INSEAN

National Research Center for Ships
and Marine Systems

INSEAN Technical Report N. 2007-002

# On the iterative computation of a $\ell_2$-norm scaling based preconditioner

Giovanni Fasano and Massimo Roma

# On the iterative computation of a $\ell_2$-norm scaling based preconditioner

GIOVANNI FASANO[1,2] and MASSIMO ROMA[2]

(1) Istituto Nazionale per Studi ed Esperienze di Architettura Navale INSEAN,
via di Vallerano, 139 – Roma, ITALY
E-mail: g.fasano@insean.it.

(2) Dipartimento di Informatica e Sistemistica "A. Ruberti",
Sapienza Università di Roma , via Buonarroti, 12 – Roma, ITALY
E-mail: (fasano,roma)@dis.uniroma1.it.

**Abstract**

In this paper we consider the Krylov subspace based method introduced in [Fasano, 2005a], for iteratively solving the symmetric and possibly indefinite linear system $Ax = b$. We emphasize the application of the latter method to compute a diagonal preconditioner. The approach proposed is based on the approximate computation of the $\ell_2$-norm of the rows (columns) of the matrix $A$ and on its use to equilibrate the matrix $A$. The distinguishing feature of this approach is that the computation of the $\ell_2$-norm is performed without requiring the knowledge of the entries of the matrix $A$ but only using a routine which provides the product of $A$ times a vector.

**Keywords:** Diagonal Preconditioning, Indefinite linear systems, Krylov subspace methods.

## 1 Introduction

In this paper we consider the solution of the symmetric (possibly) indefinite linear system

$$Ax = b \tag{1.1}$$

where $A$ is a $n \times n$ dense matrix and $n$ is "large". Krylov subspace methods are a broad class of iterative methods widely used for the solution of the system (1.1). They include the well known Conjugate Gradients (CG) method along with many variants. In particular, we focus on a modification of the standard Conjugate Gradient method, in order to cope also with the indefinite case. The Krylov subspace method we adopt can be used both for solving the linear system and for iteratively approximating the $\ell_2$-norm of the rows (columns) of the linear system matrix. The resulting approximation of the $\ell_2$-norm is then used to generate a diagonal preconditioner. The application of such a preconditioner corresponds to a scaling of the matrix, aiming at obtaining an equilibrated system.

It is well known that equilibrating the matrix of a linear system is of great importance (see e.g. [Duff et al., 1986, Schneider and Zenios, 1990, Ruiz, 2001]), especially when solving linear systems arising from real world applications, when the entries of the matrix may even vary of many orders of magnitude and the eigenvalues may be very spread. In general, to balance the matrix to have an equilibrated system is a difficult and computationally heavy task. Some algorithms have been proposed for equilibrating matrices in a specific norm (see e.g. [Bunch, 1971, Ruiz, 2001, Parlett and Landis, 1982]). Moreover, some optimality properties have been proved for the $\ell_p$-norm scaling, equilibrating the rows or the columns of a matrix for minimizing the condition number (see e.g. Section 7.3 of [Higham, 2002]). On the basis of these properties, an optimal diagonal scaling is obtain by using the Jacobi preconditioner, which enables to minimize the condition number in the $\ell_2$-norm of the transformed system, with respect to all the diagonal preconditioners. Jacobi preconditioner is simply defined by using as diagonal preconditioner the diagonal elements of the matrix itself, but it is impracticable in the large scale setting whenever the actual entries of the matrix are unknown.

Of course, diagonal preconditioning is a particular case of more general preconditioning techniques. The literature on general preconditioning linear systems is very broad and we refer, e.g. to [Greenbaum, 1997] for a survey. Here we focus on diagonal preconditioners since they enjoy the following good features:

- it is possible to construct a diagonal preconditioner without requiring the knowledge of the entries of the matrix $A$ but only by means of a routine, which provides us with the matrix–vector product of the matrix $A$ times a vector. This is a key point since this routine is already available in any implementation of a CG type algorithm, and it is usually the only available information on $A$;
- even if they carry out a simple scaling, they enable to greatly reduce the condition number of the matrix and they do not destroy the possible sparsity of the matrix;
- they require a minimal additional work to be implemented.

As far as the authors are aware, very few proposals satisfying these requirements exist. Recently, in [Roma, 2005] a diagonal preconditioner which satisfies these requirements has been proposed. It is based on the approximate $\ell_1$-norm scaling of the columns vectors and it is proved to be successful in the framework of truncated Newton methods for large scale unconstrained optimization. Other proposals base on diagonal scaling exist (see e.g. [Liu and Nocedal, 1989] and [Barlow and Toraldo, 1995] in the context of nonlinear optimization). Usually a preconditioning strategy requires the knowledge of the system matrix, therefore it often results impracticable in the large scale setting.

In this paper we concentrate on the $\ell_2$-norm scaling and we show how it is possible to approximate compute it by means of a particular Krylov subspace method. The paper is organized as follows: in Section 2 we briefly recall the algorithm FLR which is the Krylov subspace methods we adopt, along with some useful features. In Section 3 we describe its use for obtaining a tridiagonal decomposition of the system matrix. In Section 4 we describe the approximate computation of the $\ell_2$-norm which is at the basis of our preconditioner. We will denote by $\mathcal{K}_h(A,b) = \text{span}\{b, Ab, \ldots, A^{h-1}b\}$ the $h$-dimensional Krylov subspace.

## 2 The CG type algorithm FLR

The Krylov subspace method we use in this paper is the CG type method FLR introduced in [Fasano, 2005a]. It belongs to the class of Krylov subspace methods and, unlike the Conjugate Gradient, it copes with the indefinite case too. A complete scheme of the algorithm FLR is reported in Table 2.1. It is a "planar"–CG algorithm and is a modification of the stan-

## Algorithm FLR

```
Step 1: $k = 1$, $x_1 = 0$, $r_1 = b$.  If $r_1 = 0$ then STOP,
        else compute $p_1 = r_1$.

Step k: Compute $\sigma_k = p_k^T A p_k$.

    If $| \sigma_k | \geq \epsilon_k \|p_k\|^2$ then go to Step $k_A$ else go to Step $k_B$

      – Step $k_A$ (standard CG step) :
          Set $x_{k+1} = x_k + a_k p_k$,  $r_{k+1} = r_k - a_k A p_k$ ,  where  $a_k = \frac{r_k^T p_k}{\sigma_k} = \frac{\|r_k\|^2}{\sigma_k}$.
          If $r_{k+1} = 0$ then STOP
          else compute  $p_{k+1} = r_{k+1} + \beta_k p_k$ with  $\beta_k = \frac{-p_k^T A r_{k+1}}{\sigma_k} = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}$.
          Set $k = k + 1$ and go to Step $k$.

      – Step $k_B$ (planar CG step) :
          If $k = 1$ then compute the vector $q_k = A p_k$,
          else compute the vector
              $q_k = \begin{cases} A p_k + b_{k-1} p_{k-1}, & \text{if the previous step is } Step\ (k-1)_A \\ A p_k + \frac{\hat{b}_{k-2}}{\Delta_{k-2}} (\sigma_{k-2} q_{k-2} - \delta_{k-2} p_{k-2}), & \text{if the previous step is } Step\ (k-2)_B \end{cases}$
          where  $b_{k-1} = -(A p_{k-1})^T A p_k / \sigma_{k-1}$  and  $\hat{b}_{k-2} = -(A q_{k-2})^T A p_k$.
          Compute  $c_k = r_k^T p_k$,  $\delta_k = p_k^T A q_k$,  $e_k = q_k^T A q_k$,  $\Delta_k = \sigma_k e_k - \delta_k^2$
          and  $\hat{c}_k = (c_k e_k - \delta_k q_k^T r_k) / \Delta_k$,  $\hat{\sigma}_k = (\sigma_k q_k^T r_k - \delta_k c_k) / \Delta_k$.
          Set $x_{k+2} = x_k + \hat{c}_k p_k + \hat{\sigma}_k q_k$ ,  $r_{k+2} = r_k - \hat{c}_k A p_k - \hat{\sigma}_k A q_k$.
          If $r_{k+2} = 0$ then STOP
          else compute  $p_{k+2} = r_{k+2} + \frac{\hat{\beta}_k}{\Delta_k} (\sigma_k q_k - \delta_k p_k)$  with $\hat{\beta}_k = -q_k^T A r_{k+2}$.
          Set $k = k + 2$ and go to Step $k$.
```

Table 2.1: The Planar CG method FLR to solve the linear system $Ax = b$.

dard CG algorithm [Hestenes, 1980]. Further details on the FLR algorithm can be found in [Fasano, 2005a, Fasano, 2005b]; here we simply consider some useful results.

First observe that as long as at Step $k$ the planar CG *Step* $k_B$ is not performed, the FLR algorithm reduces to the standard CG and hence, at *Step* $k_A$ the algorithm detects the solution of (1.1) along the conjugate direction $p_k$. On the contrary, if a pivot breakdown occurs at Step $k$ (i.e. $p_k^T A p_k \approx 0$), the FLR algorithm generates another direction at *Step* $k_B$ (namely $q_k$). Then, it performs a search for the system solution on the 2-dimensional linear manifold $x_k + \text{span}\{p_k, q_k\}$, and generates the new point $x_{k+2}$. In addition it can be easily proved (see [Fasano, 2005a]) that, if the indefinite matrix $A$ is nonsingular and at Step $k$ we have $r_k \neq 0$, then the FLR algorithm can always perform either *Step* $k_A$ or *Step* $k_B$. As concerns the assessment of the parameter $\epsilon_k$ at the *Step* $k$, some proposals where considered in [Fasano, 2006, Fasano and Roma, 2005], in order to avoid possible instabilities.

## 3 Tridiagonalization via the FLR algorithm

In this section we will show how to obtain a tridiagonal decomposition of the system matrix $A$ via the FLR algorithm, which will be at the basis of the construction of the preconditioner. In particular, we will describe how, under very mild assumptions described in [Fasano, 2005a], after $h \leq n$ steps, the FLR algorithm provides $h$ orthogonal vectors $r_1, \ldots, r_h$, so that, if $r_i \neq 0$, $i = 1, \ldots, h$ and $r_{h+1} = 0$, the relation

$$AR_h = R_h T_h, \qquad h \leq n, \tag{3.1}$$

holds with $R_h \equiv \left( \frac{r_1}{\|r_1\|} \quad \cdots \quad \frac{r_h}{\|r_h\|} \right)$. Moreover, since $A$ is nonsingular, $T_h$ is a tridiagonal *irreducible* matrix. Furthermore, if $r_{h+1} \neq 0$ the FLR algorithm provides the relation

$$AR_h \; = \; R_h T_h \; + \; \rho_{h+1} \;\; r_{h+1} e_h^T, \qquad \text{for some } \rho_{h+1} \neq 0, \tag{3.2}$$

in place of relation (3.1), that is, the matrix equality (3.2) is obtained from (3.1) with the additional rank–one update $\rho_{h+1} r_{h+1} e_h^T$. Finally, the orthogonality of the vectors $r_1, \ldots, r_{h+1}$ and relation (3.1) yield

$$T_h = R_h^T A R_h. \tag{3.3}$$

When $h = n$, $R_h$ is an orthogonal matrix and (3.3) represents a factorization of matrix $A$ into a tridiagonal form. Furthermore, it is possible to define the *unit lower tridiagonal* matrix $L_h \in \mathbb{R}^{h \times h}$, and the *nonsingular* $2 \times 2$ block diagonal matrix $B_h \in \mathbb{R}^{h \times h}$, such that

$$T_h = L_h B_h L_h^T. \tag{3.4}$$

The aim of this paper is to use relations (3.1)-(3.4), computed via the Krylov subspace method FLR, in order to determine a preconditioner for solving indefinite linear systems.

We will show in the remainder of this section that the entries of the matrix $L_h$ and $B_h$ have very simple expressions, in terms of the coefficients of the FLR algorithm.

Let us consider the Algorithm FLR at a generic Step $k$ and let us introduce the following notation [Fasano and Roma, 2005]: if at Step $k$ of the FLR algorithm the condition $|p_k^T A p_k| \geq \epsilon_k \|p_k\|^2$ is satisfied, then we set $w_k = p_k$ (standard CG step); otherwise we set $w_k = p_k$ and $w_{k+1} = q_k$ (planar CG step). According with the latter positions, the sequence $\{w_i\}$ represents the sequence of directions generated by the Algorith FLR which contains, at most, pairs of consecutive non–conjugate directions.

Similarly, we consider the sequence $\{r_i\}$ of the residuals generated by the algorithm FLR up to Step $k$. However, in the planar CG Step $k_B$ we observe that two directions ($w_k$ and $w_{k+1}$) and only one residual $r_k$ are generated. Nevertheless, if the Step $k$ is the planar Step $k_B$, we can introduce a "dummy" residual $r_{k+1}$, which completes the sequence of orthogonal vectors $\{r_1, \ldots, r_k, r_{k+1}\}$ [Bank and Chan, 1994, Fasano, 2001]. We can soon realize [Fasano, 2001] that the possible choices for $r_{k+1}$ are:

$$r_{k+1} \;=\; \pm\left[\hat{\alpha}_k r_k + (1+\hat{\alpha}_k)\; sgn(\sigma_k) A p_k\right]\;, \qquad \hat{\alpha}_k \;=\; -\frac{|\sigma_k|}{\|r_k\|^2 + |\sigma_k|}, \tag{3.5}$$

which satisfy the conditions $r_{k+1} \in \mathcal{K}_k(A, r_1)$ and $r_{k+1} \notin \mathcal{K}_{k-1}(A, r_1)$. Also observe that the coefficient $\hat{\alpha}_k$ in (3.5) is computed by imposing the orthogonality condition $r_{k+1}^T p_k = r_{k+1}^T r_k = 0$. Moreover, from (3.5) and Theorem 2.1 in [Fasano, 2005a], it can be readily seen that the dummy residual $r_{k+1}$ satisfies also the required orthogonality properties

$$r_{k+1}^T r_i = 0, \qquad i \le k, \qquad\qquad \text{and} \qquad\qquad r_i^T r_{k+1} = 0, \qquad i > k+1.$$

Now, suppose that the FLR algorithm performs up to step $h$ and w.l.o.g. *the only one planar CG step is Step* $k_B < h$. Let us consider the matrices

$$R_h = \left(\frac{r_1}{\|r_1\|} \cdots \frac{r_h}{\|r_h\|}\right) \in \mathbb{R}^{n\times h}, \qquad\qquad P_h = \left(\frac{w_1}{\|r_1\|} \cdots \frac{w_h}{\|r_h\|}\right) \in \mathbb{R}^{n\times h},$$

generated by the FLR Algorithm. Then, from (3.5) and the instructions at *Step* $k_B$ we obtain the following result [Fasano and Roma, 2005].

**Theorem 3.1** *Consider the* FLR *algorithm where* $A$ *is symmetric, indefinite and nonsingular. Suppose* $\epsilon_k > 0$ *and let* $\|r_i\| \neq 0$, $i \le h$. *Assume the only one planar CG step performed by the* FLR *Algorithm is Step* $k_B < h$. *Then the following relations hold:*

$$P_h \tilde{L}_h^T \;=\; R_h, \tag{3.6}$$

$$A P_h \;=\; \left(R_h \;\vdots\; \frac{r_{h+1}}{\|r_{h+1}\|}\right) \begin{pmatrix} \bar{L}_h \\ \bar{l}_{h+1,h} e_h^T \end{pmatrix} D_h, \qquad h < n, \tag{3.7}$$

$$A R_h \;=\; \left(R_h \;\vdots\; \frac{r_{h+1}}{\|r_{h+1}\|}\right) \begin{pmatrix} T_h \\ t_{h+1,h} e_h^T \end{pmatrix}, \qquad h < n, \tag{3.8}$$

*where*

$$\tilde{L}_h = \begin{pmatrix} 1 & & & & & & & & & \\ -\sqrt{\beta_1} & \cdot & & & & & & & & \\ & \cdot & 1 & & & & & & & \\ & & -\sqrt{\beta_{k-1}} & 1 & & & 0 & & & \\ & & & \tilde{\alpha}_1 & \tilde{\alpha}_2 & & & & & \\ & & & \tilde{\alpha}_3 & \tilde{\alpha}_4 & 1 & & & & \\ & 0 & & & & -\sqrt{\beta_{k+2}} & \cdot & & & \\ & & & & & & \cdot & 1 & & \\ & & & & & & & -\sqrt{\beta_{h-1}} & 1 & \end{pmatrix} \qquad (3.9)$$

*with* $(\beta_k = \|r_{k+1}\|^2/\|r_k\|^2,\ \beta_{k+1} = \|r_{k+2}\|^2/\|r_{k+1}\|^2)$

$$\begin{array}{rclrcl} \tilde{\alpha}_1 & = & \dfrac{\hat{\alpha}_k}{\sqrt{\beta_k}}, & \tilde{\alpha}_2 & = & (1+\hat{\alpha}_k)\ sgn(\sigma_k), \\ \tilde{\alpha}_3 & = & \dfrac{\hat{\beta}_k \delta_k}{\Delta_k \sqrt{\beta_{k+1}\beta_k}}, & \tilde{\alpha}_4 & = & -\dfrac{\hat{\beta}_k\ \sigma_k}{\Delta_k \sqrt{\beta_{k+1}}}, \end{array} \qquad (3.10)$$

$$D_h = \begin{pmatrix} \frac{1}{a_1} & & & & & & & \\ & \cdot & & & & & & \\ & & \frac{1}{a_{k-1}} & & & 0 & & \\ & & & \frac{1}{\xi_k} & & & & \\ & & & & \frac{1}{\xi_{k+1}} & & & \\ & 0 & & & & \frac{1}{a_{k+2}} & & \\ & & & & & & \cdot & \\ & & & & & & & \frac{1}{a_h} \end{pmatrix}, \qquad (3.11)$$

$$\bar{L}_h \; = \; \left( \begin{array}{ccccccccc} 1 & & & & & & & & \\ -\sqrt{\beta_1} & \cdot & & & & & & & \\ & \cdot & 1 & & & 0 & & & \\ & & -\sqrt{\beta_{k-1}} & \bar{\alpha}_1 & \bar{\alpha}_3 & & & & \\ & & & \bar{\alpha}_2 & \bar{\alpha}_4 & & & & \\ & & & & \bar{\alpha}_5 & 1 & & & \\ & & 0 & & & -\sqrt{\beta_{k+2}} & \cdot & & \\ & & & & & & \cdot & 1 & \\ & & & & & & & -\sqrt{\beta_{h-1}} & 1 \end{array} \right) . \tag{3.12}$$

*The coefficients $\xi_k$ and $\xi_{k+1}$ are independent arbitrary non-zero parameters, and $\bar{\alpha}_i, \;\; i = 1, \ldots, 5$, have the following values:*

$$\begin{aligned} \bar{\alpha}_1 \; &= \; \frac{\sigma_k}{\|r_k\|^2} \xi_k, & \bar{\alpha}_2 \; &= \; \sqrt{\beta_k} \left[ \; sgn(\sigma_k) + \frac{\sigma_k}{\|r_k\|^2} \right] \xi_k, \\ \bar{\alpha}_3 \; &= \; \frac{\xi_{k+1}}{\sqrt{\beta_k} \hat{\sigma}_k} \left[ 1 - \frac{\sigma_k}{\|r_k\|^2} \hat{c}_k \right], & \bar{\alpha}_4 \; &= \; - \frac{\hat{c}_k \xi_{k+1}}{\hat{\sigma}_k (1 + \hat{\alpha}_k) \; sgn(\sigma_k)}, \\ \bar{\alpha}_5 \; &= \; - \frac{\xi_{k+1}}{\hat{\sigma}_k} \sqrt{\beta_{k+1}}. \end{aligned} \tag{3.13}$$

*Finally, $T_h$ is an irreducible symmetric tridiagonal matrix defined by*

$$\left( \begin{array}{c} T_h \\ 0 \cdots 0 \; t_{h+1,h} \end{array} \right) \; = \; \left( \begin{array}{c} \bar{L}_h \\ 0 \cdots 0 \; \bar{l}_{h+1,h} \end{array} \right) D_h \tilde{L}_h^T, \tag{3.14}$$

*where $\bar{l}_{h+1,h}$ and $t_{h+1,h}$ are the element $(h+1, h)$ of the matrix $\bar{L}_{h+1}$ and $T_{h+1}$, respectively.* □

Since both the matrices $\bar{L}_h$ and $\tilde{L}_h$ are nonsingular from (3.1) we also have the following result (see [Fasano and Roma, 2005]).

**Proposition 3.2** *Consider the* FLR *algorithm where $\epsilon_k > 0$ and let $\|r_i\| \neq 0, \; i \leq h$. Suppose the only one planar CG step performed by the* FLR *Algorithm is Step $k_B < h$. Then, the nonsingular matrix $Q_h \in \mathbb{R}^{h \times h}$ exists such that*

$$\bar{L}_h = \tilde{L}_h Q_h, \tag{3.15}$$

*where*

$$Q_h \; = \; \left(\begin{array}{ccc|cc|ccc} 1 & & & & & & & \\ & \ddots & & & & & 0 & \\ & & 1 & & & & & \\ \hline & & & \pi_{k,k} & \pi_{k,k+1} & & & \\ & & & \pi_{k+1,k} & \pi_{k+1,k+1} & & & \\ \hline & & & & & 1 & & \\ & 0 & & & & & \ddots & \\ & & & & & & & 1 \end{array}\right), \tag{3.16}$$

*with*

$$\begin{array}{llll} \pi_{k,k} \; = \; \bar{\alpha}_1, & & \pi_{k,k+1} \; = \; \bar{\alpha}_3, & \\ \pi_{k+1,k} \; = \; \dfrac{\bar{\alpha}_2 - \bar{\alpha}_1\tilde{\alpha}_1}{\tilde{\alpha}_2}, & & \pi_{k+1,k+1} \; = \; \dfrac{\bar{\alpha}_4 - \bar{\alpha}_3\tilde{\alpha}_1}{\tilde{\alpha}_2}. & \end{array} \tag{3.17}$$

□

Hence, by (3.14) and (3.15) we obtain

$$T_h \; = \; \tilde{L}_h(Q_h D_h)\tilde{L}_h^T \; = \; \tilde{L}_h B_h \tilde{L}_h^T, \tag{3.18}$$

and iterating (3.15) in case many planar CG steps are performed, it can be proved that the matrix $B_h$ is nonsingular, indefinite and $2 \times 2$ block diagonal.

# 4 Iterative approximation of a $\ell_2$-norm diagonal preconditioner

In the previous sections we have introduced the iterative scheme for solving the indefinite linear system (1.1). Now we show how the same iterative method can be used in order to build a diagonal preconditioner, based on the $\ell_2$-norm scaling of the columns of matrix $A$.

Let us denote by $a_1, \ldots, a_n$ the columns of the symmetric matrix $A$, that is

$$A = A^T = (a_1 \cdots a_n), \qquad a_i \in \mathbb{R}^n, \quad i = 1, \ldots, n.$$

Our aim is to compute the $\ell_2$-norm of the columns vectors $a_i$, i.e. the quantities $\|a_i\|_2$, $i = 1, \ldots, n$, in order to define the matrix

$$M = diag_{1 \leq i \leq n} \left\{\|a_i\|_2\right\} \tag{4.1}$$

to be used as diagonal preconditioner. Hence, the preconditioned matrix is $AM^{-1}$ and it is straightforward to see that to apply such a preconditioner is equivalent to scale the columns of the matrix $A$. Moreover, note that the preconditioned matrix $AM^{-1}$ is similar to the matrix $M^{-1/2}AM^{-1/2}$.

In order to compute the preconditioner $M$, first, observe that the diagonal elements of the matrix $A^2 = A^T A$ are given by $\|a_i\|_2^2$, i.e.

$$\left[A^2\right]_{ii} = diag\{\|a_i\|_2^2\} \quad i = 1, \ldots, n.$$

Of course the matrix $A^2$ is not available, but the decompositions of the matrix $A$ obtained in the previous section enable to compute an approximation of the diagonal elements $A^2$, as showed in the next proposition.

**Proposition 4.1** *Consider the* FLR *algorithm used to solve the linear system (1.1), where $\varepsilon_k$ satisfies the assumption (2.6) in [Fasano and Roma, 2005] for any $k$. Suppose to iterate the* FLR *algorithm up to step $h$ and that $\mathcal{K}_h(A,b) = Range(A)$. Then it results*

$$A^2 \;=\; R_h \left[\bar{L}_h D_h diag_{1\le i\le h}\left\{\|r_i\|_2\right\} B_h^{-1} diag_{1\le i\le h}\left\{\|r_i\|_2\right\} D_h \bar{L}_h^T\right]^2 R_h^T. \tag{4.2}$$

**Proof:** Let us denote by $A^+$ the Moore–Penrose pseudoinverse of the matrix $A$. In [Fasano, 2006]) it has been proved that

$$\left[A^+ - (w_1 \cdots w_h) B_h^{-1} (w_1 \cdots w_h)^T\right] y = 0, \qquad \text{for each} \qquad y \in \mathcal{K}_h(A,b),$$

where $(w_1 \cdots w_h)$ are the directions generated by the algorithm FLR as defined in Section 3. Since

$$(w_1 \cdots w_h) = P_h \; diag_{1\le i\le h}\left\{\|r_i\|_2\right\} \qquad \text{and} \qquad P_h = R_h \tilde{L}_h^{-T},$$

if $\mathcal{K}_h(A,b) = Range(A)$ we have

$$\begin{aligned} A^+ \;&=\; P_h \left[diag_{1\le i\le h}\left\{\|r_i\|_2\right\} B_h^{-1} diag_{1\le i\le h}\left\{\|r_i\|_2\right\}\right] P_h^T \\ &=\; R_h \tilde{L}_h^{-T} \left[diag_{1\le i\le h}\left\{\|r_i\|_2\right\} B_h^{-1} diag_{1\le i\le h}\left\{\|r_i\|_2\right\}\right] \tilde{L}_h^{-1} R_h^T. \end{aligned}$$

Now, since $A^+$ is also inner inverse of $A$, then $A = AA^+A$, which yields

$$A \;=\; AA^+A \;=\; AR_h \tilde{L}_h^{-T} \left[diag_{1\le i\le h}\left\{\|r_i\|_2\right\} B_h^{-1} diag_{1\le i\le h}\left\{\|r_i\|_2\right\}\right] \tilde{L}_h^{-1} R_h^T A.$$

By using the conditions $A = A^T$, $AR_h = R_h T_h$ and $T_h = T_h^T$ we obtain

$$A \;=\; R_h T_h \tilde{L}_h^{-T} \left[diag_{1\le i\le h}\left\{\|r_i\|_2\right\} B_h^{-1} diag_{1\le i\le h}\left\{\|r_i\|_2\right\}\right] \tilde{L}_h^{-1} T_h R_h^T,$$

so that, the relation $T_h = \bar{L}_h D_h \tilde{L}_h^T = \tilde{L}_h D_h \bar{L}_h^T$ yields

$$\begin{aligned} A^2 \;&=\; R_h \left[T_h \tilde{L}_h^{-T} diag_{1\le i\le h}\left\{\|r_i\|_2\right\} B_h^{-1} diag_{1\le i\le h}\left\{\|r_i\|_2\right\} \tilde{L}_h^{-1} T_h\right]^2 R_h^T \\ &=\; R_h \left[\bar{L}_h D_h diag_{1\le i\le h}\left\{\|r_i\|_2\right\} B_h^{-1} diag_{1\le i\le h}\left\{\|r_i\|_2\right\} D_h \bar{L}_h^T\right]^2 R_h^T. \end{aligned}$$

□

This proposition shows that to compute an approximation of $\|a_i\|_2^2 = e_i^T A^2 e_i$, we do not need to explicitely compute $A^2$. It suffices to apply the Krylov subspace method FLR and generate, as by product, the matrices $R_h$, $T_h$, $B_h$ and $\bar{L}_h$. Of course, it must be taken into account that this approximation has been obtained by using the pseudoinverse of the matrix $A$, restricted to the Krylov subspace spanned up to the $h$-th iteration of the method.

**Remark 4.1** Note that if no planar steps are performed by the FLR method, then the expression of $A^2$ reduces to

$$A^2 = R_h \left[L_h D_h L_h^T\right]^2 R_h^T = R_h T_h^2 R_h^T,$$

which gives for the $i$-th diagonal entry

$$\left[A^2\right]_{ii} = (e_i^T R_h) T_h^2 (R_h^T e_i) = \|T_h R_h^T e_i\|^2.$$

## 5 Conclusions and perspectives

In this paper we extend the use of a Krylov subspace method for iteratively generating a diagonal preconditioner based on the $\ell_2$-norm scaling, which should improve the efficiency of the iterative method in solving a large symmetric indefinite linear system. This approach can be easily embedded within a Newton–Krylov method, where a good efficiency is required to solve the Newton's system at each outer iteration of the method. A numerical investigation is certainly needed to assess the behaviour of a preconditioning technique, which makes use of the preconditioner proposed in this paper. It will be subject of a future work.

**Acknowledgements**

The author G.Fasano thanks the research programs "VISIR" and "SICUREZZA" for their support. G.Fasano also thanks Patrizia Novello and Viola Balestri for their unaware suggestions.

## References

[Bank and Chan, 1994] Bank, R. and Chan, T. (1994). A composite step bi-conjugate gradient algorithm for nonsymmetric linear systems. *Numerical Algorithms*, 7, pag. 1–16.

[Barlow and Toraldo, 1995] Barlow, J. and Toraldo, G. (1995). The effect of diagonal scaling on projected gradient methods for bound constrained quadratic programming problems. *Optimization Methods and Software*, 5, pag. 235–245.

[Bunch, 1971] Bunch, J. (1971). Equilibration of symmetric matrices in the max-norm. *Journal of ACM*, 18, pag. 566–572.

[Duff et al., 1986] Duff, I. S., Erisman, A. M., and Reid, J. K. (1986). *Direct methods for sparse matrices.* Oxford University Press, London.

[Fasano, 2001] Fasano, G. (2001). *Use of conjugate directions inside Newton–type algorithms for large scale unconstrained optimization.* PhD thesis, Università di Roma "La Sapienza", Rome, Italy.

[Fasano, 2005a] Fasano, G. (2005a). Planar–conjugate gradient algorithm for large–scale unconstrained optimization, Part 1: Theory. *Journal of Optimization Theory and Applications*, 125, pag. 523–541.

[Fasano, 2005b] Fasano, G. (2005b). Planar–conjugate gradient algorithm for large–scale unconstrained optimization, Part 2: Application. *Journal of Optimization Theory and Applications*, 125, pag. 543–558.

[Fasano, 2006] Fasano, G. (2006). Lanczos-conjugate gradient method and pseudoinverse computation, in unconstrained optimization. *Journal of Optimization Theory and Applications*, 130(3). (to appear).

[Fasano and Roma, 2005] Fasano, G. and Roma, M. (2005). Iterative computation of negative curvature directions in large scale optimzation: theory and preliminary numerical results. Technical Report n. 12, Dipartimento di Informatica e Sistemistica. To appear on Computational Optimization and Applications.

[Greenbaum, 1997] GREENBAUM, A. (1997). *Iterative methods for solving linear systems.* SIAM, Philadelphia, PA.

[Hestenes, 1980] HESTENES, M. (1980). *Conjugate Direction Methods in Optimization.* Springer Verlag, New York.

[Higham, 2002] HIGHAM, N. (2002). *Accuracy and stability of numerical algorithms.* SIAM, Philadelphia, PA. Second edition.

[Liu and Nocedal, 1989] LIU, D. and NOCEDAL, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45, pag. 503–528.

[Parlett and Landis, 1982] PARLETT, B. and LANDIS, T. (1982). Methods for scaling to double stochastic form. *Linear Algebra and Applications*, 48, pag. 53–79.

[Roma, 2005] ROMA, M. (2005). A dynamic scaling based preconditioning for truncated newton methods in large scale unconstrained optimization. *Optimization Methods and Software*, 20, pag. 693–713.

[Ruiz, 2001] RUIZ, D. (2001). A scaling algorithm to equilibrate both rows and columns norms in matrices. Techical Report RAL–TR–2001–034, Computational Sciences and Engineering Department, Rutherford Appleton Laboratory, Oxon, UK. Submitted to *Linear Algebra and Applications.*

[Schneider and Zenios, 1990] SCHNEIDER, M. H. and ZENIOS, S. (1990). A comparative study of algorithms for matrix balancing. *Operations Research*, 38, pag. 439–455.