

Planar-Conjugate Gradient Algorithm for Large-Scale Unconstrained Optimization

Part 1: *Theory*^{1,2}

G. FASANO³

Communicated by L.C.W.Dixon

Abstract. In this paper we define a new conjugate gradient (CG) based algorithm, in the class of planar conjugate gradient methods. These methods aim at solving systems of linear equations whose coefficient matrix is indefinite and nonsingular. This is the case where the application of the standard CG algorithm by Hestenes and Stiefel (Ref. 1) may fail, due to a possible division by zero. We give a complete proof of global convergence for a new planar method endowed with a general structure; furthermore, we describe some important features of our planar algorithm, which will be used within the optimization framework of the companion paper Part 2 (Ref. 2). Here, preliminary numerical results are reported.

Key Words. Large-scale unconstrained optimization, iterative methods, conjugate gradient, planar conjugate gradient, indefinite matrices.

1. Introduction

In this paper we describe a new iterative algorithm for solving *symmetric* linear systems with the following general form:

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

where $A \in \mathbb{R}^{n \times n}$ may be *indefinite* and *nonsingular*, $b \in \mathbb{R}^n$, and n is *large*. Several iterative algorithms were proposed in literature for the solution of (1), however when n is large a specific attention was devoted to iterative schemes, since their practical implementation often requires much less than $\mathcal{O}(n^3)$ floating point operations for a solution. This has led to the development of many iterative schemes (see Refs. 3 - 9 for complete tutorials), aiming at guaranteeing both efficiency and effectiveness in the computation.

In the last decades, a larger number of real-life industrial applications have considerably taken advantage of the use of iterative methods, for the solution of large scale indefinite linear systems. In addition the sparsity of the problems usually encourages the use of specific iterative methods (see Refs. 9, 5 for some references).

¹This work was supported by MIUR, FIRB Research Program on *Large-Scale Nonlinear Optimization*, Rome, Italy.

²The author acknowledges the daily presence of Luigi Grippo and Stefano Lucidi, who contributed considerably to the elaboration of this paper. The mutual exchange of experiences with Massimo Roma was a constant help for investigating this topic. Also, the author expresses his gratitude to the Associate Editor and the referees for suggestions and corrections.

³Postdoctoral Fellow, Department of Computer Science and Systems A.Ruberti, University of Rome La Sapienza, Rome, Italy.

The versatility of iterative algorithms in solving large-scale problem (1), suggests their natural embedding within optimization schemes too. In fact, consider the problem of minimizing the nonlinear function $f(x)$, where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice continuously differentiable. We adopt the iterative scheme $x_{k+1} = x_k + d_k$, where the sequence $\{x_k\}$ approaches the solution x^* and d_k is a suitable direction. We can use Newton method (Ref. 10) for efficiently calculating direction d_k which solves Newton equation (see Ref. 11)

$$\nabla^2 f(x_k)d + \nabla f(x_k) = 0, \quad d \in \mathbb{R}^n, \quad (2)$$

where $\nabla^2 f(x_k)$ and $\nabla f(x_k)$ are respectively the Hessian matrix and the gradient of nonconvex function $f(x)$, calculated at the current point x_k . The convergence performance of the optimization method is strongly affected by the accuracy in solving (2), however the adoption of truncated schemes when n is large, often does not require high precision in calculating an approximate solution d_k . In particular, whenever the current point x_k is far from x^* , the calculation of the *exact* solution of (2) turns to be worthlessly expensive. Thus, a reliable but simple iterative algorithm, which copes with the case of $\nabla^2 f(x_k)$ *indefinite*, would be highly desirable as a solver for the linear system (2). Unfortunately the solution of (2) may be eventually a saddle point or a maximum point of $f(x)$; therefore the globalization scheme should properly take into account the local information on $f(x)$ contained in the direction d_k , in order to avoid at least the convergence towards a maximum point. The latter result may be achieved, in a linesearch approach, by means of a proper application of the iterative method which solves large scale system (2). In particular, one way for obtaining such a result is using the iterative method for providing a couple of directions, say d_k and s_k , with the following purposes:

d_k approximately solves the system (2) and ensures the convergence performances in a neighborhood of the solution point x^* ;

s_k is a negative curvature direction of the function $f(x)$, i.e. $s_k^T \nabla^2 f(x_k) s_k \leq 0$, and is calculated in such a way that (Ref. 12)

$$s_k^T \nabla^2 f(x_k) s_k \longrightarrow 0 \quad \text{implies} \quad \min\{0, \lambda_m[\nabla^2 f(x_k)]\} \longrightarrow 0, \|s_k\| \longrightarrow 0,$$

where $\lambda_m[\nabla^2 f(x_k)]$ is the smallest eigenvalue of $\nabla^2 f(x_k)$. The direction s_k has the specific purpose of forcing the convergence of the optimization method towards the point x^* , which satisfies the second order necessary optimality conditions, i.e. $\nabla f(x^*) = 0$ and $s^T \nabla^2 f(x^*) s \geq 0$, $\forall s \in \mathbb{R}^n$.

Under mild assumptions (Ref. 13) it can be proved that, replacing the scheme $x_{k+1} = x_k + d_k$, with the scheme $x_{k+1} = x_k + \alpha_k d_k + \beta_k s_k$, for suitable $\alpha_k, \beta_k \in \mathbb{R}$, is efficient and effective for the convergence towards a x^* , that satisfies the second order necessary conditions of optimality (Ref. 12). We remark that in this case the choice of the iterative method is crucial for the calculation of vectors d_k and s_k .

Another way for avoiding the convergence of Newton method towards a maximum point is by means of a so called *modified Newton method* (Ref. 14), which is a globally convergent modification of Newton method, in case the Hessian matrix $\nabla^2 f(x_k)$ is not positive definite.

Here, the iterative method adopted for approximately solving equation (2), has to rearrange the information on $\nabla^2 f(x_k)$ provided by the Newton direction d_k . In particular, the iterative method should suitably separate the information contained in d_k , which is related to both the convexity and the concavity regions of $f(x)$ near x_k . In the related paper "*Planar-Conjugate Gradient algorithm for Large-Scale Unconstrained Optimization, Part 2: Application*" we shall consider a modified Newton

method, which uses the planar-CG algorithm proposed in this paper, within a linesearch framework. We shall give evidence about the importance of choosing the latter iterative method within an optimization framework, by solving several problems of CUTE collection (Ref. 15).

The method we propose here is an extension of the CG method, which is an example of simple and appealing iterative method that can be used as long as $\nabla^2 f(x_k)$ is positive definite. The CG method was originally proposed by Hestenes and Stiefel (Ref. 1) and is usually quite effective for approximately solving the symmetric linear system (1). There are several iterative variants of CG (see for instance Refs. 16 - 17), essentially aiming at a generalization of some properties of stability and accuracy. However, when the matrix A is *indefinite* and we try to apply the CG, the basic algorithm can stop beforehand and therefore it cannot be the method of choice.

Some attempts for overcoming the latter shortcoming are provided by the introduction of suitable iterative methods (see Refs. 18 - 20, 9). Essentially, like the CG they generate at step k ($k \leq n$) an increasing basis of independent vectors $\{b, Ab, \dots, A^{k-1}b\}$ (the Krylov subspace $\mathcal{K}_k(A, b)$); then they approximate the solution of (1) on this subspace. We can classify all these algorithms into the following two classes (Ref. 21):

- the *Ritz-Galerkin class*: includes those methods which provide at step k the new residual $r_k = b - Ax_k$, in such a way that:

$$r_k \perp \mathcal{K}_k(A, b);$$

- the *minimal residual class*: here the iterative schemes generate at step k a solution x_k according to:

$$x_k = \operatorname{argmin}_{x \in x_1 + \mathcal{K}_k(A, b)} \|Ax - b\|_2^2.$$

In this paper we focus on an iterative algorithm in the class of Ritz-Galerkin, which retains the low overall computational cost of CG with respect to Lanczos, MINRES, GMRES, and maintains a satisfactory exactness when applied for solving problem (1). More precisely we consider the category of *Planar Methods* (for specific references see Refs. 22 - 24 along with Refs. 25 - 28); the general rationale behind these methods may be roughly summarized as follows. Let A be indefinite and nonsingular, and suppose we apply the CG for solving (1). Let p_k be the conjugate direction at step k , such that $p_k^T A p_k = 0$, then a pivot breakdown occurs and the CG stops prematurely.

On the contrary, the planar CG methods generate a second direction q_k , and instead of performing the search of the stationary point on the line $x_k + \alpha p_k$, $\alpha \in \mathbb{R}$ (namely the k_A -th CG-step), they perform the search on the 2-dimensional linear manifold (namely the k_B -th planar step):

$$x_k + \operatorname{span}\{p_k, q_k\}. \quad (3)$$

The planar methods generate the direction q_k such that the set $\{p_1, \dots, p_k, q_k\}$ is independent. Moreover, they calculate the subsequent direction p_{k+2} according to the relations $p_i^T A p_{k+2} = q_i^T A p_{k+2} = 0$, $i \leq k$, i.e. p_{k+2} recovers the conjugacy with all the previous directions p_1, \dots, p_k, q_k .

The algorithm in Ref. 22 (addressed here as algorithm Hes) generates the vector q_k as follows: in any case at step k the couple $\{p_k, q_k\}$ is calculated, where the expression of q_k is such that $q_k^T A p_i = q_k^T A q_j = 0$, with $i, j \leq k - 1$. If p_k and q_k form a sufficiently wide angle, then the k_B -th planar step (3) is performed; otherwise the standard CG iteration is calculated. On the contrary, formally the algorithms in Ref. 23 (addressed here as Lue) and Ref. 24 (addressed here as Fas), provide the second direction q_k *if and only if* relation

$$p_k^T A p_k = 0 \quad (4)$$

holds. Consequently in case $0 < |p_k^T A p_k| < \varepsilon_k$, $k < n$ and ε_k is a **small** number, the algorithms Lue and Fas perform a standard CG-step, even though it might be numerically unstable. Thus, further accuracy in the practical implementation of these methods must be used, otherwise they might work out inaccurate solutions. Algorithm Hes does not suffer for the latter drawback and in our experience it usually provides more precise solutions with respect to the others. On the other hand Algorithms Lue and Fas are computationally cheaper. Moreover, by setting condition (4) in Hestenes method, we can straightforwardly device the coefficients of Algorithms Lue and Fas. Therefore the latter algorithms will be roughly interpreted as a “simplification” of the former one. On this stream, the present paper is concerned with introducing and developing an iterative algorithm, which recovers both the general structure of Algorithm Hes and the low computational cost of Algorithms Lue and Fas, in solving problem (1).

In the following sections we use the symbol $\|\cdot\|$ to denote both the Euclidean norm of a real n -dimensional vector and a real $n \times n$ matrix. Moreover we shall use either the notation $x^T y$ or $\langle x, y \rangle$ for the inner product between vectors x, y . The angle between vectors x and y will be indicated with $\widehat{x, y}$, the field of complex numbers by \mathbf{C} and with “ $x \perp y$ ” we mean that $x^T y = 0$. Finally λ_M and λ_m denote the largest and the smallest absolute value of the eigenvalues of the Hessian matrix $\nabla^2 f(x_k)$ (which will be often addressed as matrix A), and the symbol “ \triangleq ” stands for “...by definition...”. All the quantities calculated at step k will be reported with subscript k .

Section 2 introduces the new proposed algorithm where we suppose that the matrix A is *indefinite* and *nonsingular*; the convergence properties are extensively pointed out in Section 2.1. Within these sections a full analysis of global convergence is carried out. Finally, Section 2.2 deals with some features of the directions generated by our algorithm, and Section 3 carries on a few conclusions and perspectives, related to the treated subject.

2. New Planar Algorithm

As already observed in the previous section, at step k the planar methods contain a test for switching between the CG-step k_A (in the manifold $x_k + \alpha p_k$, $\alpha \in \mathbb{R}$) and the planar step k_B (in the manifold (3)): this test can seriously affect the behaviour of the algorithms. For Lue and Fas the test simply attempts to verify whenever $p_k^T A p_k = 0$. Thus, when $p_k^T A p_k$ is “small” but not exactly zero, the application of these algorithms may involve some numerical approximations. This shortcoming turns to be less relevant for algorithm Hes, since at step k the test on quantity $\Delta_k = (p_k^T A p_k)(q_k^T A q_k) - (p_k^T A q_k)^2$ (i.e. $|\Delta_k| \leq \epsilon_k (p_k^T A q_k)^2$, $\epsilon_k = 1/2$) is an “inequality test” (see also Section 2.2). However, the test on the quantity Δ_k is more expensive since it requires, at each step, the computation of the vectors $A p_k, A q_k$. Hence we conclude that for the planar algorithms investigated, there is the following trade-off: if the internal test is computationally cumbersome (e.g. Hes with respect to Lue and Fas), the algorithm is less sensible to numerical approximations. Of course this property holds within each iteration; thus, no final conclusion can be argued *a priori* on the overall behaviour of the algorithms.

A question which deserves a further investigation, is the possibility of developing a new planar algorithm from Hes, with the following two features:

- it must avoid the troublesome check of relation $p_k^T A p_k = 0$ and replace it with an inequality test;
- it must preserve the low computational cost of Algorithms Lue and Fas, in order to be computationally cheaper than algorithm Hes.

In Table 1 we propose algorithm FLR, which matches the latter requirements and partially recovers the features of the algorithm in Ref. 22.

Table 1: Algorithm **FLR** 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 $d_k = p_k^T Ap_k$; 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 $a_k = r_k^T p_k / d_k$, $x_{k+1} = x_k + a_k p_k$, $r_{k+1} = r_k - a_k Ap_k$.
 If $r_{k+1} = 0$, then STOP. Else, set $b_k = -p_k^T Ar_{k+1} / d_k$ 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 .

2.1. Convergence Results

The following results can be established for the algorithm FLR described in Table 1. We introduce the following convention which will be used for the proofs, in order to simplify the treatment ($i \leq n$):

$$\begin{aligned} \text{if } |p_i^T Ap_i| \geq \epsilon_i \|p_i\|^2 & \quad \text{then set} \quad \alpha_i = a_i \quad \text{and} \quad t_i = p_i, \\ \text{if } |p_i^T Ap_i| < \epsilon_i \|p_i\|^2 & \quad \text{then set} \quad \begin{cases} \alpha_i = \hat{c}_i \quad \text{and} \quad t_i = p_i, \\ \alpha_{i+1} = \hat{d}_i \quad \text{and} \quad t_{i+1} = q_i. \end{cases} \end{aligned}$$

Lemma 2.1. If residual r_{k+1} [r_{k+2}], calculated at step k_A [k_B] of algorithm FLR, is not the null vector, then the directions t_i , $i = 1, \dots, k+1$ [$k+2$], do not coincide with the null vector.

Proof.

The statement follows directly from the definition of r_{k+1} at Step k_A and r_{k+2} at Step k_B . \square

Lemma 2.2. If $r_k \neq 0$, then $d_k = 0$ implies $\Delta_k \neq 0$.

Proof.

Suppose $d_k = 0$ and by contradiction consider $\Delta_k = 0$. This implies

$$0 = d_k e_k - \delta_k^2 = -\delta_k^2 \stackrel{\Delta}{=} -(q_k^T Ap_k)^2. \quad (5)$$

If the previous step was step $(k-1)_A$ (i.e. $d_{k-1} \neq 0$), by construction $p_k^T Ap_{k-1} = 0$, therefore relation (5) becomes $0 = [(Ap_k + \beta_{k-1} p_{k-1})^T Ap_k]^2 = \|Ap_k\|^4$, which is a contradiction since matrix A is

nonsingular and Lemma 2.1 holds. If the previous step was step $(k-2)_B$ (i.e. $\Delta_{k-2} \neq 0$), relation (5) becomes

$$0 = \left[\left(Ap_k + \hat{\beta}_{k-2}(d_{k-2}q_{k-2} - \delta_{k-2}p_{k-2})/\Delta_{k-2} \right)^T Ap_k \right]^2 = \|Ap_k\|^4,$$

where the last equality is obtained by construction (the choice of coefficients \hat{b}_{k-2} and $\hat{\beta}_{k-2}$). Again, Lemma 2.1 yields a contradiction. \square

The previous lemma reveals that if matrix A is indefinite and nonsingular, algorithm FLR can *always perform* either step k_A or step k_B , hence it is well defined and cannot stick. In other words, provided that the solution of (1) is not yet detected, from a theoretical viewpoint *the algorithm cannot stop*.

Theorem 2.1. If residual r_{k+1} [or r_{k+2}], calculated at Step k_A [or k_B] of Algorithm FLR, is not the null vector, then we have

$$At_k \in \text{span}\{t_1, \dots, t_{k+1}\}, \quad (6)$$

and the following properties hold:

$$\begin{aligned} \text{(A1)} \quad p_{k+1}^T At_i = 0, \quad i \leq k & \quad [\text{(A2)} \quad p_{k+2}^T At_i = 0, \quad i \leq k+1] \\ & \quad [\text{(B2)} \quad q_k^T At_i = 0, \quad i \leq k-1] \\ \text{(C1)} \quad r_{k+1}^T t_i = 0, \quad i \leq k & \quad [\text{(C2)} \quad r_{k+2}^T t_i = 0, \quad i \leq k+1] \\ \text{(D1)} \quad r_{k+1}^T r_i = 0, \quad i \leq k & \quad [\text{(D2)} \quad r_{k+2}^T r_i = 0, \quad i \leq k] \\ \text{(E1)} \quad r_i^T p_{k+1} = r_1^T p_{k+1}, \quad i \leq k+1 & \quad [\text{(E2)} \quad r_i^T p_{k+2} = r_1^T p_{k+2}, \quad i \leq k+2]. \end{aligned}$$

Moreover item (B2) holds if $r_{k+2} = 0$ too.

Proof.

We prove the statements by means of complete induction. At first we verify them with $k=1$, then we suppose they hold for index $k-1$, finally they will be proved for index k . The symbol $(\Delta i)_h$, $\Delta = A, B, C, D, E$; $i = 1, 2$; $h = 1, k-1, k$ will be used to refer the item (Δi) in the statement of the theorem, at step h of the induction process.

For $k=1$ it is either $t_1 = p_1$ (if the first step was 1_A) or $t_1 = p_1$ (if the first step was 1_B). In either the case, since $\alpha_1 = a_1 \neq 0$ we obtain $Ap_1 = At_1 \in \text{span}\{t_1, t_2\}$. Moreover for $k=1$, we have:

$$\text{(A1)}_1: \text{ here } t_1 = p_1, \text{ thus } p_{1+1}^T At_1 = p_2^T Ap_1 = (r_2 + b_1 p_1)^T Ap_1 = 0.$$

$$\text{(A2)}_1: \text{ Here we have to consider both the cases } p_{1+2}^T Ap_1 \text{ (} t_1 = p_1 \text{) and } p_{1+2}^T Aq_1 \text{ (} t_2 = q_1 \text{), which yield respectively:}$$

$$p_{1+2}^T Ap_1 = \left(r_3 + \hat{b}_1(d_1 q_1 - \delta_1 p_1)/\Delta_1 \right)^T Ap_1 = r_3^T Ap_1 \stackrel{\Delta}{=} r_3^T q_1 = 0,$$

where the last equality is due to the choice of coefficients \hat{c}_1 and \hat{d}_1 , and

$$p_{1+2}^T Aq_1 = \left(r_3 + \hat{b}_1(d_1 q_1 - \delta_1 p_1)/\Delta_1 \right)^T Aq_1 = r_3^T Aq_1 + \hat{b}_1 = 0.$$

(B2)₁: Here the first significative value for index k is $k = 3$ (i.e. step 3_B was preceded by step 1_B ⁴), thus there will be the two cases $q_3^T Ap_1$ ($t_1 = p_1$) and $q_3^T Aq_1$ ($t_2 = q_1$). For the former one it is:

$$q_3^T Ap_1 = \left(Ap_3 + \hat{\beta}_1(d_1q_1 - \delta_1p_1)/\Delta_1 \right)^T Ap_1 = (Ap_3)^T Ap_1 = p_3^T Aq_1 = 0,$$

where the last equality is a consequence of item (A2)₁. The case $q_3^T Aq_1 = 0$ similarly holds for the choice of coefficient $\hat{\beta}_1$.

(C1)₁: We have the only case $t_1 = p_1$: $r_{1+1}^T p_1 = (r_1 - a_1 Ap_1)^T p_1 = 0$.

(C2)₁: Two cases are possible (respectively with $t_1 = p_1$ and $t_2 = q_1$); and we simply obtain $r_{1+2}^T p_1 = r_{1+2}^T q_1 = 0$, where the last equalities follow from the choice of coefficients \hat{c}_1 and \hat{d}_1 .

Suppose now that the statements hold for index $k - 1$, let us prove they hold for index k . In order to prove $At_k \in \text{span}\{t_1, \dots, t_{k+1}\}$, three cases must be considered: t_{k+1} is calculated at step k_A (i.e. $t_{k+1} = p_{k+1}$), t_{k+1} is calculated at step $(k - 1)_B$ (i.e. again $t_{k+1} = p_{k+1}$), $t_{k+1} = q_k$. In the first case we have $a_k \neq 0$, therefore $t_{k+1} = r_{k+1} + b_k p_k = (p_1 - \sum_{j=1}^k \alpha_j At_j) + b_k p_k$, so that the inductive hypothesis yields $Ap_k = At_k \in \text{span}\{t_1, \dots, t_{k+1}\}$. In the second case we have analogously:

$$t_{k+1} = p_{k+1} = \left(p_1 - \sum_{j=1}^k \alpha_j At_j \right) + \hat{b}_{k-1}(d_{k-1}q_{k-1} - \delta_{k-1}p_{k-1})/\Delta_{k-1},$$

hence, by the inductive hypothesis $Aq_{k-1} = At_k \in \text{span}\{t_1, \dots, t_{k+1}\}$ ⁵.

The third case will be split in two subcases: the previous step was $(k - 1)_A$, or the previous step was $(k - 2)_B$. In the first subcase we have for q_k the expression:

$$t_{k+1} = q_k = Ap_k + \beta_{k-1}p_{k-1} \implies Ap_k = At_k \in \text{span}\{t_{k-1}, t_{k+1}\},$$

while the second subcase follows similarly.

(A1)_k: Two subcases are possible: $p_{k+1}^T Ap_i$ ($t_i = p_i$) with $i \leq k$ and $p_{k+1}^T Aq_{i-1}$ ($t_i = q_{i-1}$) with $i \leq k - 1$ (where the last relation between indices i and k holds since p_{k+1} was calculated at step k_A). In the first subcase if $i = k$, then (A1)_k holds for the choice of coefficient b_k ; if $i < k$ we have $p_{k+1}^T Ap_i = (r_{k+1} + b_k p_k)^T Ap_i = r_{k+1}^T Ap_i$, where the last equality is a consequence of the inductive hypothesis. Moreover we obtain $r_{k+1}^T Ap_i = (r_k - a_k Ap_k)^T Ap_i = 0$, $i < k$, from the inductive hypothesis, relation (6) and the choice of a_k .

Now let us examine the second subcase:

$$p_{k+1}^T Aq_{i-1} = (r_{k+1} + b_k p_k)^T Aq_{i-1} = r_{k+1}^T Aq_{i-1}, \quad i \leq k - 1, \quad (7)$$

which is a consequence of the inductive hypothesis. Now observe that vector q_{i-1} was introduced at step $(i - 1)_B$, moreover from relation (6) we have:

$$Aq_{i-1} = At_i \in \text{span}\{t_1, \dots, t_{i+1}\}, \quad i \leq k - 1.$$

Hence for $i = k - 1$ we obtain $Aq_{k-2} \in \text{span}\{t_1, \dots, t_k\}$, therefore by substituting in (7), two cases will be allowed: either $r_{k+1}^T t_k = 0$ for the choice of coefficient a_k , or $r_{k+1}^T t_j = 0$, $j < k$, for the inductive hypothesis. For $i < k - 1$, considering again relation $Aq_{i-1} = At_i \in \text{span}\{t_1, \dots, t_{i+1}\}$ in (7) and the inductive hypothesis, we obtain again $p_{k+1}^T Aq_{i-1} = 0$.

⁴Observe that the case $k = 2$ where step 2_B is preceded by step 1_A is a trivial case.

⁵Indeed suppose by contradiction $\alpha_k = 0$, for the inductive hypothesis this means $p_{k+1} \in \text{span}\{t_1, \dots, t_k\}$, which yields a contradiction since p_{k+1} is also conjugate to the linear subspace $\text{span}\{t_1, \dots, t_k\}$.

(A2)_k: Again two subcases can occur: $p_{k+2}^T Ap_i$ ($t_i = p_i$) with $i \leq k$ and $p_{k+2}^T Aq_{i-1}$ ($t_i = q_{i-1}$) with $i \leq k + 1$. In the first subcase if $i = k$ then $p_{k+2}^T Ap_k = (r_{k+2} + \hat{b}_k(d_k q_k - \delta_k p_k)/\Delta_k)^T Ap_k = r_{k+2}^T Ap_k$, moreover from relation (6) we have:

$$Ap_k = At_k \in \text{span}\{t_1, \dots, t_{k+1}\} \equiv \text{span}\{t_1, \dots, t_{k-1}, p_k, q_k\}.$$

Now, since $r_{k+2}^T p_k = r_{k+2}^T q_k = 0$ for the choice of coefficients \hat{c}_k and \hat{d}_k , in order to prove that $r_{k+2}^T Ap_k = 0$ it suffices to show that $r_{k+2}^T w = 0$, $\forall w \in \text{span}\{t_1, \dots, t_{k-1}\}$. On this purpose the inductive hypothesis yields:

$$r_{k+2}^T w = \left(r_k - \hat{c}_k Ap_k - \hat{d}_k Aq_k \right)^T w = 0, \quad \forall w \in \text{span}\{t_1, \dots, t_{k-1}\}.$$

On the other hand, if $i < k$ we have:

$$p_{k+2}^T Ap_i = \left(r_{k+2} + \hat{b}_k(d_k q_k - \delta_k p_k)/\Delta_k \right)^T Ap_i = r_{k+2}^T Ap_i = 0, \quad i < k,$$

where the second equality is a consequence of inductive hypothesis, and the last equality holds from relation (6), the inductive hypothesis and the choice of coefficients \hat{c}_k , \hat{d}_k .

In the second subcase, if $i = k + 1$, then (A2)_k holds because of the choice of coefficient \hat{b}_k ; moreover, if $i \leq k$ then:

$$p_{k+2}^T Aq_{i-1} = \left(r_{k+2} + \hat{b}_k(d_k q_k - \delta_k p_k)/\Delta_k \right)^T Aq_{i-1} = r_{k+2}^T Aq_{i-1}, \quad i \leq k,$$

where the last equality is a consequence of the inductive hypothesis. Now, by simply following the guidelines of the proof for the second subcase of (A1)_k (see relation (7)), we obtain

$$r_{k+2}^T Aq_{i-1} = 0, \quad i \leq k.$$

(B2)_k: The vector q_k is calculated at step k_B and two subcases are possible: either the previous iteration was step $(k-1)_A$ or the previous iteration was step $(k-2)_B$. In the first subcase it is $q_k = Ap_k + \beta_{k-1} p_{k-1}$, thus, if $i = k-1$ (i.e. $t_i = p_{k-1}$) then the choice of coefficient β_{k-1} yields $q_k^T Ap_{k-1} = 0$. On the other hand if $i < k-1$, then we have

$$q_k^T At_i = (Ap_k + \beta_{k-1} p_{k-1})^T At_i = (Ap_k)^T (At_i) + \beta_{k-1} p_{k-1}^T At_i = 0,$$

where the last equality follows from the inductive hypothesis and relation (6).

In the second subcase it is $q_k = Ap_k + \hat{\beta}_{k-2}(d_{k-2} q_{k-2} - \delta_{k-2} p_{k-2})/\Delta_{k-2}$. Then, if $i = k-1$ ($t_i = q_{k-2}$) the choice of coefficient $\hat{\beta}_{k-2}$ yields (B2)_k. If $i = k-2$ ($t_i = p_{k-2}$) it suffices to observe that

$$q_k^T Ap_{k-2} = \left(Ap_k + \hat{\beta}_{k-2}(d_{k-2} q_{k-2} - \delta_{k-2} p_{k-2})/\Delta_{k-2} \right)^T Ap_{k-2} = 0,$$

where the last equality is a consequence of relation (6) and (A2)_k. Finally, when $i < k-2$ we have relations:

$$q_k^T At_i = \left(Ap_k + \hat{\beta}_{k-2}(d_{k-2} q_{k-2} - \delta_{k-2} p_{k-2})/\Delta_{k-2} \right)^T At_i = (Ap_k)^T At_i,$$

where the inductive hypothesis yields the last equality. Again invoking relation (6) and using (A2)_k we complete the proof of item (B2).

(C1)_k: Since residual r_{k+1} was calculated at step k_A we have $r_{k+1}^T t_i = (r_k - a_k A p_k)^T t_i$, $i \leq k$. If $i = k$, the choice of a_k annihilates the right hand side. On the other hand if $i < k$ it can be either $t_i = p_i$ or $t_i = q_{i-1}$, then the inductive hypothesis annihilates both terms of the right hand side.

(C2)_k: Since residual r_{k+2} was calculated at step k_B we have $r_{k+2}^T t_i = (r_k - \hat{c}_k A p_k - \hat{d}_k A q_k)^T t_i$, $i \leq k+1$, and we can follow step by step the guideline of (C1)_k.

This completes the proof of items (A1), (A2), (B2), (C1), (C2).

(D1): It can be proved by means of complete induction. For $k = 1$ we have $r_{1+1}^T r_i = r_2^T r_1 = r_2^T p_1 = 0$, where the last equality holds from point (C1). Now suppose (D1) holds for index $k - 1$, let us prove it for index k . Since residual r_{k+1} was calculated at step k_A , we have relations:

$$\begin{aligned} r_{k+1}^T r_i &= (r_k - a_k A p_k)^T (p_i - b_{i-1} p_{i-1}), & i \leq k \\ r_{k+1}^T r_i &= (r_k - a_k A p_k)^T \left(p_i - \hat{b}_{i-2} (d_{i-2} q_{i-2} - \delta_{i-2} p_{i-2}) / \Delta_{i-2} \right), & i \leq k. \end{aligned}$$

When $i = k$, items (A1), (A2) and (C1) along with the choice of coefficient a_k annihilate the right hand sides of the previous relations. When $i < k$, then items (A1), (A2), (C1) and the inductive hypothesis again annihilate the right hand sides of the last relations.

(D2): Substantially we follow the guidelines of the previous item (by means of complete induction). For $k = 1$ we obtain relation $r_{1+2}^T r_i = r_3^T r_1 = r_3^T p_1 = 0$, where (C2) yields the last equality. Now suppose (D2) holds for index $k - 1$. Observe that residual r_{k+2} was calculated at step k_B , then we obtain relations:

$$\begin{aligned} r_{k+2}^T r_i &= (r_k - \hat{c}_k A p_k - \hat{d}_k A q_k)^T (p_i - b_{i-1} p_{i-1}), & i \leq k \\ r_{k+2}^T r_i &= (r_k - \hat{c}_k A p_k - \hat{d}_k A q_k)^T \left(p_i - \hat{b}_{i-2} (d_{i-2} q_{i-2} - \delta_{i-2} p_{i-2}) / \Delta_{i-2} \right), & i \leq k \end{aligned}$$

and with a similar reasoning in respect to (D1), the right hand sides of the previous relations are zero.

(E1): From item (A1) we have directly relations

$$r_i^T p_{k+1} = \left(r_1 - \sum_{j=1}^{i-1} \alpha_j A t_j \right)^T p_{k+1} = r_1^T p_{k+1}, \quad i \leq k + 1.$$

(E2): Analogously to (E1) we get $r_i^T p_{k+2} = r_1^T p_{k+2}$, $i \leq k + 2$, from item (A2). \square

Theorem 2.2. Suppose matrix A is indefinite and nonsingular, and Algorithm FLR generated directions t_1, \dots, t_k . Then t_1, \dots, t_k are linearly independent.

Proof.

By contradiction suppose

$$\sum_{s=1}^k \gamma_s t_s = 0, \quad \gamma_s \in \mathbb{R}, \quad (8)$$

where at least one of the coefficients γ_s (say $\gamma_{\bar{s}}$) is non zero. Two cases must be considered, depending on whether Algorithm FLR performed step \bar{s}_A or step \bar{s}_B ⁶.

In the former case we simply multiply both the sides of relation (8) by the vector $Ap_{\bar{s}}$, and since $p_{\bar{s}}^T Ap_{\bar{s}} \neq 0$, we obtain the contradiction:

$$0 = \gamma_{\bar{s}} p_{\bar{s}}^T Ap_{\bar{s}} \iff \gamma_{\bar{s}} = 0.$$

In the second case the algorithm performed the planar step \bar{s}_B and generated the directions $t_{\bar{s}} = p_{\bar{s}}$ and $t_{\bar{s}+1} = q_{\bar{s}}$; thus, by multiplying relation (8) by vectors $Ap_{\bar{s}}$ and $Aq_{\bar{s}}$, and considering that $\Delta_{\bar{s}} \neq 0$, we obtain the contradiction:

$$\begin{aligned} 0 = \gamma_{\bar{s}} p_{\bar{s}}^T Ap_{\bar{s}} + \gamma_{\bar{s}+1} q_{\bar{s}}^T Ap_{\bar{s}}, \quad 0 = \gamma_{\bar{s}} p_{\bar{s}}^T Aq_{\bar{s}} + \gamma_{\bar{s}+1} q_{\bar{s}}^T Aq_{\bar{s}} \\ \Downarrow \\ \gamma_{\bar{s}} = \gamma_{\bar{s}+1} = 0. \end{aligned}$$

□

The next theorem summarizes the convergence features of the proposed algorithm.

Theorem 2.3. Suppose that the symmetric matrix A in (1) is indefinite and nonsingular, then Algorithm FLR solves linear system (1) in at most n steps.

Proof.

At step k , Algorithm FLR has already generated k linearly independent directions t_1, \dots, t_k , thus $k \leq n$. In addition, suppose the algorithm stops at step m , then the point

$$x^* = x_1 + \sum_{i=1}^m \alpha_i t_i, \quad m \leq n, \quad (9)$$

is a solution of problem (1), i.e. $r_1 = \sum_{i=1}^m \alpha_i At_i$. Indeed, this follows from Theorem 2.1, after the multiplication of relation (9), by means of either vector Ap_i (step i_A) or vectors Ap_i, Aq_i (step i_B). Thus, we obtain the expressions of α_i (step i_A) and α_i, α_{i+1} (step i_B) in Algorithm FLR. □

We remark that, like the CG, in this planar scheme for each step we attempt to determine the solution of system (1) on a linear manifold, whose dimension is increased step by step.

A final numerical consideration should be pointed out with regard to the quantity Δ_k . In fact we can interpret the statement of Lemma 2.2 in the following *weaker form*: although quantities d_k and Δ_k cannot be both zero, whenever d_k is near zero then Δ_k may be near zero too. Of course this situation may occur in practice and the application of Algorithm FLR stops prematurely: this motivates further investigation on the properties of the quantity Δ_k in the next section.

2.2. Direction angles generated by Algorithm FLR

In this section we shall point out an interesting feature of the vectors t_i , $i \geq 1$ (see Section 2.1) generated by algorithm FLR. In particular, we prove that the test performed at step k by the latter algorithm, affects the angles among the directions that it generates.

⁶More exactly a third possibility should be examined, in case algorithm FLR performs step $(\bar{s}-1)_B$. However this case may be treated likewise the second case.

2.2.1. Direction angle in a planar step

Let us consider the test at step k of algorithm FLR; we are concerned with proving the following:

Proposition 2.1. At step k_B of algorithm FLR the relation $\Delta_k = 0$ holds if and only if vectors p_k and q_k are linearly dependent.

Proof.

After a short calculation we can verify the relation $\Delta_k = \det(\tilde{A})$, where:

$$\tilde{A} = \begin{pmatrix} p_k & q_k \\ p_k & q_k \end{pmatrix}^T \begin{pmatrix} A/2 & 0 \\ 0 & A/2 \end{pmatrix} \begin{pmatrix} p_k & q_k \\ p_k & q_k \end{pmatrix} \in \mathbb{R}^{2 \times 2}, \quad (10)$$

hence $\Delta_k = 0$ if and only if the matrix \tilde{A} has not full rank (Sylvester's inequality), that is *if and only if* vectors p_k and q_k are parallel. \square

This result ensures that if $\Delta_k \neq 0$ at step k_B , then vectors p_k and q_k identify a “plane”. Now, we prove that by a proper choice of the parameter ϵ_k we have (see also Ref. 22):

$$-\delta_k^2 \leq \Delta_k \leq -\delta_k^2/2, \quad (11)$$

where $\Delta_k = d_k e_k - \delta_k^2$, with $d_k = p_k^T A p_k$, $e_k = q_k^T A q_k$, $\delta_k = p_k^T A q_k = \|A p_k\|^2$. On this purpose let $\bar{\epsilon} > 0$, and suppose p_k and q_k are both available at step k . Then, set $\epsilon_k > 0$ according to the expression

$$\bar{\epsilon} \leq \epsilon_k \leq \min \left\{ \lambda_M^2 \|p_k\| / \|q_k\|, \lambda_m^4 \|p_k\|^2 / (2\lambda_M \|q_k\|^2) \right\}. \quad (12)$$

Proposition 2.2. Suppose at step k of algorithm FLR the test

$$|d_k| \leq \epsilon_k \|p_k\|^2 \quad (13)$$

holds, where ϵ_k is chosen according to (12). Then, at step k_B of algorithm FLR the following relation holds

$$-\delta_k^2 \leq \Delta_k \leq -\delta_k^2/2, \quad (14)$$

where $\delta_k = p_k^T A q_k = \|A p_k\|^2$.

Proof.

At first observe that by definition $d_k = 0$ implies $\Delta_k = -\delta_k^2$. Moreover, in case we are performing the k_B -th step we have $|d_k e_k| = |(p_k^T A p_k)(q_k^T A q_k)| \leq \epsilon_k \|p_k\|^2 |q_k^T A q_k| \leq \epsilon_k \lambda_M \|p_k\|^2 \|q_k\|^2$. This means that in order to impose the relation

$$|(p_k^T A p_k)(q_k^T A q_k)| \leq \delta_k^2/2, \quad (15)$$

it suffices to set ϵ_k such that $\epsilon_k \lambda_M \|p_k\|^2 \|q_k\|^2 \leq \delta_k^2/2$. Now, from the expression of q_k and Theorem 2.1, $\delta_k^2 = \|A p_k\|^4$, thus the previous relation yields $\epsilon_k \leq \|A p_k\|^4 / (2\lambda_M \|p_k\|^2 \|q_k\|^2)$, which straightforwardly holds in case ϵ_k satisfies relation (12). Therefore, in case ϵ_k is chosen sufficiently small, according to (12), we have from (15)

$$\Delta_k = d_k e_k - \delta_k^2 \leq |(p_k^T A p_k)(q_k^T A q_k)| - \|A p_k\|^4 \leq \|A p_k\|^4/2 - \|A p_k\|^4 = -\|A p_k\|^4/2, \quad (16)$$

which is the rightmost inequality (14). \square

Now, suppose relation (14) holds. Then, from Proposition 2.1 and the relation $\|p_k\| \geq \|r_k\| > 0$ vectors p_k and q_k are not parallel. Moreover, Proposition 2.2 ensures that there exists a negative constant σ_k such that

$$\Delta_k = \sigma_k \delta_k^2, \quad -1 \leq \sigma_k \leq -1/2. \quad (17)$$

We shall prove now that the relation (17) implies a condition over the angle φ_k between p_k and q_k . On this purpose we rearrange relation (17) as $\Delta_k - \sigma_k \delta_k^2 = 0$. Then, similarly to (10) we will find the matrix $B_k \in \mathbb{R}^{2 \times 2}$ such that $\Delta_k - \sigma_k \delta_k^2 = \det(B_k)$. Finally we shall point out suitable conditions on the coefficients of B_k , by imposing $\Delta_k - \sigma_k \delta_k^2 = 0$ (i.e. relation (17)): the latter conditions will be used for investigating the angle φ_k . To this end we want to determine a pair of complex coefficients a and b (and the corresponding complex conjugate \bar{a} and \bar{b}), which verify the relation:

$$\begin{aligned} 0 = \Delta_k - \sigma_k \delta_k^2 &= (p_k^T A p_k) (q_k^T A q_k) - (1 - \sigma_k) \delta_k^2 = \det(B_k) \\ &= \det \left[\begin{pmatrix} a p_k & q_k \\ p_k & b q_k \end{pmatrix}^T \begin{pmatrix} A/2 & \emptyset \\ \emptyset & A/2 \end{pmatrix} \begin{pmatrix} \bar{a} p_k & q_k \\ p_k & \bar{b} q_k \end{pmatrix} \right] \\ &= \det \begin{bmatrix} 1/2(a\bar{a} + 1)p_k^T A p_k & 1/2(a + \bar{b})p_k^T A q_k \\ 1/2(\bar{a} + b)q_k^T A p_k & 1/2(b\bar{b} + 1)q_k^T A q_k \end{bmatrix}. \end{aligned} \quad (18)$$

Thus, if we indicate with $|a|$ and $|b|$ the moduli of a and b , from the calculation of the last determinant we can deduce the conditions:

$$\begin{aligned} (|a|^2 + 1)(|b|^2 + 1)/4 &= 1 \\ |a + \bar{b}|^2 / 4 &= 1 - \sigma_k, \end{aligned} \quad (19)$$

which will be used later on. Now, observe that

$$\left| \left\langle \begin{pmatrix} \bar{a} p_k \\ p_k \end{pmatrix}, \begin{pmatrix} q_k \\ \bar{b} q_k \end{pmatrix} \right\rangle \right| = \left| \left\langle \begin{pmatrix} a p_k \\ p_k \end{pmatrix}, \begin{pmatrix} q_k \\ b q_k \end{pmatrix} \right\rangle \right| = |(a + \bar{b})p_k^T q_k|. \quad (20)$$

Considering again the relation (18), we have $\det(B_k) = 0$ if and only if the complex vectors

$$\begin{pmatrix} a p_k \\ p_k \end{pmatrix}, \quad \begin{pmatrix} q_k \\ b q_k \end{pmatrix}$$

are linearly dependent (with a and b defined by (19)). Thus, relations (20) and (18) imply:

$$|p_k^T q_k| = \frac{\left| \left\langle \begin{pmatrix} a p_k \\ p_k \end{pmatrix}, \begin{pmatrix} q_k \\ b q_k \end{pmatrix} \right\rangle \right|}{|a + \bar{b}|} = \frac{\left\| \begin{pmatrix} a p_k \\ p_k \end{pmatrix} \right\| \cdot \left\| \begin{pmatrix} q_k \\ b q_k \end{pmatrix} \right\|}{|a + \bar{b}|}$$

and performing the calculation we obtain:

$$|p_k^T q_k| = (a\bar{a} + 1)^{1/2} \|p_k\| (b\bar{b} + 1)^{1/2} \|q_k\| / |a + \bar{b}|.$$

If we denote with φ_k the angle between vectors p_k and q_k , we obtain:

$$|\cos \varphi_k| \triangleq \frac{|p_k^T q_k|}{\|p_k\| \|q_k\|} = \frac{[(a\bar{a} + 1)(b\bar{b} + 1)]^{1/2}}{|a + \bar{b}|} = \frac{[(|a|^2 + 1)(|b|^2 + 1)]^{1/2}}{|a + \bar{b}|},$$

and from (19)

$$|\cos \varphi_k| = 1/\sqrt{1 - \sigma_k}, \quad -1 \leq \sigma_k \leq -1/2.$$

Finally, by considering all the feasible values for $\cos \varphi_k$ we have either:

$$\begin{aligned} \cos \varphi_k &= +1/\sqrt{1 - \sigma_k}, & -1 \leq \sigma_k \leq -1/2 \\ \cos \varphi_k &= -1/\sqrt{1 - \sigma_k}, & -1 \leq \sigma_k \leq -1/2. \end{aligned}$$

Therefore, we summarize the latter result with the following proposition:

Proposition 2.3. Let φ_k be the angle between vectors p_k and q_k at step k_B of algorithm FLR. Suppose the test on d_k is performed with ϵ_k according to (12); then, the angle φ_k verifies one of the following bounds:

$$\begin{aligned} \sqrt{2/3} &\geq \cos \varphi_k \geq 1/\sqrt{2} \\ -1/\sqrt{2} &\geq \cos \varphi_k \geq -\sqrt{2/3}. \end{aligned} \quad (21)$$

□

This implies that, as long as ϵ_k is chosen according to (12), at the k_B -th planar step the directions p_k and q_k are *linearly independent*. Furthermore, observe that when at step k we perform the test (12), the direction q_k is not available. However, the computation of q_k is a straightforward combination of the vector Ap_k with either p_{k-1} (if the previous step was step $(k-1)_A$) or p_{k-2}, q_{k-2} (if the previous step was step $(k-2)_B$). These vectors are all available at step k , therefore no further calculation is required in performing the test (12), for vector q_k .

2.2.2. Angles among the directions belonging to different steps

Here we want to prove that the directions t_1, \dots, t_k generated by the algorithm FLR up to step k_A or $(k-1)_B, k \leq n$, are *uniformly linearly independent*. In particular we accomplish the result evaluating an estimate for the angles formed by directions t_1, \dots, t_k . Suppose the set of directions $\{t_1, \dots, t_k\}, k \leq n$, was generated by algorithm FLR and at step $i < k$ the parameter ϵ_i is chosen according to relation (12). Three possible cases may be considered:

1. Directions $t_i \equiv p_i$ and $t_j, i \neq j \leq k$, are respectively used inside steps i_A and either step j_A or j_B of Algorithm FLR⁷, thus from Theorem 2.1

$$p_i^T A t_j = 0. \quad (22)$$

Now, consider that at step i_A of algorithm FLR either

$$p_i^T A p_i > \epsilon_i \|p_i\|^2 \quad (23)$$

or

$$p_i^T A p_i < -\epsilon_i \|p_i\|^2. \quad (24)$$

From (23) we derive the following result:

$$\cos(\widehat{Ap_i, p_i}) \triangleq \frac{(Ap_i)^T p_i}{\|Ap_i\| \|p_i\|} > \frac{\epsilon_i \|p_i\|^2}{\lambda_M \|p_i\|^2} = \epsilon_i / \lambda_M,$$

⁷We remark that if direction t_j is used inside step j_B , then the results that we obtain in this item hold for direction t_{j+1} too.

and a similar conclusion holds for relation (24) too. Thus from relation (22) and (23) we get

$$\pi/2 - \arccos(\epsilon_i/\lambda_M) \leq |\widehat{p_i, t_j}| \leq \pi/2 + \arccos(\epsilon_i/\lambda_M), \quad (25)$$

while from relation (22) and (24) we obtain likewise

$$\pi/2 - \arccos(\epsilon_i/\lambda_M) \leq |\widehat{p_i, t_j}| \leq \pi/2 + \arccos(\epsilon_i/\lambda_M). \quad (26)$$

2. Directions $t_i \equiv p_i$ and t_j , $j < i \leq k$ (the same results hold for t_{j+1} too), are respectively used inside step i_B and step j_B of algorithm FLR, thus $p_i^T A t_j = 0$. From Theorem 2.1 we have $r_i^T p_i = \|r_i\|^2 = \cos(r_i, p_i) \|r_i\| \|p_i\|$, thus

$$\cos(r_i, p_i) = \|r_i\| / \|p_i\|. \quad (27)$$

Since we performed the i_B -th planar step, it is $t_i = p_i = r_i + \hat{b}_{i-2}(d_{i-2}q_{i-2} - \delta_{i-2}p_{i-2})/\Delta_{i-2}$ (see Table 1), with $\hat{b}_{i-2} = -q_{i-2}^T A r_i$, or $p_i = r_i + b_{i-1}p_{i-1}$, with $b_{i-1} = -p_{i-1}^T A r_i / p_{i-1}^T A p_{i-1}$.

- In the first case it is straightforwardly seen that $\hat{b}_{i-2} = -q_{i-2}^T A r_i = (A p_{i-2})^T A r_i$. Furthermore if we consider for ϵ_{i-2} the relation (12) and for Δ_{i-2} the relation (16) we have:

$$\begin{aligned} \|p_i\| &\leq \|r_i\| + \left\| \hat{b}_{i-2}(d_{i-2}q_{i-2} - \delta_{i-2}p_{i-2})/\Delta_{i-2} \right\| \\ &\leq \|r_i\| + \left| \hat{b}_{i-2}/\Delta_{i-2} \right| \|d_{i-2}q_{i-2} - \|A p_{i-2}\|^2 p_{i-2}\| \\ &\leq \|r_i\| + \frac{|(A p_{i-2})^T A r_i| [\|d_{i-2}\| \|q_{i-2}\| + \lambda_M^2 \|p_{i-2}\|^3]}{|(p_{i-2}^T A p_{i-2})(q_{i-2}^T A q_{i-2}) - \|A p_{i-2}\|^4|} \\ &\leq \|r_i\| \left[1 + \frac{\lambda_M^2 \|p_{i-2}\| [\epsilon_{i-2} \|p_{i-2}\|^2 \|q_{i-2}\| + \lambda_M^2 \|p_{i-2}\|^3]}{|\|A p_{i-2}\|^4 - (p_{i-2}^T A p_{i-2})(q_{i-2}^T A q_{i-2})|} \right] \\ &\leq \|r_i\| \left[1 + \frac{\lambda_M^2 \|p_{i-2}\| [\lambda_M^2 \|p_{i-2}\|^3 + \lambda_M^2 \|p_{i-2}\|^3]}{1/2 \lambda_m^4 \|p_{i-2}\|^4} \right] \\ &\leq \|r_i\| [1 + 4(\lambda_M/\lambda_m)^4]. \end{aligned}$$

Hence, the choice (12), the previous relation and relation (27) yield

$$\cos(r_i, p_i) \geq \lambda_m^4 / (\lambda_m^4 + 4\lambda_M^4).$$

Finally, since $r_i^T t_j = 0$ (and $r_i^T t_{j+1} = 0$) we simply have the final relation

$$\pi/2 - \arccos\left[\frac{\lambda_m^4}{\lambda_m^4 + 4\lambda_M^4}\right] \leq |\widehat{p_i, t_j}| \leq \pi/2 + \arccos\left[\frac{\lambda_m^4}{\lambda_m^4 + 4\lambda_M^4}\right]. \quad (28)$$

- In the second case, with a similar reasoning we obtain

$$\|r_i\| / \|p_i\| \geq \epsilon_{i-1} / (\epsilon_{i-1} + \lambda_M) \implies \cos(r_i, p_i) \geq \epsilon_{i-1} / (\epsilon_{i-1} + \lambda_M).$$

and a relation similar to (28) holds.

3. Directions $t_{i+1} \equiv q_i$ and t_j (or t_{j+1}) are respectively used inside steps i_B and j_B of algorithm FLR. We already know that $q_i^T A t_j = 0$ and, in order to estimate the angle $\widehat{q_i, t_j}$, we consider relation $q_i^T A p_i = \cos(q_i, A p_i) \|q_i\| \|A p_i\|$. From the expression of q_i in Algorithm FLR and Theorem 2.1 we have $q_i^T A p_i = \|A p_i\|^2$, thus $\cos(q_i, A p_i) = \|A p_i\| / \|q_i\| \geq \lambda_m \|p_i\| / \|q_i\|$. Finally, from (12) we retrieve the expression of $\|p_i\| / \|q_i\|$ and we obtain

$$\cos(q_i, A p_i) \geq \min \left\{ \bar{\epsilon} / \lambda_M^2, \sqrt{2\bar{\epsilon}\lambda_M} / \lambda_m^2 \right\}, \quad (29)$$

hence, since $(A p_i)^T t_j = 0$ (and $(A p_i)^T t_{j+1} = 0$) we simply have the final relation

$$\frac{\pi}{2} - \arccos \left[\min \left\{ \frac{\bar{\epsilon}}{\lambda_M^2}, \frac{\sqrt{2\bar{\epsilon}\lambda_M}}{\lambda_m^2} \right\} \right] \leq |\widehat{q_i, t_j}| \leq \frac{\pi}{2} + \arccos \left[\min \left\{ \frac{\bar{\epsilon}}{\lambda_M^2}, \frac{\sqrt{2\bar{\epsilon}\lambda_M}}{\lambda_m^2} \right\} \right]. \quad (30)$$

Taking into account relations (21), (25), (26), (28) and (30) we have the following result, that summarizes the contents of the last two sections:

Proposition 2.4. Let $\{t_1, \dots, t_h\}$, $h \leq n$, be the vectors (defined in Section 2.1) generated by Algorithm FLR. Suppose at step k of Algorithm FLR the test on d_k is performed with ϵ_k according to (12). Then, the directions $\{t_1, \dots, t_h\}$ are uniformly linearly independent.

We complete this section by observing that the test (13) is inexpensive inasmuch as all the quantities it contains were already calculated at the general k -th step. In addition consider that $\|p_k\| > \|r_k\|$; thus, the bounds (12) on ϵ_k may become unreliable only in case of large ill-conditioning of the matrix A .

In order to appreciate the conclusion of Proposition 2.4, observe that in case exactly n directions t_i , $i = 1, \dots, n$ are generated by the algorithm FLR, for the matrix A the following factorization holds

$$A = P^T B P, \quad P = [t_1 \cdots t_n],$$

where the matrix B has the expression

$$B = \text{diag}_{i \leq n} \{B_i\},$$

$$B_i = \begin{cases} p_i^T A p_i & \text{if the step } i \text{ is the step } i_A \\ \begin{pmatrix} p_i^T A p_i & p_i^T A q_i \\ q_i^T A p_i & q_i^T A q_i \end{pmatrix} & \text{if the step } i \text{ is the step } i_B. \end{cases}$$

Thus, whenever the directions $\{t_1, \dots, t_n\}$ are uniformly linearly independent, the algorithm FLR has explored the Krylov subspace $\mathcal{K}_n(r_1, A) \equiv \mathbb{R}^n$ and the condition number $\kappa(P) = \|P\| \|P^{-1}\|$ of matrix P can be suitably bounded.

3. Conclusions and Perspectives

In this paper we have proposed a new CG-type method for the iterative solution of large-scale indefinite linear systems. One of the remarkable features of the scheme, is the capability of exploiting the negative eigenspaces of the indefinite matrix A in (1). This avoids the well known premature

stopping of CG algorithm in the indefinite case.

A complete comparison between the planar CG methods and the other Krylov algorithms for indefinite linear systems will be investigated in future works.

Although Algorithm FLR in Table 1 was conceived for being embedded in an optimization framework, we also tested it as a solver of symmetric indefinite linear systems. In particular we considered the solution of linear system (1) with $n = 500$, where we assigned both the condition number ($cond$) and the clustering of the eigenvalues of the matrix A . More specifically, we randomly generated the symmetric nonsingular matrix A with the same number ($n/2$) of positive (λ_i^+ , $i = 1, \dots, n/2$) and negative (λ_j^- , $j = 1, \dots, n/2$) eigenvalues. We assigned the condition number $cond$, by means of imposing $0 < \lambda_1^+ \leq \lambda_i^+ \leq \lambda_{n/2}^+$ and $0 > \lambda_1^- \geq \lambda_j^- \geq \lambda_{n/2}^-$, with $\lambda_m = \lambda_1^+ = |\lambda_1^-| = 1$ and $\lambda_M = \lambda_{n/2}^+ = |\lambda_{n/2}^-| = \exp(cond)$. Furtherly, we introduced a clustering for the eigenvalues λ_i^+ , $i = 2, \dots, (n/2 - 1)$ and λ_j^- , $j = 2, \dots, (n/2 - 1)$, around either λ_1^+ (and respectively λ_1^-) or $\lambda_{n/2}^+$ (and respectively $\lambda_{n/2}^-$). The results are reported in Table 2, where \bar{x} is the solution detected, x^* is the known solution of the linear system, $r = b - A\bar{x}$, $iter$ is the number of iterations, Pla the number of planar steps and with $frac \in (0, 1)$ we control the clustering. Each row figures out the average results over 20 instances independently generated.

We highlight that the versatility of iterative methods in investigating the solution of indefinite problems, induces to conjecture that the application of the proposed new algorithm, might be specifically fruitful when used within optimization frameworks. In particular, this holds whenever we consider either “highly” nonlinear and/or nonconvex problems, where the overall optimization method often requires the use of *negative curvatures* (see Refs. 11, 13, 30, 31, 12) and the CG is definitely ineffective.

To this end a preliminary numerical experience in applying Algorithm FLR, is provided in Part 2. Further results will be provided in forthcoming papers, where the case of *singular matrix* A will be considered too. Finally, it seems still necessary to give a full evidence that our approach may be competitive with other algorithms in the literature. Indeed, the clear identification of those problems where the planar methods might be preferable, is under investigation.

Table 2: Algorithm FLR as a solver of nonsingular indefinite linear systems.

		$\lambda_i^+ - \lambda_m \leq frac(\lambda_M - \lambda_m)$				$\lambda_M - \lambda_i^+ \leq frac(\lambda_M - \lambda_m)$			
		$ \lambda_j^- - \lambda_m \leq frac(\lambda_M - \lambda_m)$				$\lambda_M - \lambda_j^- \leq frac(\lambda_M - \lambda_m)$			
<i>cond</i>	$\ \bar{x} - x^*\ $	$\ r\ /\ r_1\ $	<i>iter</i>	<i>Pla</i>	$\ \bar{x} - x^*\ $	$\ r\ /\ r_1\ $	<i>iter</i>	<i>Pla</i>	
<i>frac = 1.0</i>									
exp(0)	0.739E-15	0.413E-12	2.0	0.0	0.113E-14	0.622E-12	1.9	0.1	
exp(2)	0.885E-08	0.106E-05	93.7	0.8	0.860E-08	0.103E-05	93.5	0.5	
exp(4)	0.891E-08	0.152E-06	527.5	0.8	0.898E-08	0.156E-06	520.6	1.2	
exp(6)	0.855E-08	0.204E-07	1216.1	1.6	0.872E-08	0.207E-07	1344.1	1.4	
exp(8)	0.471E-07	0.156E-07	1635.0	1.0	0.958E-04	0.296E-04	1982.2	1.6	
exp(10)	0.119E-03	0.530E-05	2510.4	3.8	0.368E-07	0.153E-08	1664.6	0.2	
<i>frac = 0.8</i>									
exp(0)	0.181E-14	0.975E-12	1.9	0.1	0.175E-14	0.953E-12	1.9	0.1	
exp(2)	0.830E-08	0.118E-05	80.5	0.5	0.811E-08	0.881E-06	52.0	0.1	
exp(4)	0.906E-08	0.195E-06	480.4	0.9	0.800E-08	0.125E-06	98.5	0.1	
exp(6)	0.872E-08	0.256E-07	983.1	0.6	0.815E-08	0.172E-07	129.8	0.1	
exp(8)	0.475E-03	0.195E-03	2345.1	3.3	0.818E-08	0.234E-08	170.9	0.1	
exp(10)	0.186E-03	0.949E-05	1969.3	1.0	0.745E-08	0.284E-09	184.6	0.0	
<i>frac = 0.6</i>									
exp(0)	0.954E-15	0.527E-12	1.9	0.1	0.189E-14	0.104E-11	1.9	0.1	
exp(2)	0.790E-08	0.138E-05	65.9	0.5	0.461E-08	0.459E-06	35.7	0.1	
exp(4)	0.880E-08	0.245E-06	389.4	0.7	0.741E-08	0.102E-06	58.9	0.1	
exp(6)	0.888E-08	0.339E-07	1553.4	2.4	0.602E-08	0.114E-07	69.6	0.0	
exp(8)	0.669E-05	0.347E-05	2400.2	2.4	0.646E-08	0.166E-08	83.4	0.0	
exp(10)	0.112E-03	0.850E-05	1726.5	0.2	0.621E-08	0.215E-09	97.3	0.0	
<i>frac = 0.4</i>									
exp(0)	0.334E-14	0.186E-11	1.9	0.1	0.783E-15	0.432E-12	2.0	0.0	
exp(2)	0.775E-08	0.175E-05	50.4	0.3	0.275E-08	0.244E-06	26.0	0.0	
exp(4)	0.896E-08	0.360E-06	298.4	0.5	0.540E-08	0.666E-07	36.0	0.1	
exp(6)	0.868E-08	0.490E-07	1044.8	1.1	0.360E-08	0.608E-08	45.7	0.0	
exp(8)	0.143E-02	0.113E-02	2199.4	2.5	0.411E-08	0.935E-09	54.6	0.0	
exp(10)	0.256E-04	0.287E-05	1773.5	1.0	0.629E-08	0.192E-09	90.0	0.0	
<i>frac = 0.2</i>									
exp(0)	0.111E-14	0.614E-12	1.9	0.1	0.320E-14	0.176E-11	2.0	0.0	
exp(2)	0.528E-08	0.167E-05	32.2	0.1	0.402E-08	0.324E-06	17.9	0.1	
exp(4)	0.913E-08	0.649E-06	171.2	0.2	0.610E-08	0.679E-07	24.0	0.1	
exp(6)	0.854E-08	0.884E-07	842.4	0.8	0.401E-08	0.600E-08	30.0	0.0	
exp(8)	0.177E-01	0.260E-01	2002.5	2.6	0.322E-01	0.670E-02	284.2	0.0	
exp(10)	0.337E-07	0.604E-08	1778.2	0.7	0.275E-08	0.767E-10	41.6	0.0	

References

1. HESTENES, M.R., and STIEFEL, E., *Methods of Conjugate Gradients for Solving Linear Systems*, Journal of Research of the National Bureau of Standards, Vol. 49 B, pp. 409-436, 1952.
2. FASANO, G., *Planar-Conjugate Gradient Algorithm for Large-Scale Unconstrained Optimization, Part 2: Application*, submitted to Journal of Optimization Theory and Applications.
3. FREUND, R.W., GOLUB, G.H., and NACHTIGAL, N.M., *Iterative Solution of Linear Systems*, Acta Numerica, pp. 1-44, 1992.
4. BUNCH, J.R., and PARLETT, B.N., *Direct Methods for Solving Symmetric Indefinite Systems of Linear Equations*, SIAM Journal on Numerical Analysis, Vol. 8, pp. 639-655, 1971.
5. SAAD, Y., and VAN DER VORST, H.A., *Iterative Solution of Linear Systems in the 20th Century*, Journal on Computational and Applied Mathematics, Vol. 123, pp. 1-33, 2000.
6. GOLUB, G.H., and VAN DER VORST, H.A., *Closer to the Solution: Iterative Linear Solvers*, The State of the Art in Numerical Analysis, Edited by I.S.Duff and G.A.Watson, Clarendon Press, Oxford, UK, pp. 63-92, 1997.
7. SLEIJPEN, G.L.G., VAN DER VORST, H.A., and MODERSITZKI, J., *Differences in the Effects of Rounding Errors in Krylov Solvers for Symmetric Indefinite Linear Systems*, SIAM Journal on Matrix Analysis and Applications, Vol. 3, pp. 726-751, 2000.
8. VAN DER VORST, H.A., and CHAN, T.F., *Linear System Solvers: Sparse-Iterative Methods*, Parallel Numerical Algorithms, ICASE/LaRC Interdisciplinary Series in Science and Engineering, Edited by D.E.Keyes, A.Samed and V.Venkatakrishnan, Dodrecht, Kluwer Academic, Dodrecht, Holland, Vol. 4, pp. 91-118, 1997.
9. SLEIJPEN, G.L.G., and VAN DER VORST, H.A., *Krylov Subspace Methods for Large Linear Systems of Equations*, Preprint 803, Department of Mathematics, University of Utrecht, Utrecht, Holland, 1993.
10. ORTEGA, J.M., and RHEINBOLDT, W.C., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, NY, 1970.
11. NASH, S.G., *A Survey of Truncated-Newton Methods*, Journal of Computational and Applied Mathematics, Vol. 124, pp. 45-59, 1999.
12. MORE', J.J., and SORENSEN, D.C., *On the Use of Directions of Negative Curvature in a Modified Newton Method*, Mathematical Programming, Vol. 16, pp. 1-20, 1979.
13. MCCORMICK, G.P., *A Modification of Armijo's Stepsize Rule for Negative Curvature*, Mathematical Programming, Vol. 13, pp. 111-115, 1977.
14. BERTSEKAS, D.P., *Nonlinear Programming*, Athena Scientific, Belmont, Massachusetts, 1995.
15. BONGARTZ, I., CONN, A.R., GOULD, N., AND TOINT, PH.L., *CUTE: Constrained and Unconstrained Test Environment*, ACM Transactions on Mathematical Software, Vol. 21, pp. 123-160, 1995.

16. FLETCHER, R., and REEVES, C.M., *Function Minimization by Conjugate Gradients*, Computer Journal, Vol. 7, pp. 149-154, 1964.
17. POLAK, E., and RIBIERE, G., *Note sur la Convergence de Methodes de Directions Conjugées*, Revue Francaise d'Informatique et de Recherche Operationelle, Vol. 16, pp. 35-43, 1969.
18. FLETCHER, R., *Conjugate Gradient Methods for Indefinite Systems*, Proceedings of the Dundee Biennial Conference on Numerical Analysis, Edited by G.A.Watson, Springer, Berlin, Germany, pp. 73-89, 1975.
19. PAIGE, C.C., and SAUNDERS, M.A., *Solution of Sparse Indefinite Systems of Linear Equations*, SIAM Journal on Numerical Analysis, Vol. 12, pp. 617-629, 1975.
20. CULLUM, J.K., and WILLOUGHBY, R.A., *Lanczos Algorithm for Large Symmetric Eigenvalue Computations*, Birkhauser, Boston, Massachusetts, 1985.
21. HANSEN, P.C., *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, Pennsylvania, 1998.
22. HESTENES, M.R., *Conjugate Direction Methods in Optimization*, Springer Verlag, New York, NY, 1980.
23. LUENBERGER, D.G., *Hyperbolic Pairs in the Method of Conjugate Gradients*, SIAM Journal on Applied Mathematics, Vol. 17, pp. 1263-1267, 1969.
24. FASANO, G., *Use of Conjugate Directions inside Newton-Type Algorithms for Large Scale Unconstrained Optimization*, PhD Dissertation, Rome, Italy, 2001.
25. HU, Y.F., and STOREY, C., *Efficient Generalized Conjugate Gradient Algorithms, Part 2: Implementation*, Journal of Optimization Theory and Applications, Vol. 69, pp. 139-152, 1991.
26. 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.
27. 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.
28. 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.
29. GREENBAUM, A., *Iterative Methods for Solving Linear Systems*, Frontiers in Applied Mathematics, SIAM, Philadelphia, Pennsylvania, 1997.
30. GOULD, N.I.M., LUCIDI, S., ROMA, M., and TOINT, PH.L., *Exploiting Negative Curvature Directions in Line Search Methods for Unconstrained Optimization*, Optimization Methods and Software, Vol. 14, pp. 75-98, 2000.
31. LUCIDI, S., and ROMA, M., *Numerical Experiences with Truncated Newton Methods in Large-Scale Unconstrained Optimization*, Computational Optimization and Applications, Vol. 7, pp. 71-87, 1997.