i-1} = AQ_{i-1}(A)y + \beta_{i-1}[Q_{i-2}(A)y + m_{i-2}z] \\ &= R_i(A)y + n_i z, \end{aligned}$$

according to (13)

On the other hand, assuming

$$\begin{aligned} r_{i-2} &= P_{i-3}(A)y + z, \\ p_{i-2} &= Q_{i-3}(A)y + m_{i-3}z, \end{aligned}$$

let the FLR algorithm perform step $(i-2)_B$. Then,

$$\begin{aligned} r_i &= P_{i-3}(A)y + z + \hat{c}_{i-2}AQ_{i-3}(A)y - \hat{d}_{i-2}AR_{i-2}(A)y \\ &= P_{i-1}(A)y + z, \\ p_i &= P_{i-1}(A)y + z + (\hat{b}_{i-2}/\Delta_{i-2})[d_{i-2}(R_{i-2}(A)y + n_{i-2}z) \\ &\quad - \delta_{i-2}(Q_{i-3}(A)y + m_{i-3}z)] = Q_{i-1}(A)y + m_{i-1}z. \end{aligned}$$

Furthermore, if the subsequent step is step i_B, then we obtain

$$\begin{aligned} q_i &= A[Q_{i-1}(A)y + m_{i-1}z] + (\hat{\beta}_{i-2}/\Delta_{i-2})(d_{i-2}q_{i-2} - \delta_{i-2}p_{i-2}) \\ &= R_i(A)y + n_i z, \end{aligned}$$

according to (13). As regards (14), the hypotheses ensure that $p_i \in N(A)$ if and only if $Q_{i-1}(A)y = 0$. By Lemma 2.1, the latter equality cannot hold as long as $i \le \hat{l}$. Similarly, we have

$$q_i \in N(A), \quad \text{if and only if } R_i(A)y = 0;$$

hence, as long as $i \le \hat{l} - 1$, $q_i \notin N(A)$. □

Now consider the FLR algorithm in Table 1 and let the vectors $t_1, \ldots, t_k$, $k \le n$ be defined in the following way:

$$\text{if} |d_k| \ge \epsilon_k ||p_k||^2, \quad \text{then set} \quad \alpha_k = a_k \quad \text{and} \quad t_k = p_k, \tag{15a}$$

$$\text{if} |d_k| < \epsilon_k ||p_k||^2, \quad \text{then set} \begin{cases} \alpha_k = \hat{c}_k, & t_k = p_k, \\ \alpha_{k+1} = \hat{d}_k, & t_{k+1} = q_k. \end{cases} \tag{15b}$$

Proposition 3.1. Let the matrix A in (1) be indefinite and possibly singular; let $\hat{l}$ and r_1 satisfy (7) and the hypothesis of Lemma 2.1. Then, the FLR algorithm generates directions $t_1, \ldots, t_{\hat{l}}$, with $t_i \notin N(A)$, $i = 1, \ldots, \hat{l}$, and these vectors are linearly independent.

Proof. The result is straightforward from Ref. 1, Theorem 3.1 and Lemma 2.1. □

3.1. Algorithm FLR and Moore-Penrose Pseudoinverse. Consider Theorem 3.1 and suppose that the FLR algorithm has generated directions $t_1, \ldots, t_{\hat{l}}$ before stopping. We prove that, if $b \in R(A)$, i.e., $z = \mathrm{Pr}_{N(A)}(r_1) = 0$, the FLR algorithm can provide an approximation of the Moore-Penrose pseudoinverse A^+, where A is indefinite and possibly singular.

More specifically, from Lemma 2.1, we introduce the following linear subspace, dependent on both the matrix A and the vector r_1:

$$R^p(r_1, A) = \mathrm{span}\{y_{j_1}, \ldots, y_{j_{\hat{l}}}\} \subseteq R(A), \tag{16}$$

where $y_{j_h} \in E_{j_h}$, $h = 1, \ldots, \hat{l}$ are orthogonal eigenvectors and $E_{j_1}, \ldots, E_{j_{\hat{l}}}$ are the eigenspaces of the matrix A, respectively associated to the distinct nonzero eigenvalues $\lambda_{j_1}, \ldots, \lambda_{j_{\hat{l}}}$, with $\mathrm{Pr}_{E_{j_h}}(r_1) \ne 0$, $h = 1, \ldots, \hat{l}$. Now, since $r_1 = y + z$, from the relation (13) the FLR algorithm can give the solution $\tilde{x}$ of (1) provided that $z = 0$. Moreover, if $b \in R(A)$, i.e., $z = 0$, exactly $\hat{l}$ directions are generated by the FLR algorithm before converging to $\tilde{x}$. Indeed, Lemma 2.1 and

Theorem 3.1 ensure that the FLR algorithm generates exactly the independent directions $t_1, \ldots, t_{\hat{l}}$, since the last step performed is either step $(\hat{l}-1)_A$ or step $(\hat{l}-2)_B$. As a consequence, if $\tilde{x}$ is a solution of the linear system (1) detected by the FLR algorithm, by the definition of Moore-Penrose pseudoinverse (Ref. 9)

$$\begin{aligned} \Pr_{R(A)}(\tilde{x}) &= A^+ b \\ &= A^+(r_1 + Ax_1) \\ &= A^+ r_1 + \Pr_{R(A)}(x_1), \end{aligned} \tag{17}$$

where the matrix A is indefinite and possibly singular. Moreover, from (15),

$$\tilde{x} = x_1 + \sum_{i=1}^{\hat{l}} \alpha_i t_i, \tag{18}$$

and assuming $z = 0$, from (13) of Theorem 3.1,

$$\Pr_{R(A)}(\tilde{x}) = \Pr_{R(A)}(x_1) + \sum_{i=1}^{\hat{l}} \alpha_i t_i. \tag{19}$$

Finally, combining (17) and (19), and considering again the relation $z = 0$, along with the expression of the coefficients $\alpha_i,\ i = 1, \ldots, \hat{l}$, in (15), we have[4]

$$\begin{aligned} A^+ y &= \sum_{i=1}^{\hat{l}} \alpha_i t_i \\ &= \sum_{i \in S_1} a_i p_i + \sum_{i \in S_2} \left(\hat{c}_i p_i + \hat{d}_i q_i\right) \\ &= \sum_{i \in S_1} \left(p_i^T r_i / p_i^T A p_i\right) p_i \\ &\quad + \sum_{i \in S_2} (1/\Delta_i)[(e_i p_i - \delta_i q_i)^T r_i p_i + (d_i q_i - \delta_i p_i)^T r_i q_i]. \end{aligned} \tag{20}$$

Now, it can be readily proved that

$$p_i^T r_i = p_i^T r_1, \quad q_i^T r_i = q_i^T r_1;$$

[4] In the following relations, we have introduced the pair of disjoint sets S_1 and S_2: S_1 is the set of indices $h \leq \hat{l}$ for which the FLR algorithm performs step h_A, while S_2 is the set of indices $h \leq \hat{l}$ for which the FLR algorithm performs step h_B. Thus, for the cardinality of the sets S_1 and S_2 the relation $|S_1| + 2|S_2| = \hat{l}$ holds.

see also Ref. 1. Thus, recalling that $z = 0$ and $\Delta_i \neq 0$ in (20), Table 1 and (16) yield, for any $y \in R^p(r_1, A)$,

$$
\begin{aligned}
0 &= A^+ y - \sum_{i\in S_1} \left(p_i p_i^T / p_i^T A p_i\right) r_1 \\
&\quad - \sum_{i\in S_2} (1/\Delta_i)[p_i(e_i p_i - \delta_i q_i)^T + q_i(d_i q_i - \delta_i p_i)^T] r_1 \\
&= A^+ y - \left[\sum_{i\in S_1} \left(p_i p_i^T / p_i^T A p_i\right) + \sum_{i\in S_2} (1/\Delta_i)(p_i\ q_i) \begin{pmatrix} e_i & -\delta_i \\ -\delta_i & d_i \end{pmatrix} \begin{pmatrix} p_i^T \\ q_i^T \end{pmatrix}\right] y \\
&= \left[A^+ - \sum_{i\in S_1} \left(p_i p_i^T / p_i^T A p_i\right) - \sum_{i\in S_2} (p_i\ q_i) \begin{pmatrix} d_i & \delta_i \\ \delta_i & e_i \end{pmatrix}^{-1} \begin{pmatrix} p_i^T \\ q_i^T \end{pmatrix}\right] y \\
&= [A^+ - (t_1 \cdots t_{\hat{l}}) B_{\hat{l}}^{-1} (t_1 \cdots t_{\hat{l}})^T] y, \qquad (21)
\end{aligned}
$$

$$
B_{\hat{l}} x = \mathrm{diag}_{i\in S_1, j\in S_2} \left\{ d_i, \begin{pmatrix} d_j & \delta_j \\ \delta_j & e_j \end{pmatrix} \right\}. \qquad (22)
$$

Observe that, in (21), whenever the pairs (p_i, q_i), $i \in S_2$, are conjugate (i.e., $\delta_i = 0$, for any $i \in S_2$, so that the FLR algorithm reduces to CG), the matrix $B_{\hat{l}}$ in relation (22) is diagonal; see also Ref. 12.

In addition, let (λ_i, v_i), $i = 1, \ldots, n$, be the eigenpairs of the symmetric nonsingular matrix $C \in \mathbb{R}^{n\times n.}$ Then, the spectral form of C^{-1} is simply (Ref. 3)

$$
\begin{aligned}
C^{-1} &= \sum_{i=1}^{n} (1/\lambda_i) v_i v_i^T \\
&= (v_1 \cdots v_n) \Lambda^{-1} (v_1 \cdots v_n)^T, \qquad (23a) \\
\Lambda &= \mathrm{diag}_{1\le i\le n}\{\lambda_i\}, \qquad (23b)
\end{aligned}
$$

which can be generalized by (22) in the singular case. Indeed, from (22), we have

$$
A^+ y = (t_1 \cdots t_{\hat{l}}) B_{\hat{l}}^{-1} (t_1 \cdots t_{\hat{l}})^T y, \quad \forall y \in R^p(r_1, A), \qquad (24)
$$

so that, if $\hat{l} = n$, then

$$
R^p(r_1, A) \equiv \mathbb{R}^n, \quad A^+ = A^{-1},
$$

and (24) yields

$$
A^{-1} y = (t_1 \cdots t_n) B_n^{-1} (t_1 \cdots t_n)^T y. \qquad (25)
$$

Finally, recalling that, in (23), the orthogonal eigenvectors v_i, v_j also satisfy $v_i^T C^{-1} v_j = 0$, for any $i \neq j \le n$, we recognize that (25) is a generalization of the spectral form (23). Relation (22) gives an approximation of the pseudoinverse matrix A^+ on the linear subspace $R^p(r_1, A)$. Finally, we prove the following result, which holds in particular, also for the CG, in the positive semidefinite case.

Theorem 3.2. Let $b \notin R(A)$ and let the hypothesis of Theorem 3.1 hold. Then, the solution $\tilde{x} = x_1 + \Sigma_{i=1}^{\hat{l}} \alpha_i t_i$, calculated by the FLR algorithm when solving (1) is not a least square solution of (1).

Proof. Consider the relation (13) and let $b \notin R(A)$, i.e. $z \neq 0$. From Lemma 2.1 the FLR algorithm provides in exact arithmetic $t_{\hat{l}+1} \in N(A)$, after the generation of directions $t_1, \dots t_{\hat{l}}$. Now, by means of the substitutions

$$\bar{p}_i = Q_{i-1}(A)y \text{ and } \bar{q}_i = R_i(A)y$$

in relations (13), we obtain from Table 1

$$\begin{aligned}
\tilde{x} &= x_1 + \sum_{i=1}^{\hat{l}} \alpha_i t_i \\
&= x_1 + \left[\sum_{i \in S_1} \left(p_i p_i^T / p_i^T A p_i \right) + \sum_{i \in S_2} (p_i \; q_i) \begin{pmatrix} d_i & \delta_i \\ \delta_i & e_i \end{pmatrix}^{-1} \begin{pmatrix} p_i^T \\ q_i^T \end{pmatrix} \right] r_1 \\
&= x_1 + \sum_{i \in S_1} \left(1/\bar{p}_i^T A \bar{p}_i \right) [\bar{p}_i + m_{i-1} z][\bar{p}_i + m_{i-1} z]^T r_1 \\
&\quad + \sum_{i \in S_2} (\bar{p}_i + m_{i-1} z \quad \bar{q}_i + n_i z) \begin{pmatrix} d_i & \delta_i \\ \delta_i & e_i \end{pmatrix}^{-1} \begin{pmatrix} \bar{p}_i^T + m_{i-1} z^T \\ \bar{q}_i^T + n_i z^T \end{pmatrix} r_1,
\end{aligned}$$

and since

$$\bar{p}_i^T z = \bar{q}_i^T z = 0, \quad z^T r_1 = ||z||^2,$$

we obtain

$$\begin{aligned}
\tilde{x} = x_1 &+ \sum_{i \in S_1} \left[\left(\bar{p}_i \bar{p}_i^T / \bar{p}_i^T A \bar{p}_i \right) r_1 + ||z||^2 \left(m_{i-1} / \bar{p}_i^T A \bar{p}_i \right) \bar{p}_i \right] + \lambda_1 z \\
&+ \sum_{i \in S_2} \left[(\bar{p}_i \;\; \bar{q}_i) \begin{pmatrix} d_i & \delta_i \\ \delta_i & e_i \end{pmatrix}^{-1} \begin{pmatrix} \bar{p}_i^T \\ \bar{q}_i^T \end{pmatrix} r_1 \right. \\
&\left. \qquad + ||z^2|| (\bar{p}_i \;\; \bar{q}_i) \begin{pmatrix} d_i & \delta_i \\ \delta_i & e_i \end{pmatrix}^{-1} \begin{pmatrix} m_{i-1} \\ n_i \end{pmatrix} \right] + \lambda_2 z,
\end{aligned} \tag{26}$$

where

$$\begin{aligned}
\lambda_1 &= m_{i-1} \left(\bar{p}_i^T r_1 / \bar{p}_i^T A \bar{p}_i \right) + m_{i-1}^2 \left(||z||^2 / \bar{p}_i^T A \bar{p}_i \right) \\
\lambda_2 &= \left(\bar{p}_i^T r_1 + m_{i-1} ||z||^2 \quad \bar{q}_i^T r_1 + n_i ||z||^2 \right) \begin{pmatrix} d_i & \delta_i \\ \delta_i & e_i \end{pmatrix}^{-1} \begin{pmatrix} m_{i-1} \\ n_i \end{pmatrix}.
\end{aligned}$$

Now, observe that $\tilde{x}$ can be a least squares solution of (1) if and only if

$$\tilde{x} = A^{+}b + \tilde{z}, \quad \text{with } \tilde{z} \in N(A).$$

Thus, projecting $\tilde{x}$ in (26) onto the subspace $R(A)$, we simply have

$$\begin{aligned}\mathrm{Pr}_{R(A)}(\tilde{x}) = \mathrm{Pr}_{R(A)}(x_1) + \sum_{i \in S_1} \left[\left(\bar{p}_i \bar{p}_i^T / \bar{p}_i^T A \bar{p}_i \right) y + ||z||^2 \left(m_{i-1} / \bar{p}_i^T A \bar{p}_i \right) \bar{p}_i \right] \\ + \sum_{i \in S_2} \left[(\bar{p}_i \;\; \bar{q}_i) \begin{pmatrix} d_i & \delta_i \\ \delta_i & e_i \end{pmatrix}^{-1} \begin{pmatrix} \bar{p}_i^T \\ \bar{q}_i^T \end{pmatrix} y \right. && (27) \\ \left. + ||z||^2 (\bar{p}_i \;\; \bar{q}_i) \begin{pmatrix} d_i & \delta_i \\ \delta_i & e_i \end{pmatrix}^{-1} \begin{pmatrix} m_{i-1} \\ n_i \end{pmatrix} \right]. && (28)\end{aligned}$$

Finally, recalling (17), (21), and considering in (27) the terms which contain $||z||^2$, we conclude that, if $b \notin R(A)$ $\tilde{x}$ is not a least-square solution of the linear system (1). □

4. Lanczos Vectors and FLR Algorithm Residuals

In this section, we describe a twofold result: first, we report some theoretical properties of the Lanczos process (Table 2) when matrix A in (1) is singular. This aims at investigating possible similarities with the results of Section 3, where the FLR algorithm is studied in the singular case. Then, a relevant relationship between the Lanczos vectors $\{u_i\}$ and the residuals $\{r_i\}$ calculated by the FLR algorithm is accomplished. We prove that the proper choice of the parameter ϵ_k, at step k of the FLR algorithm, plays a key role for the latter purpose.

Theorem 4.1. Consider the linear system (1), where A is indefinite and possibly singular. Consider the Lanczos process in Table 2 and let $u_1 = y + z$, with $y = \mathrm{Pr}_{R(A)}(u_1)$ and $z = \mathrm{Pr}_{N(A)}(u_1)$. Let $\lambda_1, \ldots, \lambda_1, \lambda_k$, $k \leq n$, be the nonzero eigenvalues (possibly not all distinct) of the matrix A. Suppose that the vector y and the integer $\hat{l}$ satisfy the hypothesis of Lemma 2.1. Then, the Lanczos process generates the sequence of orthonormal vectors

$$u_i = U_{i-1}(A)y + \eta_{i-1}z, \quad 1 \leq i \leq \hat{l}, \tag{29}$$

where $U_j(\bullet)$ is a real polynomial of degree j and $\eta_j \in \mathbb{R}$, with $j \geq 3$,

$$U_0(A) = I, \qquad \eta_0 = 1, \tag{30a}$$

$$U_1(A) = (1/\delta_1)(A - \gamma_1 I)U_0(A), \qquad \eta_1 = -\gamma_1 \eta_0 / \delta_1, \tag{30b}$$

$$U_{j-1}(A) = (1/\delta_{j-1})[(A - \gamma_{j-1}I)U_{j-2}(A) - \delta_{j-2}U_{j-3}(A)],$$
$$\eta_{j-1} = -\frac{(\gamma_{j-1}\eta_{j-2} + \delta_{j-2}\eta_{j-3}),}{\delta_{j-1}}. \tag{30c}$$

Moreover, $u_i \notin N(A)$, for any $i \leq \hat{l}$.

Proof. From the hypothesis and Lemma 2.1, the Lanczos process performs exactly $\hat{l}$ iterations before stopping. Then, Table 2 yields

$$u_1 = y + z = U_0(A)y + \eta_0 z$$

and

$$\begin{aligned} u_2 &= v_1/\delta_1 \\ &= (1/\delta_1)[(A - \gamma_1 I)u_1 - \delta_0 u_0] \\ &= (1/\delta_1)(A - \gamma_1 I)U_0(A)y - \gamma_1(\eta_0/\delta_1)z, \end{aligned}$$

so that the first two lines in (30) hold. Finally, by Table 2, relation (29), and complete induction, we obtain, for $i \geq 3$,

$$\begin{aligned} u_i &= v_{i-1}/\delta_{i-1} \\ &= (1/\delta_{i-1})[(A - \gamma_{i-1}I)u_{i-1} - \delta_{i-2}u_{i-2}] \\ &= (1/\delta_{i-1})\{(A - \gamma_{i-1}I)[U_{i-2}(A)y + \eta_{i-2}z] - \delta_{i-2}[U_{i-3}(A)y + \eta_{i-3}z]\}, \end{aligned}$$

which yields the third line in (30). □

Lemma 4.1. Let matrix A in (1) be indefinite and possibly singular. Suppose the Lanczos process and the FLR algorithm are applied to solve (1), with $x_1 = 0$ in the FLR algorithm. Then, in exact arithmetic the Lanczos and FLR algorithms perform the same number of iterations.

Proof. Evidently, if at the step k both the Lanczos process and the FLR algorithm have not yet stopped, they have respectively generated the orthogonal sequences $u_1, \ldots, u_k$ and $t_1, \ldots, t_k$, in the Krylov subspaces $\mathcal{K}_k(u_1, A)$ and $\mathcal{K}_k(r_1, A)$. Since $x_1 = 0$,

$$\mathcal{K}_k(u_1 A) \equiv \mathcal{K}_k(r_1, A), \tag{31}$$

so that the statement holds from (13), (29), and Lemma 2.1. □

Theorem 4.2. The vectors u_i, $i \geq 1$, and $r_i/||r_i||$, $i \geq 1$, generated respectively by the Lanczos process and the FLR algorithm with $x_1 = 0$, in exact arithmetic satisfy the relation

$$u_i = s_i(r_i/||r_i||), \quad s_i \in \{+1, -1\}. \tag{32}$$

Proof. By complete induction, $x_1 = 0$ yields

$$u_1 = r_1/||r_1|| = b/||b||. \tag{33}$$

Now, suppose that

$$u_{i-1} = s_{i-1} r_{i-1}/||r_{i-1}||;$$

we prove that

$$u_i = s_i r_i/||r_i||.$$

On this purpose, let $\hat{l}$ be the number of iterations performed by Lanczos process and the FLR algorithm, according to Lemma 4.1. Recall that the Lanczos vectors $u_1, \ldots, u_{\hat{l}}$, satisfy $u_i^T u_j = 0,\ \hat{l} \geq i \neq j \geq 1$ (Ref. 3). Furthermore, considering at step k_B of the FLR algorithm the dummy residual (Refs. 24, 20)

$$r_{k+1} = -\bar{\alpha}_k r_k - (1 + \bar{\alpha}_k)\ \text{sign}\ (d_k) A p_k, \tag{34a}$$

$$\bar{\alpha}_k = -\frac{|d_k|}{(||r_k||^2 + |d_k|)}, \tag{34b}$$

$$\text{sign}(d_k) = \begin{cases} 1, & d_k \geq 0, \\ -1, & d_k < 0, \end{cases} \tag{34c}$$

the sequence $r_1, \ldots, r_{\hat{l}}$ satisfies $r_i^T r_j = 0$, where $\hat{l} \geq i \neq j \geq 1$ (Ref. 1). Now, observe that

$$u_i \in \mathcal{K}_i(u_1, A), \tag{35a}$$

$$u_i \perp \mathcal{K}_{i-1}(u_1, A) = \text{span}\{\mathcal{K}_{i-2}(u_1, A), u_{i-1}\}, \tag{35b}$$

$$r_i \in \mathcal{K}_i(r_1, A), \tag{35c}$$

$$r_i \perp \mathcal{K}_{i-1}(r_1, A) = \text{span}\{\mathcal{K}_{i-2}(r_1, A), r_{i-1}\}, \tag{35d}$$

and from (33) and the inductive hypothesis

$$\mathcal{K}_{i-1}(u_1, A) = \mathcal{K}_{i-1}(r_1, A).$$

Thus, from (33) and (35), u_i and r_i are parallel. Finally, since $||u_i|| = 1$, the relation (32) holds. □

Theorem 4.3. Consider the FLR algorithm in Table 1. Let $x_1 = 0$ and let at i_B the dummy residual (34) be calculated. At step i, if the parameter ϵ_i is chosen according with

$$0 < \bar{\epsilon} \leq \epsilon_i, \quad \text{step}\ i_A, \tag{36a}$$

$$0 < \bar{\epsilon} \leq \epsilon_i < \min\{||Ap_i||^2||r_i||^2/||p_i||^4,\ ||Ap_i||^4/(\lambda_M||p_i||^2||q_i||^2)\}, \text{step}\ i_B, \tag{36b}$$

then in exact arithmetic the sequences $\{u_i\}$ and $\{r_i/||r_i||\}$ generated by the Lanczos and FLR algorithms satisfy

$$u_i = s_i(r_i/||r_i||), \quad i \geq 1, \tag{37}$$

where

$$s_1 = 1, \tag{38a}$$

$$s_i = -s_{i-1}\,\text{sign}\big(p_{i-1}^T A p_{i-1}\big), \qquad \text{if step } (i-1)_A \text{ is performed}, \tag{38b}$$

$$s_{i-1} + -s_{i-2}\,\text{sign}\big(p_{i-2}^T A p_{i-2}\big), \qquad \text{if step } (i-2)_B \text{ is performed}, \tag{38c}$$

$$s_i = -s_{i-2}, \qquad \text{if step } (i-2)_B \text{ is performed}. \tag{38d}$$

Proof. See Ref. 25.

Remark 4.1. Observe that condition (36) on ϵ_i is slightly less restrictive in respect to condition (12) in Ref. 1, since it does not require the knowledge of λ_m. As regards the apparently cumbersome computation of q_i in (36), refer to the considerations in Ref. 1. We highlight also that the approximation of the Moore-Penrose pseudoinverse A^+, provided in (22) by algorithm FLR, is not inexpensively available from the Lanczos process. In particular, the set of directions $t_1, \ldots, t_{\hat{l}}$ should be ad hoc generated by the Lanczos process.

5. Lanczos Process from the FLR Algorithm

Note that the relations (37) and (38) are also a generalization of the results, reported in Ref. 26, by replacing CG with the FLR algorithm. In particular, in matrix terms, the Lanczos process gives at step k (Ref. 26)

$$T_k^{(L)} = U_k^T A U_k, \tag{39}$$

where

$$T_k^{(L)} = \begin{bmatrix} \delta_0 & \gamma_1 & & & \\ \gamma_1 & \delta_1 & \cdot & & \\ & & \cdots & & \\ & & \cdot & \delta_{k-1} & \gamma_k \\ & & & \gamma_k & \delta_k \end{bmatrix}, \qquad U_k = (u_1 \cdots u_k),$$

and relation (37) can be restated as

$$U_k = R_k S_k, \tag{40}$$

where

$$R_k = (r_1/||r_1||, \ldots, r_k/||r_k||)\,, \qquad S_k = \text{diag}_{1 \leq i \leq k}\{s_i\}.$$

Combining (39) and (40), we obtain

$$T_k^{(L)} = S_k^T \left(R_k^T A R_k\right) S_k = S_k T_k^{\mathrm{FLR}} S_k, \tag{41}$$

where the tridiagonal matrix T_k^{FLR} is available at step k of the FLR algorithm. The explicit expression of T_k^{FLR}, in terms of the coefficients of the FLR algorithm, is given in Ref. 24.

Proposition 5.1. In the hypothesis of Theorem 4.3 and in exact arithmetic, the tridiagonal matrix $T_k^{(L)}$ of the Lanczos process is a straighforward by product of the FLR algorithm, as indicated in (41).

Furthermore, in the hypothesis of Theorems 4.1 and 4.3, the solution $\tilde{x}$ of (1) provided by the Lanczos process, may be given in terms of the FLR algorithm quantities. Indeed (Ref. 21),

$$\tilde{x} = U_{\hat{l}}\left[T_{\hat{l}}^{(L)}\right]^{-1}\delta_0 e_1, \quad e_1^T = (1,\, 0,\, \ldots 0),$$

and from (40)–(41)

$$\tilde{x} = U_{\hat{l}} S_{\hat{l}} \left[T_{\hat{l}}^{\mathrm{FLR}}\right]^{-1} S_{\hat{l}}\delta_0 e_1 = R_{\hat{l}} \left[T_{\hat{l}}^{\mathrm{FLR}}\right]^{-1} S_{\hat{l}}\delta_0 e_1 = R_{\hat{l}} \left[T_{\hat{l}}^{\mathrm{FLR}}\right]^{-1} ||b|| e_1,$$

where the last equality follows from Table 2, relation $e_1^T = (1,\, 0,\, \ldots,\, 0)$, and (38).

6. Conclusions and Perspectives

This paper has described several properties of the FLR planar-CG algorithm, proposed in Ref. 1, for solving indefinite linear systems. We have proved that the sequence of orthogonal residuals $\{r_i\}$ of the FLR algorithm, yields the sequence of orthogonal vectors $\{u_i\}$ of the Lanczos process, provided that the parameter ϵ_i at step i of the FLR algorithm is chosen according with Theorem 4.3. Since the FLR algorithm is a cheap CG-type method, this result encourages us to consider a numerical comparison of these methods within nonconvex optimization frameworks, where efficient tools for the solution of indefinite linear systems are claimed.

On the other hand,we have studied the solution of the linear system $Ax = b,\ A \in \mathbb{R}^{n\times n}$ indefinite and possibly singular, by means of the FLR algorithm: this extended the results provided by the CG in the positive semidefinite case (Ref. 12).

We conclude that the FLR algorithm proved to be a general tool for the solution of symmetric linear systems i.e., for the search of stationary points of quadratic forms in unconstrained optimization frameworks. In addition, the approximation of the Moore-Penrose pseudoinverse A^+ provided by the FLR algorithm, may be a fruitful instrument for the construction of preconditioners (Ref. 27). Finally as Section 1 reported, the Newton method for the computation of real eigenvectors could gain advantage from the consideration of FLR algorithm.

References

1. FASANO, G., *Planar-Conjugate Gradient Algorithm for Large-Scale Unconstrained Optimization, Part 1: Theory*, Journal of Optimization Theory and Applications, Vol. 125, pp. 523–541, 2005.
2. HESTENES, M. R., and STIEFEL, E., *Methods of Conjugate Gradients for Solving Linear Systems*, Journal of Research of the National Bureau of Standards, Vol. 49B, pp. 409–436, 1952.
3. GOLUB, G. H., and VAN LOAN, C. F., *Matrix Computations*, 3rd Edition, John Hopkins Press, Baltimore, Maryland, 1989.
4. GREENBAUM, A., *Iterative Methods for Solving Linear Systems*, SIAM, Frontiers in Applied Mathematics, Philadelphia, Pennsylvania, 1997.
5. SAAD, Y., and VAN DER VORST, H. A., *Iterative Solution of Linear Systems in the 20th Century*, Journal of Computational and Applied Mathematics, Vol. 123, pp. 1–33, 2000.
6. HANSEN, P. C., *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, Pennsylvania, 1998.
7. BJORCK, A., *Numerical Methods for Least-Squares Problems*, SIAM, Philadelphia, Pennsylvania, 1996.
8. MCCORMICK, G. P., *Nonlinear Programming: Theory, Algorithms, and Applications*, Wiley and Sons, New York, NY, 1983.
9. CAMPBELL, S. L., and MEYER JR., C. D., *Generalized Inverses of Linear Transformations*, Dover Publications, New York, NY, 1979.
10. LAMPARIELLO, F., and SCIANDRONE, M., *Use of the Minimum-Norm Search Direction in a Nonmonotone Version of the Gauss-Newton Method*, Journal of Optimization Theory and Applications, Vol. 119, pp. 65–82, 2003.
11. FASANO, G., LAMPARIELLO, F., and SCIANDRONE, M., *A Truncated Nonmonotone Gauss-Newton Method for Large-Scale Nonlinear Least-Squares Problems*, Technical Report 590, IASI-CNR, Rome, Italy, 2004; Computational Optimization and Applications (to appear).
12. HESTENES, M. R., *Pseudoinverses and Conjugate Gradients*, Communications of the ACM, Vol. 18, pp. 40–43, 1975.
13. WU, K., SAAD, Y., and STATHOPOULOS, A., *Inexact Newton Preconditioning Techniques for Large Symmetric Eigenvalue Problems*, Electronic Transactions on Numerical Analysis, Vol. 7, pp. 202–214, 1998.
14. SLEIJPEN, G. L. G., and VAN DER VORST, H. A., *The Jacobi-Davidson Method for Eigenvalue Problems as an Accelerated Inexact Newton Scheme*, Iterative Methods in Linear Algebra II: Proceedings of the IMACS International Symposium on Iterative Methods in Linear Algebra, Blagoevgrad, Bulgaria, pp. 377–389, 1995; Elsevier, Amsterdam, Netherlands, 1996.
15. LUENBERGER, D. G., *Hyperbolic Pairs in the Method of Conjugate Gradients*, SIAM Journal on Applied Mathematics, Vol. 17, pp. 1263–1267, 1969.
16. HESTENES, M. R., *Conjugate Direction Methods in Optimization*, Springer Verlag, New York, NY, 1980.

17. LIU, Y., and STOREY, C., *Efficient Generalized Conjugate Gradient Algorithms, Part 1: Theory*, Journal of Optimization Theory and Applications, Vol. 69, pp. 129–137, 1991.
18. DIXON, L. C. W., DUCKSBURY, P. G., and SINGH, P., *A New Three-Term Conjugate Gradient Method*, Technical Report 130, Numerical Optimization Centre, Hatfield Polytechnic, Hatfield, Hertfordshire, England, 1985.
19. MIELE, A., and CANTRELL, J. W., *Study on a Memory Gradient Method for the Minimization of Functions*, Journal of Optimization Theory and Applications, Vol. 3, pp. 459–470, 1969.
20. BANK, R. E., and CHAN, T. F., *A Composite Step Biconjugate Gradient Algorithm for Nonsymmetric Linear Systems*, Numerical Algorithms, Vol. 7, pp. 1–16, 1994.
21. SAAD, Y., *Iterative Methods for Sparse Linear Systems*, 2nd Edition, SIAM, Philadelphia, Pennsylvania, 2003.
22. EIJKHOUT, V., *LAPACK Working Note 66; A Characterization of Polynomial Iterative Methods*, Technical Report UT-CS-93-216, Department of Computer Science, University of Tennessee, 1993.
23. EIJKHOUT, V., *LAPACK Working Note 51: Qualitative Properties of the Conjugate Gradient and Lanczos Methods in a Matrix Framework*, Technical Report UT-CS-92-170, Department of Computer Science, University of Tennessee, 1992.
24. FASANO, G., *Use of Conjugate Directions inside Newton-Type Algorithms for Large-Scale Unconstrained Optimization*, PhD Dissertation, Rome, Italy, 2001.
25. FASANO, G., *Lanczos-Conjugate Gradient Method and Pseudoinverse Computation in Unconstrained Optimization*, Technical Report INSEAN 2004-036, Rome, Italy, 2004.
26. CONN, A. R., GOULD, N. I. M., and TOINT, P. L., *Trust-Region Methods*, SIAM, Philadelphia, Pennsylvania, 2000.
27. KOSCHINSKI, C., *New Methods for Adapting and for Approximating Inverses as Preconditioners*, Applied Numerical Mathematics, Vol. 41, pp. 179–218, 2002.