

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 present 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 methods, planar conjugate gradient methods, indefinite matrices.

1. Introduction

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

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

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

²The author acknowledges Luigi Grippo and Stefano Lucidi, who contributed considerably to the elaboration of this paper. The exchange of experiences with Massimo Roma was a constant help in the investigation. 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.

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

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

The versatility of iterative algorithms solving the large-scale problem (1) suggests their natural embedding within optimization schemes. 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 the Newton method (Ref. 10) for calculating efficiently the direction d_k which solves the 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 the nonconvex function $f(x)$ calculated at the current point x_k . The convergence of the optimization method is affected strongly 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 out to be worthlessly expensive. Thus, a reliable but simple iterative algorithm, coping with the case where $\nabla^2 f(x_k)$ is indefinite, would be highly desirable as a solver for the linear system (2). Unfortunately, the solution of (2) might be 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 convergence to a maximum point. In a line search approach, the latter result may be achieved by means of proper application of the iterative method solving the large-scale system (2). One option is using an iterative method providing a pair of directions, say d_k and s_k , with the following purposes:

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

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

where $\lambda_m[\nabla^2 f(x_k)]$ is the smallest eigenvalues of $\nabla^2 f(x_k)$. The direction s_k has the specific purpose of forcing the convergence of the optimization method toward the point x^* which satisfies the second-order necessary optimality conditions; i.e.,

$$\nabla f(x^*) = 0 \text{ and } s^T \nabla^2 f(x^*) s \geq 0, \quad \forall s \in \mathbb{R}^n.$$

Under the mild assumptions (Ref. 13), it can be proved that, replacing the scheme

$$x_{k+1} = x_k + d_k$$

with the scheme

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

with suitable $\alpha_k, \beta_k \in \mathbb{R}$, is efficient and effective for convergence toward a solution point 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 the Newton method toward a maximum point is by means of a so called modified Newton method (Ref. 14), which is globally convergent modification of the Newton method, in the case where 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 concavity regions of $f(x)$ near x_k . In a related paper (Ref. 2), we shall consider a modified Newton method, which uses the planar CG algorithm proposed in this paper, within a line search framework. We shall give evidence of its goodness by solving several problems of the CUTE collection (Ref. 15).

The method that we propose here is an extension of the CG method, which is an example of a simple and appealing iterative method that can

be used as long as $\nabla^2 f(x_k)$ is positive definite. The CG method was proposed originally 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 cannot be the method of choice.

Some attempts for overcoming the above 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 these algorithms into the following two classes (Ref. 21):

- (i) Ritz-Galerkin Class. This includes those methods which provide at step k the new residual $r_k = b - Ax_k$, in such a way that

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

- (ii) Minimal Residual Class. At step k , the solution x_k satisfies

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

In this paper we focus on an iterative algorithm in Ritz-Galerkin class, which retains the low overall computational cost of CG vis-a-vis the Lanczos, MINRES, GMRES algorithms and maintains a satisfactory exactness when applied to solving problem (1). More precisely, we consider the category of planar methods (see Refs. 22–24 and 25–28); the rationale behind these methods may be roughly summarized as follows. Let A be indefinite and nonsingular, and suppose that 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 other hand, planar CG methods generate a second direction q_k ; 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

$$p_i^T A p_{k+2} = q_i^T A p_{k+2} = 0, \quad i \leq k;$$

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 (Algorithm Hes) generates the vector q_k as follows: at step k , the pair $\{p_k, q_k\}$ is calculated, where the expression of q_k is such that

$$q_k^T A p_i = q_k^T A q_j = 0, \quad \text{with } i, j \leq k - 1.$$

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 other hand, the algorithms in Ref. 23 (Algorithm Lue) and Ref. 24 (Algorithm Fas), provide the second direction q_k if and only if relation

$$p_k^T A p_k = 0 \tag{4}$$

holds. Consequently, in case $0 < |p_k^T A p_k| < \varepsilon_k$, $k < n$, and ε_k is a small number, 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 provides usually 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 the Hestenes method, we can obtain the coefficients of Algorithms Lue and Fas. Therefore, the latter algorithm can be interpreted as simplification of the former. On this stream, the present paper is concerned with introducing and developing an iterative algorithm solving problem (1), which recovers both the general structure of Algorithm Hes and the low computational cost of Algorithms Lue and Fas.

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

Section 2 introduces the new proposed algorithm where we suppose that the matrix A is indefinite and nonsingular; the convergence properties are pointed out in Section 2.1, where an analysis of global convergence is carried out. Section 2.2 deals with particular features of the directions generated by our algorithm. Section 3 contains conclusions and perspectives.

2. New Planar Algorithm

As observed already in Section 1, 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 affect seriously the behavior of the algorithms. For Algorithms Lue and Fas, the test simply attempts to verify whether $p_k^T A p_k = 0$. Thus, when $p_k^T A p_k$ is small, but not exactly zero, the application of these algorithms may involve numerical approximations. This shortcoming turns out to be less relevant for Algorithm Hes, since at step k the test on the quantity

$$\Delta_k = \left(p_k^T A p_k \right) \left(q_k^T A q_k \right) - \left(p_k^T A q_k \right)^2$$

i.e.,

$$|\Delta_k| \leq \epsilon_k \left(p_k^T A q_k \right)^2, \quad \epsilon_k = 1/2,$$

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, for the planar algorithms investigated, there is the following tradeoff: if the internal test is computationally cumbersome (e.g. Algorithm Hes with respect to Algorithms Lue and Fas), the algorithm is less sensitive to numerical approximations. Of course, this property holds within each iteration; thus, no final conclusion can be argued a priori on the overall behavior of the algorithms.

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

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

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

2.1. Convergence Results. The following results can be established for Algorithm FLR described below. We introduce the following convention, which will be used in the proofs, in order to simplify the treatment

($i \leq n$):

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

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 . □

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

Algorithm FLR. This algorithm solves 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 $k = k + 1$ and go to Step k .
- Step k_B . If $k = 1$, then set $q_k = A p_k$.
If $k > 1$ and the previous step is $(k - 1)_A$, then set $\beta_{k-1} = -(A p_{k-1})^T A p_k / d_{k-1}$ and $q_k = A p_k + \beta_{k-1} p_{k-1}$.
If $k > 1$ and the previous step is $(k - 2)_B$, then set $\hat{\beta}_{k-2} = -(A q_{k-2})^T A p_k$ and $q_k = A p_k + \hat{\beta}_{k-2} (d_{k-2} q_{k-2} - \delta_{k-2} p_{k-2}) / \Delta_{k-2}$.
Compute $c_k = r_k^T p_k$, $\delta_k = p_k^T A q_k$, $e_k = q_k^T A q_k$, $\Delta_k = d_k e_k - \delta_k^2$.
Compute $\hat{c}_k = (c_k e_k - \delta_k q_k^T r_k) / \Delta_k$, $\hat{d}_k = (d_k q_k^T r_k - \delta_k c_k) / \Delta_k$.
Set $x_{k+2} = x_k + \hat{c}_k p_k + \hat{d}_k q_k$, $r_{k+2} = r_k - \hat{c}_k A p_k - \hat{d}_k A q_k$.
If $r_{k+2} = 0$, then stop. Else, compute $\hat{b}_k = -q_k^T A r_{k+2}$ and set $p_{k+2} = r_{k+2} + \hat{b}_k (d_k q_k - \delta_k p_k) / \Delta_k$. Set $k = k + 2$ and go to Step k .

Proof. See Ref. 29. □

The previous lemma reveals that, if matrix A is indefinite and non-singular, Algorithm FLR can perform always 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 will not stop.

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

$$At_k \in \text{span}\{t_1, \dots, t_{k+1}\}, \tag{5}$$

and the following properties hold:

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

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

Proof. See Ref. 29. □

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

Proof. See Ref. 29. □

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 the linear system (1) in at most n steps.

Proof. At Step k , Algorithm FLR has generated already k linearly independent directions t_1, \dots, t_k ; thus $k \leq n$. In addition, suppose that the

algorithm stops at step m . Then, the point

$$x^* = x_1 + \sum_{i=1}^m \alpha_i t_i, \quad m \leq n, \tag{6}$$

is the solution of problem (1); i.e.,

$$r_1 = \sum_{i=1}^m \alpha_i A t_i.$$

Indeed, this follows from Theorem 2.1, after the multiplication of the relation (6) by means of either the vector $A p_i$ (Step i_A) or the vectors $A p_i, A q_i$ (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. \square

We remark that, like in the CG, in this planar scheme for each step we attempt to determine the solution of the 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 the 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 Algorithm FLR stops prematurely: this motivates a 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 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 between the directions that it generates.

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.

Proposition 2.1. At Step k_B of Algorithm FLR, the relation $\Delta_k = 0$ holds if and only if the 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

$$\tilde{A} = \begin{pmatrix} p_k & q_k \\ p_k & q_k \end{pmatrix}^T \begin{pmatrix} A/2 & 0 \\ 0 & A/2 \end{pmatrix} \begin{pmatrix} p_k & q_k \\ p_k & q_k \end{pmatrix} \in \mathbb{R}^{2 \times 2}; \tag{7}$$

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 the vectors p_k and q_k are parallel. \square

This result ensures that, if $\Delta_k \neq 0$ at Step k_B , then the 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):

$$-\delta_k^2 \leq \Delta_k \leq -\delta_k^2/2, \tag{8}$$

where

$$\Delta_k = d_k e_k - \delta_k^2,$$

with

$$d_k = p_k^T A p_k, \quad e_k = q_k^T A q_k, \quad \delta_k = p_k^T A q_k = \|A p_k\|^2.$$

On this purpose, let $\bar{\epsilon} > 0$ and suppose that p_k and q_k are both available at Step k . Then, set $\epsilon_k > 0$ according to the expression

$$\bar{\epsilon} \leq \epsilon_k \leq \min \left\{ \lambda_M^2 \|p_k\| / \|q_k\|, \lambda_m^4 \|p_k\|^2 / (2\lambda_M \|q_k\|^2) \right\}. \tag{9}$$

Proposition 2.2. Suppose that at Step k of Algorithm FLR, the test

$$|d_k| \leq \epsilon_k \|p_k\|^2 \tag{10}$$

holds, where ϵ_k is chosen according to (9). Then, at Step k_B of Algorithm FLR, the following relation holds:

$$-\delta_k^2 \leq \Delta_k \leq -\delta_k^2/2, \tag{11}$$

where

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

Proof. See Ref. 29. \square

Now, suppose that the relation (11) holds. Then, from Proposition 2.1 and the relation $\|p_k\| \geq \|r_k\| > 0$, the vectors p_k and q_k are not parallel. Moreover, Proposition 2.2 ensures that there exists a negative constant σ_k such that

$$\Delta_k = \sigma_k \delta_k^2, \quad -1 \leq \sigma_k \leq -1/2. \tag{12}$$

We shall prove now that the relation (12) implies a condition on the angle φ_k between the vectors p_k and q_k . On this purpose, we rearrange the relation (12) as

$$\Delta_k - \sigma_k \delta_k^2 = 0.$$

Then, similarly to (7), we 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 point out suitable conditions on the coefficients of B_k , by imposing $\Delta_k - \sigma_k \delta_k^2 = 0$ [i.e., the relation (12)]: 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:

$$\begin{aligned} 0 &= \Delta_k - \sigma_k \delta_k^2 \\ &= \left(p_k^T A p_k \right) \left(q_k^T A q_k \right) - (1 - \sigma_k) \delta_k^2 = \det(B_k) \\ &= \det \left[\begin{pmatrix} a p_k & q_k \\ p_k & b q_k \end{pmatrix}^T \begin{pmatrix} A/2 & \emptyset \\ \emptyset & A/2 \end{pmatrix} \begin{pmatrix} \bar{a} p_k & q_k \\ p_k & \bar{b} q_k \end{pmatrix} \right] \\ &= \det \left[\begin{matrix} (1/2)(a\bar{a} + 1) p_k^T A p_k & (1/2)(a + \bar{b}) p_k^T A q_k \\ (1/2)(\bar{a} + b) q_k^T A p_k & (1/2)(b\bar{b} + 1) q_k^T A q_k \end{matrix} \right]. \end{aligned} \tag{13}$$

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:

$$\left(|a|^2 + 1 \right) \left(|b|^2 + 1 \right) / 4 = 1, \tag{14a}$$

$$|a + \bar{b}|^2 / 4 = 1 - \sigma_k, \tag{14b}$$

which will be used later on. Now, observe that

$$\left| \left\langle \begin{pmatrix} \bar{a} p_k \\ p_k \end{pmatrix}, \begin{pmatrix} q_k \\ \bar{b} q_k \end{pmatrix} \right\rangle \right| = \left| \left\langle \begin{pmatrix} a p_k \\ p_k \end{pmatrix}, \begin{pmatrix} q_k \\ b q_k \end{pmatrix} \right\rangle \right| = \left| (a + \bar{b}) p_k^T q_k \right|. \tag{15}$$

Considering again the relation (13), we have

$$\det(B_k) = 0$$

if and only if the complex vectors

$$\begin{pmatrix} ap_k \\ p_k \end{pmatrix}, \begin{pmatrix} q_k \\ bq_k \end{pmatrix}$$

are linearly dependent, with a and b defined by (14). Thus, the relations (15) and (13) imply

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

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

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

and from (14),

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

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

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

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 that the test on d_k is performed with ϵ_k according to (9); then, the angle φ_k verifies one of the following bounds:

$$\sqrt{2/3} \geq \cos \varphi_k \geq 1/\sqrt{2}, \tag{16a}$$

$$-1\sqrt{2} \geq \cos \varphi_k \geq -\sqrt{2/3}. \tag{16b}$$

This implies that, as long as ϵ_k is chosen according to (9), 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 (9), 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 (9) for the vector q_k .

2.2.2. Angles among the Directions Belonging to Different Steps. Here, we prove that the directions t_1, \dots, t_k generated by 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 the directions t_1, \dots, t_k . Suppose that the set of directions $\{t_1, \dots, t_k\}, k \leq n$, was generated by Algorithm FLR and that at Step $i < k$, the parameter ϵ_i is chosen according to the relation (9). Three possible cases may be considered:

(C1) The directions $t_i \equiv p_i$ and $t_j, i \neq j \leq k$, are respectively used inside Step i_A and either Step j_A or j_B of Algorithm FLR⁴. Thus, from Theorem 2.1,

$$p_i^T A t_j = 0. \tag{17}$$

Now, consider that, at Step i_A of Algorithm FLR,

$$\text{either } p_i^T A p_i > \epsilon_i \|p_i\|^2, \tag{18}$$

$$\text{or } p_i^T A p_i < -\epsilon_i \|p_i\|^2. \tag{19}$$

From (18), we derive the following result:

$$\begin{aligned} \cos(\widehat{A p_i, p_i}) &\triangleq (A p_i)^T p_i / \|A p_i\| \|p_i\| \\ &> \epsilon_i \|p_i\|^2 / \lambda_M \|p_i\|^2 \\ &= \epsilon_i / \lambda_M, \end{aligned}$$

⁴We remark that if the direction t_j is used inside Step j_B , then the results that we obtain in this item hold for the direction t_{j+1} too.

and a similar conclusion holds for relation (19) too. Thus, from the relations (17) and (18), we get

$$\pi/2 - \arccos(\epsilon_i/\lambda_M) \leq |\widehat{p_i, t_j}| \leq \pi/2 + \arccos(\epsilon_i/\lambda_M), \tag{20}$$

while from the relations (17) and (19), we obtain likewise

$$\pi/2 - \arccos(\epsilon_i/\lambda_M) \leq |\widehat{p_i, t_j}| \leq \pi/2 + \arccos(\epsilon_i/\lambda_M). \tag{21}$$

(C2) The directions $t_i \equiv p_i$ and $t_j, j < i \leq k$ (the same results hold for t_{j+1} too), are used respectively 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,

$$\cos(r_i, p_i) = \|r_i\| / \|p_i\|. \tag{22}$$

Since we performed the i_B th planar step, we have that

$$\begin{aligned} \text{either } & t_i = p_i = r_i + \hat{b}_{i-2} (d_{i-2} q_{i-2} - \delta_{i-2} p_{i-2}) / \Delta_{i-2}, \\ & \text{with } \hat{b}_{i-2} = -q_{i-2}^T A r_i, \\ \text{or } & p_i = r_i + b_{i-1} p_{i-1}, \text{ with } b_{i-1} = -p_{i-1}^T A r_i / p_{i-1}^T A p_{i-1}. \end{aligned}$$

In the first case, it is 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 (9) and for Δ_{i-2} the relation (11), we have

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

Hence, the choice (9), the previous relation, and relation (22) 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

$$\begin{aligned} \pi/2 - \arccos \left[\lambda_m^4 / (\lambda_m^4 + 4\lambda_M^4) \right] &\leq |\widehat{p_i, t_j}| \\ &\leq \pi/2 + \arccos \left[\lambda_m^4 / (\lambda_m^4 + 4\lambda_M^4) \right]. \end{aligned} \tag{23}$$

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 (23) holds.

(C3) The directions $t_{i+1} \equiv q_i$ and t_j or t_{j+1} are used respectively inside Steps i_B and j_B of Algorithm FLR. We know already that $q_i^T A t_j = 0$ and, in order to estimate the angle $\widehat{q_i, t_j}$, we consider the 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 (9), we retrieve the expression of $\|p_i\|/\|q_i\|$ and we obtain

$$\cos(q_i, A p_i) \geq \min \left\{ \bar{\epsilon}/\lambda_M^2, \sqrt{2\bar{\epsilon}\lambda_M}/\lambda_m^2 \right\}; \tag{24}$$

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

$$\begin{aligned} \pi/2 - \arccos \left[\min \left\{ \bar{\epsilon}/\lambda_M^2, \sqrt{2\bar{\epsilon}\lambda_M}/\lambda_m^2 \right\} \right] \\ \leq |\widehat{q_i, t_j}| \leq \pi/2 + \arccos \left[\min \left\{ \bar{\epsilon}/\lambda_M^2, \sqrt{2\bar{\epsilon}\lambda_M}/\lambda_m^2 \right\} \right]. \end{aligned} \tag{25}$$

Taking into account the relations (16), (20), (21), (23), (25), we have the following result summarizing 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 that at Step k of Algorithm FLR, the test on d_k is performed with ϵ_k according to (9). Then, the directions $\{t_1, \dots, t_h\}$ are uniformly linearly independent.

We complete this section by observing that the test (10) is inexpensive inasmuch as all the quantities it contains are already calculated at the general k th step. In addition, consider that $\|p_k\| > \|r_k\|$; thus, the bounds (9) on ϵ_k may become unreliable only in the case of large ill-conditioning of the matrix A .

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

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

where the matrix B has the expression

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

with

$$B_i = p_i^T A p_i, \quad \text{if Step } i \text{ is Step } i_A,$$

$$B_i = \begin{pmatrix} p_i^T A p_i & p_i^T A q_i \\ q_i^T A p_i & q_i^T A q_i \end{pmatrix}, \quad \text{if Step } i \text{ is Step } i_B.$$

Thus, whenever the directions $\{t_1, \dots, t_n\}$ are uniformly linearly independent, 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 the 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 feature 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 a CG algorithm in the indefinite case.

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

Although Algorithm FLR was conceived as being embedded in an optimization framework, we tested it also as a solver of symmetric indefinite linear systems. In particular, we considered the solution of the linear system (1) with $n=500$, where we assigned both the condition number (cond) and the clustering of the eigenvalues of the matrix A . The results are reported in Ref. 29.

The versatility of iterative methods in investigating the solution of indefinite problems induces us 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 requires often the use of negative curvatures (see Refs. 28, 11, 13, 31, 32, 12) and the CG is definitely ineffective.

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

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. 49B, pp. 409–436, 1952.
2. FASANO, G., *Planar-Conjugate Gradient Algorithm for Large-Scale Unconstrained Optimization, Part 2: Application*, Journal of Optimization Theory and Applications, Vol. 125, pp. 543–558, 2005.
3. FREUND, R. W., GOLUB, G. H., and NACHTIGAL, N. M., *Iterative Solution of Linear Systems*, Acta Numerica, Vol. 1, pp. 57–100, 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, Kluwer Academic Publishers, Dordrecht, 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, 1983.
 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 Step Size 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, P. 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. PAIGEE, 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-Deficit and Discrete Ill-Posed Problems*, SIAM, Philadelphia, Pennsylvania, 1998.
 22. HESTENES, M. R., *Conjugate Direction Models 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. FASANO, G., *Planar-Conjugate Gradient Algorithm for Large-Scale Unconstrained Optimization, Part 1: Theory*, Technical Report 2004-015, Istituto Nazionale per Studi ed Esperienze di Architettura (INSEAN), Rome, Italy, 2004.
30. GREENBAUM, A., *Iterative Methods for Solving Linear Systems*, Frontiers in Applied Mathematics, SIAM, Philadelphia, Pennsylvania, 1997.
31. GOULD, N. I. M., LUCIDI, S., ROMA, M., and TOINT, P. L., *Exploiting Negative Curvature Directions in Line Search Methods for Unconstrained Optimization*, Optimization Methods and Software, Vol. 14, pp. 75–98, 2000.
32. 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.