Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Lecturas en Ciencias de la Computación
ISSN 1316-6239

Pseudoinverse Preconditioners and Iterative Methods for Large Dense Linear Least-Squares Problems

Oskar Cahueñas, Luis Hernández-Ramos, and Marcos Raydan

RT 2012-04

Centro de Cálculo Científico y Tecnológico de la UCV
CCCT-UCV
Caracas, Julio 2012.

Pseudoinverse preconditioners and iterative methods for large dense linear least-squares problems

Oskar Cahueñas * Luis M. Hernández-Ramos † Marcos Raydan ‡

June 28, 2012

Abstract

We address the issue of approximating the pseudoinverse of the coefficient matrix A for dynamically building preconditioning strategies for the numerical solution of large dense linear least-squares problems. The new preconditioning strategies are embedded into simple and well-known iterative schemes that avoid the use of the, usually ill-conditioned, normal equations. We analyze a scheme to approximate the pseudoinverse, based on Schulz iterative method, and also different iterative schemes, based on extensions of Richardson's method, and the conjugate gradient method, that are suitable for preconditioning strategies. We present preliminary numerical results to illustrate the advantages of the proposed schemes.

Key words: Schulz method, pseudoinverse, linear least-squares problems, preconditioned Richardson's method, conjugate gradient method.

1 Introduction

We are interested in solving dense large-scale linear least-squares problems of the form:

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2, \tag{1}$$

where A is an $m \times n$ matrix, with $m > n$. The solution of (1) can be obtained by solving the normal equations

$$A^T A x = A^T b, \tag{2}$$

which can be solved by using direct methods or iterative methods combined with preconditioning techniques. For a complete revision on the difficulties of solving the normal equations, and for some possible remedies see, e.g., [3, 15, 16].

Direct methods for solving (2) are based on matrix factorizations (e.g. Cholesky, QR or the SVD factorization). If the pseudoinverse, $A^\dagger \in \mathbb{R}^{n \times m}$, of the matrix A is available, then the minimum norm solution of (1) is given by $x = A^\dagger b$.

From a historical perspective, iterative methods have been mainly associated with sparse large-scale problems [15, 17]. Nevertheless, in the last two decades, mainly due to the popularity of parallel architectures, there have been a growing interest in using iterative methods also for dense large-scale problems; see, e. g., [6, 19]. Dense large-scale least-squares problems appear in astronomy, geodesy, and statistics, among others; see, e.g., [1, 2, 7, 8, 11].

*Centro de Cálculo Científico y Tecnológico, Postgrado en Ciencias de la Computación, Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela (oskarcah@gmail.com).

†Departamento de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela (luis.hernandez@ciens.ucv.ve). Partially Supported by the center CCCT at UCV.

‡Departamento de Cómputo Científico y Estadística, Universidad Simón Bolívar, Ap. 89000, Caracas 1080-A, Venezuela (mraydan@usb.ve). Partially Supported by Fonacit-CNRS project No-201200140 and CEsMa at USB.

The main observation in this work is the following: if we could obtain by some means an approximation $C \in I\!R^{n\times m}$ of the matrix $A^\dagger$, then we could use it as a preconditioner for the well-known conjugate gradient method, and also for the following iterative residual-type scheme

$$x_{k+1} = x_k + \lambda_k C r_k, \tag{3}$$

where $r_k = b - Ax_k$ is the residual vector at x_k, and $\lambda_k > 0$ is a suitable step length.

The scheme (3) can be viewed as a preconditioned Richardson's method. Clearly, if $C = A^\dagger$, and $\lambda_k = 1$, then the iterative scheme (3) finds the solution of (1) in one iteration. In practice, the better C approximates $A^\dagger$ the fewer iterations are required in either the conjugate gradient method or in (3), to solve (1); see e.g., [4, 5].

In this work we will analyze some ideas to find an approximation of the rectangular matrix $A^\dagger$ which are based on the classical Schulz iterative method [18]. We will also describe how to choose the step length λ_k to guarantee the convergence of (3). A similar combination of (3) with Schulz iterative method has been analyzed for solving nonsingular linear systems of equations in [4].

The rest of this work is organized as follows. In section 2, we recall Schulz method to approximate the pseudoinverse of A and analyze some of its key features for solving (1). In section 3, we present and analyze the so-called Richardson-PR2 extension of the classical Richardson's method, developed by Brezinski [4, 5] for solving linear systems of equations. In section 4, we combine the Richardson-PR2 method, as well as the conjugate gradient method, with a dynamical preconditioning strategy based on Schulz method for solving least-squares problems. In section 5, we present preliminary numerical results on some dense ill-conditioned least-squares problems. Finally, in section 6, we present some final remarks.

2 Schulz method to approximate the pseudoinverse

Schulz method is a well-known iterative method to approximate the (pseudo) inverse of a matrix A. From a given M_0, it produces the following iterates

$$M_{k+1} = 2M_k - M_k A M_k. \tag{4}$$

Schulz method can be obtained applying Newton's method to the related map $F(X) = X^{-1} - A$. It was originally developed by Schulz [18], and possesses numerical stability, local q-quadratic convergence, and global convergence from some specific choices of M_0 (e.g., $M_0 = A^T/\|A\|_2^2$). Moreover, if A does not have an inverse the scheme given by (4) converges to $A^\dagger$, the pseudoinverse of A; see [13] and its references for a complete revision of this topic.

The necessary and sufficient condition for the convergence of Schulz method, to approximate the inverse of a given nonsingular matrix ($m = n$) or to approximate the pseudoinverse of a rectangular full-rank matrix $A \in \mathbb{R}^{m\times n}$, is well-known; see e.g., [9, 10].

Theorem 2.1. *Let $A \in \mathbb{R}^{m\times n}$ be a full-rank matrix, and let $M_0 \in \mathbb{R}^{n\times m}$. The sequence (4) generated by Schulz method converges to $A^\dagger$ if and only if $\rho(I - AM_0) < 1$, where $\rho(C)$ is the spectral radius of the matrix C.*

From a computational point of view, an interesting relation obtained from (4), that will be used later in this work, is the following. For any k,

$$I - M_{k+1}A = I - (2M_k - M_k A M_k)A = (M_k A)^2 - 2M_k A + I = (I - M_k A)^2.$$

Therefore, using an inductive argument

$$I - M_k A = (I - M_{k-1}A)^2 = ((I - M_{k-2}A)^2)^2 = \cdots = (I - M_0 A)^{2^k}.$$

Let us now define $B_k = M_k A$, which combined with the previous equality, yields

$$B_k = I - (I - B_0)^{2^k}. \tag{5}$$

Notice also that the matrix products $M_k A$ or AM_k would produce dense matrices even if A is a sparse matrix, and so, Schulz method is not suitable in the sparse case; whereas in the dense case it does not introduce any additional fill-in.

A well-known and suitable choice for $M_0 \in \mathbb{R}^{n\times m}$, that satisfies the convergence condition in Theorem 2.1 (see, e.g., [9]), is

$$M_0 = \frac{A^T}{\|A\|_2^2}. \tag{6}$$

To close this section, we will establish two theoretical results that are relevant to the application of Schulz method to rectangular matrices. First, we will discuss an orthogonality condition of the product $M_k(b - Ax)$ which is analogous to the standard orthogonality condition that holds at the solution of linear least-squares problems. Second, we are going to prove that if λ_{k_i} is the i-th eigenvalue of $M_k A$, then λ_{k_i} is also the k-th Newton iteration for finding the root of the function $f(x) = 1 - 1/x$ (which is clearly $x = 1$) from the initial guess λ_{0_i}, the i-th eigenvalue of $M_0 A$.

Before we proceed we need to show, as a preliminary result, that the eigenvalues of $M_k A$, for all k, are all uniformly bounded.

Theorem 2.2. *Let $A \in \mathbb{R}^{m\times n}$ be a full-rank matrix with $m \geq n$, and let M_k be the k-th Schulz iterate (4), where M_0 is given by (6). Then, for any eigenvalue λ_i of the matrix $M_k A$, $0 < \lambda_i \leq 1$.*

Proof. For $k = 0$, $M_0 = A^T/\|A\|_2^2$, and so $M_0 A = A^T A/\|A\|_2^2$. Since A is full-rank, then $M_0 A$ is symmetric positive definite and hence all its eigenvalues are real and positive. On the other hand, for any M, the spectral radius $\rho(M)$ is less than or equal to the norm of M for any matrix norm, and so

$$0 < \rho(M_0 A) = \rho\left(\frac{A^T A}{\|A\|_2^2}\right) \leq \left\|\frac{A^T A}{\|A\|_2^2}\right\|_2 \leq 1.$$

Therefore, the result holds for $k = 0$. Now, let $k \geq 1$. Using the recurrence (5), we have that

$$M_k A = I - (I - M_0 A)^{2^k}.$$

If we now consider the polynomial $p(x) = 1 - (1 - x)^{2^k}$, $M_k A$ can be written as $M_k A = p(M_0 A)$, and its eigenvalues are of the form $\Lambda(\lambda_i) = 1 - (1 - \lambda_i)^{2^k}$ where λ_i is an eigenvalue of $M_0 A$. Since $0 < \lambda_i \leq 1$, it follows that $0 < \Lambda(\lambda_i) \leq 1$, and the result is established for $k \geq 1$. □

We can also establish a bound for the eigenvalues of the matrix $2I - M_k A$.

Corollary 2.1. *For any eigenvalue λ_i of $2I - M_k A$, $1 < \lambda_i \leq 2$.*

Proof. Consider the polynomial $p(x) = 1 + x$, and write $2I - M_k A = p(I - M_k A)$. Since the eigenvalues of $I - M_k A$ are in the interval $(0, 1]$ the result follows directly. □

Let us now establish the key orthogonality result.

Theorem 2.3. *Let $A \in \mathbb{R}^{m\times n}$ be a full-rank matrix with $m \geq n$, $b \in \mathbb{R}^n$, and let M_k be the k-th Schulz iterate (4), where M_0 is given by (6). Then x_M is the least-squares solution of (1) if and only if the following orthogonality condition holds:*

$$M_k(b - Ax_M) = 0, \quad k = 0, 1, \ldots$$

Proof. The proof is by induction on k. For $k = 0$, $M_0 = A^T/\|A\|_2^2$, and using the standard orthogonality condition, $A^T(b - Ax) = 0$, obtained from (2) for linear least-squares solutions, we have that

$$M_0(b - Ax_M) = \frac{1}{\|A\|_2^2}\left(A^T(b - Ax)\right) = 0,$$

if and only if x_M solves (1). Now, for $k = i \geq 1$, let us assume the inductive hypothesis $M_i(b - Ax_M) = 0$. It suffices to show that the result holds for $k = i + 1$. In that case

$$M_{i+1} = 2M_i - M_iAM_i = (2I - M_iA)\, M_i,$$

and so

$$M_{i+1}(b - Ax_M) = (2I - M_iA)\, M_i(b - Ax_M).$$

Using now the inductive hypothesis $M_i(b - Ax_M) = 0$, it follows that $(2I - M_iA)\, M_i(b - Ax_M) = 0$. For the *only if* part, we need to guarantee that zero cannot be obtained in the product of $(2I - M_iA)$ times a nonzero vector; i.e., we need to verify that if $b' \in \mathbb{R}^n$ then $(2I - M_iA)\, b' = 0$ if and only if $b' = 0$; which holds when the matrix $(2I - M_iA)$ is nonsingular. By Corollary (2.1), $(2I - M_iA)$ is clearly nonsingular since all its eigenvalues are in the interval $(1, 2]$. Consequently, setting $b' = M_i(b - Ax_M)$ the result is established for $k = i + 1$. □

Theorem 2.3 plays a central role in our work, since it indicates that the choice of M_k, as a preconditioner, is very convenient in the sense that the resulting preconditioned system has n equations and n unknowns and its solution coincides with the solution of (1).

Our next result establishes the connection between the eigenvalues of M_kA and the Newton iterations for finding the root of the function $f(x) = 1 - 1/x$.

Theorem 2.4. *Let $A \in \mathbb{R}^{m\times n}$ be a full-rank matrix with $m \geq n$, M_k be the k-th Schulz iterate (4) where M_0 is given by (6), and let λ_{k_i} be the i-th eigenvalue of M_kA. Then λ_{k_i} is also the k-th Newton's iteration in one variable to find the root of $f(x) = 1 - 1/x$, from λ_{0_i}, the i-th eigenvalue of M_0A, where $M_0 = A^T/\|A\|_2^2$.*

Proof. For $f(x) = 1 - 1/x$, the one-dimensional Newton iteration is given by

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k + x_k\,(1 - x_k) = 2x_k - x_k^2 = 1 - (1 - x_k)^2\,.$$

Therefore $1 - x_{k+1} = (1 - x_k)^2$, and using an inductive argument, we obtain the recurrence

$$1 - x_k = (1 - x_{k-1})^2 = \cdots = 1 - (1 - x_0)^{2^k}\,.$$

Hence, Newton's iteration for $f(x) = 1 - 1/x$ can also be written as

$$x_k = 1 - (1 - x_0)^{2^k}\,.$$

On the other hand, for $k \geq 1$ we already established that $M_kA = I - (I - M_0A)^{2^k}$. If we now consider the polynomial $p_k(x) = 1 - (1 - x)^{2^k}$, we have that $p_k(M_0A) = M_kA$, and so $p_k(\lambda_{0_i}) = 1 - (1 - \lambda_{0_i})^{2^k} = \lambda_{k_i}$, and the proof is complete. □

Theorem 2.4 shows that using the matrices M_k as preconditioners tends to cluster the eigenvalues of the resulting preconditioned system around $\lambda = 1$ when k increases; which is a convenient effect to accelerate the convergence of iterative methods.

3 Extensions of Richardson's method for linear systems

A classical iterative method for solving linear systems of the form $Ax = b$, which is well-suited for preconditioning strategies, is Richardson's method, that can be written as

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \lambda_k r^{(k)} \\ r^{(k+1)} &= b - Ax^{(k)}, \end{aligned} \tag{7}$$

where λ_k can be chosen in different ways [5, 14, 20]. The standard choice that minimizes $\|r^{(k+1)}\|_2^2$, is given by

$$\lambda_k = -\frac{\left(Ar^{(k)}\right)^T r^{(k)}}{\|Ar^{(k)}\|_2^2}.$$

An extension of Richardson's method, which includes the use of arbitrary search directions, and hence allows the use of preconditioning strategies, was developed by Brezinski [4, 5] as follows. Consider the sequence $\left\{x^{(k)}\right\}$ generated by (7), and define a new sequence $\left\{y^{(k)}\right\}$ given by

$$\begin{aligned} y^{(k)} &= x^{(k)} - \lambda_k z^{(k)} \\ \rho^{(k)} &= r^{(k)} + \lambda_k A z^{(k)} \end{aligned} \tag{8}$$

where $\rho^{(k)} = b - Ay^{(k)}$ is the new residual vector associated with $y^{(k)}$, and $z^{(k)}$ is a new search direction that, in principle, can be chosen arbitrarily. From now on, as suggested in [4], we denote the method given by (8) as Richardson-PR2. Later in this work, we will combine Richardson-PR2 with Schulz iterations to build preconditioned directions $z^{(k)}$ and solve (1). In [12], Richardson-PR2 has been successfully combined with some standard preconditioning strategies for solving nonsymmetric positive definite linear systems of equations.

In (8), the step length λ_k is chosen to minimize $\|\rho^{(k)}\|_2^2$, i.e., it is given by

$$\lambda_{k_{min}} = -\frac{\left(Az^{(k)}\right)^T r^{(k)}}{\|Az^{(k)}\|_2^2}. \tag{9}$$

Notice that using (9)

$$\begin{aligned} \|\rho^{(k)}\|_2^2 = \rho^{(k)T}\rho^{(k)} &= \left(r^{(k)T} + \lambda_k (Az^{(k)})^T\right)\left(r^{(k)} + \lambda_k (Az^{(k)})\right) \\ &= r^{(k)T} r^{(k)} + 2\lambda_k (Az^{(k)})^T r^{(k)} + \lambda_k^2 (Az^{(k)})^T (Az^{(k)}) \\ &= \|r^{(k)}\|_2^2 - \frac{\left((Az^{(k)})^T r^{(k)}\right)^2}{\left((Az^{(k)})^T (Az^{(k)})\right)^2} \\ &= \|r^{(k)}\|_2^2 - \frac{\left((Az^{(k)})^T r^{(k)}\right)^2}{\|Az^{(k)}\|_2^2 \|r^{(k)}\|_2^2}\|r^{(k)}\|_2^2 \\ &= \left(1 - \cos^2(\theta_k)\right)\|r^{(k)}\|_2^2, \end{aligned}$$

and hence

$$\min_{\lambda_k}\left(\|\rho^{(k)}\|_2\right) = \|r^{(k)}\|_2 \sin(\theta_k), \tag{10}$$

where θ_k is the angle between $r^{(k)}$ and $Az^{(k)}$. Therefore, with this optimal choice for λ_k we have that

$$\|\rho^{(k)}\|_2 \leq \|r^{(k)}\|_2$$

and so, since $r^{(k)} \to 0$ when $k \to \infty$, we obtain

$$\lim_{k\to\infty} \frac{\|\rho^{(k)}\|_2}{\|r^{(k)}\|_2} = 0 \iff \lim_{k\to\infty} \theta_k = 0 \text{ or } \lim_{k\to\infty} \theta_k = \pi. \tag{11}$$

In other words, when using $\lambda_{k_{min}}$ a sufficient condition to obtain a significant acceleration with respect to the classical Richardson's methods is that either $\theta_k \to 0$ or $\theta_k \to \pi$. If we define

$$Q_k = \frac{(Az^{(k)})(Az^{(k)})^T}{\|Az^{(k)}\|_2^2}, \tag{12}$$

it follows that

$$\rho^{(k)} = (I - Q_k)r^{(k)}. \tag{13}$$

Notice that $Q_k^2 = Q_k$ and $Q_k^T = Q_k$, and so $I - Q_k$ is an orthogonal projection. As a consequence, we obtain a geometrical interpretation of the optimal choice of step length (9): the residual $\rho^{(k)}$ is the orthogonal projection of $r^{(k)}$ onto $Az^{(k)}$.

In case of solving a square nonsingular lineal system, the condition (11) for the convergence of $\rho^{(k)}$ implies that $\rho^{(k)}$ tends to zero if and only if $r^{(k)}$ and $Az^{(k)}$ are collinear, i.e., $z^{(k)} = \pm\beta A^{-1}r^{(k)}$ for some $\beta \in \mathbb{R}$ and $\beta \neq 0$. However, such a choice for $z^{(k)}$ is not practical, since it requires A^{-1}. A practical option, instead, is to choose a sequence of matrices $\{C_k\}$ converging to A^{-1}. In that case, the search direction is given by

$$z^{(k)} = C_k r^{(k)}. \tag{14}$$

Using (14), the scheme given by (8) can now be written as

$$\begin{array}{l} y^{(k)} = x^{(k)} - \lambda_k C_k r^{(k)} \\ \rho^{(k)} = r^{(k)} + \lambda_k A C_k r^{(k)} \\ \lambda_k = -\frac{\left(AC_k r^{(k)}\right)^T r^{(k)}}{\|AC_k r^{(k)}\|_2^2}. \end{array} \tag{15}$$

Finally, by reassigning (or cycling) the sequence of iterates (see [5]), $x^{(k+1)} = y^{(k)}$ and $r^{(k+1)} = \rho^{(k)}$, at every k, we obtain the following scheme starting from a given $x^{(0)}$ and $r^{(0)} = b - Ax^{(0)}$

$$\begin{array}{l} x^{(k+1)} = x^{(k)} + \lambda_k C_k r^{(k)} \\ r^{(k+1)} = r^{(k)} - \lambda_k A C_k r^{(k)} \\ \lambda_k = \frac{\left(AC_k r^{(k)}\right)^T r^{(k)}}{\|AC_k r^{(k)}\|_2^2}. \end{array} \tag{16}$$

Note that, as expected, if we fix $C_k = I$ for all k in (16), the classical Richardson's method (7) is obtained.

Algorithm 3.1 presents the general version of Richardson-PR2 to solve $Ax = b$. In Algorithm 3.1 we assume the existence of a procedure to build a preconditioner matrix C_k at every iteration, starting from a given C_0. For Algorithm 3.1, and for all the forthcoming algorithms in this work, N_{max} is a given positive integer that represents the maximum number of iterations; and $\epsilon > 0$ is a given real number for stopping the iterations.

4 Preconditioning least-squares problems using Schulz method

In this section we will combine Algorithm 3.1 and Schulz method, given by (4), to solve (1) using an extended preconditioned Richardson's method; i.e., we will use, in Algorithm 3.1, $C_k = M_k$ for all k. A similar combination for solving nonsingular linear systems of equations can be found in [4]. We will also describe a straightforward combination of Schulz method with the well-known preconditioned conjugate gradient method to solve (1).

First, we discuss the condition number of the least-squares problem (1) when the matrices M_k, obtained from Schulz method, are used as preconditioners. For that the condition number of the rectangular matrix A plays an important role [3]. The standard definition that generalizes the condition number of nonsingular matrices is $\kappa(A) = \|A\|_2\|A^\dagger\|_2 = \sigma_1/\sigma_r$, where $0 < r = rank(A) \leq \min(m, n)$,

Algorithm 3.1 Richardson-PR2 (General version)

```
function RichardsonPR2(A, b, x^(0), C_0, N_max, ε)
    r^(0) ← b − Ax^(0)
    for k = 0, 1 ..., N_max do
        v ← C_k r^(k)
        u ← Av
        p_1 ← u^T r^(k)
        p_2 ← u^T u
        λ_k ← p_1/p_2
        x^(k+1) ← x^(k) + λ_k v
        r^(k+1) ← r^(k) − λ_k u
        if ‖r^(k+1)‖_2 < ε then
            return x^(k+1)
        end if
    end for
end function
```

and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ are the nonzero singular values of A. It is well-known that in general the condition number of least-squares problems depends on A and also on the right-hand side b. Moreover, From Theorem 1.4.6 and Remark 1.4.1 in [3], it follows that if $rank(A) = n$, then

$$\kappa_{LS}(A, b) = \kappa(A)\left(1 + \kappa(A)\frac{\|r\|_2}{\|A\|_2\|x_{LS}\|_2}\right) \tag{17}$$

where $r = b - Ax_{LS}$ and x_{LS} solves (1).

Now, if M_k for any k, obtained from Schulz method, is used as a preconditioner, then the least-squares condition number in (17) becomes

$$\kappa_{LS}(M_kA, M_kb) = \kappa(M_kA) + \kappa^2(M_kA)\frac{\|M_k(b - Ax_{LS})\|_2}{\|M_kA\|_2\|x_{LS}\|_2}. \tag{18}$$

From Theorem (2.3) it follows that $M_k(b - Ax_{LS}) = 0$ for all k. Therefore, we deduce that using Schulz iterates as preconditioners eliminates the term of order $\kappa^2(A)$ in (18), yielding

$$\kappa_{LS}(M_kA, M_kb) = \kappa(M_kA). \tag{19}$$

Since M_k converges to $A^\dagger$, using M_k for k sufficiently large, guarantees that $\kappa_{LS}(M_kA, M_kb) \approx 1$. In other words, using the matrices M_k as preconditioners produce the desired effect of accelerating the convergence of residual iterative methods.

We are now ready to present, in Algorithm 4.1, the combination of Algorithm 3.1 and Schulz method for solving the linear least-squares problem.

To finish this section, we now describe how to combine Schulz method with the well-known preconditioned conjugate gradient method (see, e.g., [15]), for symmetric and positive definite systems, to solve (1). Notice that from the proof of Theorem 2.2 it is clear that M_kA is symmetric and positive definite for any $k \geq 0$; and also that from Theorem 2.3 solving $M_kAx = M_kb$ for any $k \geq 0$ is equivalent to solving the original least-squares problem (1). Hence, the conjugate gradient method can be applied to the system $M_kAx = M_kb$ for a preestablished value of k. This combination will be denoted from now on as the CG-Schulz method.

5 Numerical Experiments

In this section, we present some numerical experiments to illustrate several aspects of the proposed schemes CG-Schulz and and also PR2-Schulz, described in Algorithm 4.1, as well as their performance

Algorithm 4.1 PR2-Schulz

function PR2-SCHULZ($A, b, x^{(0)}, N_{max}, \epsilon$)
  $r^{(0)} \leftarrow b - Ax^{(0)}$
  $M \leftarrow A^T / \|A\|_2^2$
  $d \leftarrow Mr^{(0)}$
  for $k = 0, 1 \ldots, N_{max}$ **do**
    $u \leftarrow Ad$
    $p_1 \leftarrow u^T r^{(k)}$
    $p_2 \leftarrow u^T u$
    $\lambda_k \leftarrow p_1 / p_2$
    $x^{(k+1)} \leftarrow x^{(k)} + \lambda_k d$
    $r^{(k+1)} \leftarrow r^{(k)} - \lambda_k u$
    $M \leftarrow 2M - MAM$
    $d \leftarrow Mr^{(k+1)}$
    if $\|d\|_2 < \epsilon$ **then**
      return $x^{(k+1)}$
    end if
  end for
end function

on dense linear least-squares problems. The experiments were carried out in MATLAB© R2008a on an INTEL© Core 2 DUO@1.8 GHz machine with 2Gb of RAM, running the Windows XP operating system.

The dense matrices to be considered in our experiments are generated by MATLAB©functions:

- Matrix $NormRand(\mu, \sigma, m, n)$, which is a dense $m \times n$ real matrix with normally distributed random values (average μ and standard deviation σ). It is generated using the MATLAB©command $randn(m, n).*\sigma + \mu$. The condition number of these matrices depends on σ and μ. When $\mu = 0$, the smaller the value of σ the higher the condition number of the matrix.

- Matrix $RandSingVal((\sigma_1, \ldots, \sigma_n), m)$, which is a dense $m \times n$ real matrix with randomly distributed singular values $\sigma_1, \sigma_2, \ldots \sigma_n$, obtained as follows:

  1. Let $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ be dense matrices generated with the MATLAB©`rand` function.
  2. Let Q_1, R_1 and Q_2, R_2 be the QR factorizations of U and V respectively.
  3. The final matrix is obtained as $Q_1 * diag(\sigma_1, \sigma_2, \ldots, \sigma_n, 0, \ldots, 0) * Q_2^T$.

- Matrix $RandSVD(m)$, from the MATLAB©gallery. It is a dense $m \times m$ real matrix whose singular values are geometrically distributed between 0 and 1. These matrices are extremely ill-conditioned, with condition number of the order of $1/\sqrt{\epsilon}$, where ϵ is the machine *epsilon*.

Our numerical tests are divided in two groups. First, we present some experiments to appreciate the quality of the matrices M_k as preconditioners. We will illustrate, e.g., that when $k \to \infty$, the eigenvalues of $M_k A$ tend to 1, as indicated by Theorem 2.4. Second, we will present experiments to illustrate the advantages of our combined schemes to solve least-squares dense linear problems.

5.1 M_k as a preconditioner

In this section, we present some experiments to illustrate the quality of the matrices M_k, when used as preconditioners, to accelerate iterative methods for solving problem (1).

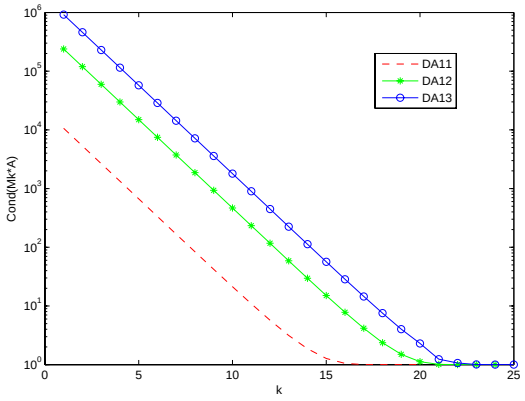

Figure 1: $\kappa(M_k A)$ when k increases for matrices in subgroup A

In our first experiment, we monitor the condition number $\kappa(M_k A)$ when k increases, where M_k is the Schulz iterate for the matrix A. For that, we consider the following test matrices:

1. **Subgroup A**
   - $DA11 = NormRand(100, 5, 200, 20)$, for which $\kappa(DA11) = 1.45 \times 10^2$.
   - $DA12 = NormRand(300, 10, 300, 100)$, for which $\kappa(DA12) = 6.9 \times 10^2$.
   - $DA13 = NormRand(500, 20, 1000, 400)$, for which $\kappa(DA13) = 1.35 \times 10^3$.
2. **Subgroup B**
   - $DB11 = RandSVD(500)$, for which $\kappa(DB11) = 6.71 \times 10^7$.
   - $DB12 = RandSingVal((10, 500, 5000, 5 \times 10^4, 5 \times 10^5, 5 \times 10^6), 100)$, for which $\kappa(DB13) = 5 \times 10^5$.
   - $DB13 = RandSingVal((100, 500, 5000, 5 \times 10^4, 5 \times 10^6, 5 \times 10^8), 100)$, for which $\kappa(DB13) = 5 \times 10^6$.

Figures 1 and 2 show the value of $M_k A$ as k increases for each subgroup. We can observe the fast convergence of $\kappa(M_k A)$ to 1, as indicated by the theoretical results in section 2.

For our second experiment, we monitor the convergence of the eigenvalues of $M_k A$ towards 1. For that we consider the test matrix $DA21 = RandSingVal((1, 2, 3, 4, 5), 100)$. We have chosen this test matrix with very few columns to easily visualize the spectrum of $M_k A$ as k increases. Figure 3 shows the behavior of the five eigenvalues of $M_k A$ as k increases. For this particular experiment we computed the eigenvalues of $M_k A$ using the MATLAB©function `eig`, and we also computed the one-variable Newton iterations on the function $f(x) = 1 - 1/x$, using as initial values the eigenvalues of $M_0 A$. We can observe that the eigenvalues converge to 1, as indicated by Theorem 2.4. Notice also that the convergence accelerates in the last iterations, which is a standard observation for Newton's method. Similar results were obtained when this experiment was applied to matrices in subgroups A and B.

5.2 PR2-Schulz and CG-Schulz for least-squares problems

In this section, we present some experiments to illustrate the performance of the proposed schemes (PR2-Schulz and CG-Schulz) on dense linear least-squares problems. For comparisons we will use the Richardson-PR2 method (Algorithm 3.1) fixing $C_k = I$ for all k, when applied directly to the normal

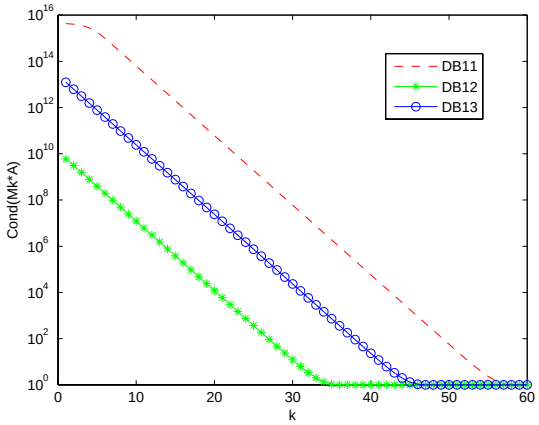

Figure 2: $\kappa(M_k A)$ when k increases for matrices in subgroup B

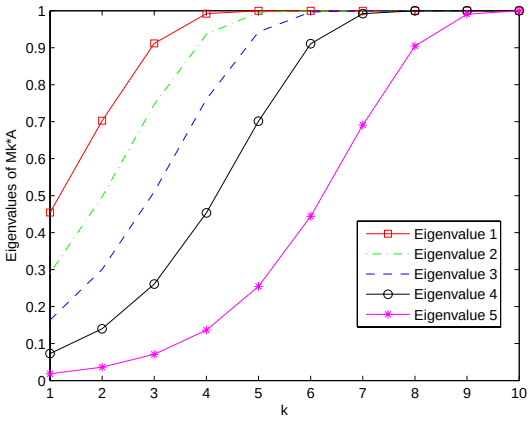

Figure 3: Convergence of the eigenvalues of $M_k A$ when k increases for $A = DA21$

equations (2), i.e., using the square and nonsingular $A^T A$ as the coefficient matrix, and $A^T b$ as the right hand side vector. This combination will be denoted as Richardson-NEQ.

We consider the following test matrices:

- $DD11 = RandSingVal((1 : 0.1 : 20), 500)$, for which $\kappa(DD11) = 20$
- $DD12 = RandSingVal((1, 2, \ldots, 98, 500, 1. \times 10^5), 500)$, for which $\kappa(DD12) = 1. \times 10^5$
- $DD13 = RandSingVal((0.01, 1, 2, \ldots, 297, 500, 1. \times 10^6), 500)$, for which $\kappa(DD13) = 1. \times 10^8$
- $DD14 = RandSingVal((0.01, 1, 2, \ldots, 297, 500, 1. \times 10^5), 5000)$, for which $\kappa(DD14) = 1. \times 10^7$

In all our experiments we set $b = (1, 1, 1, \ldots, 1)^T$ and $x_0 = (1, 1, 1, \ldots 1)^T$. The tolerance for stopping the iterations is $\epsilon = 10^{-8}$, and the maximum number of allowed iterations is $M_{max} = 200$. Based on Theorem 2.3, for the PR2-Schulz and the CG-Schulz methods we could stop the iterations when $\|M_k r_k\|_2 \leq \epsilon$. Nevertheless, in PR2-Schulz M_k changes dynamically at every k, whereas in CG-Schulz M_k is a fixed given matrix, computed in advance using Schulz method, and then applied as a preconditioner to the CG method for all iterations. Hence, for the sake of comparisons, in all our experiments we will consider the norm of the least-squares residual, $\|A^T(Ax_k - b)\|_2$, for the stopping

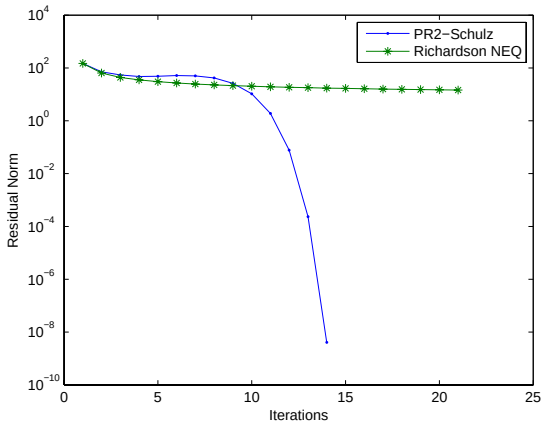

Figure 4: Norm of the residual of the first 20 iterations of PR2-Schulz and Richardson-NEQ for $A = DD11$

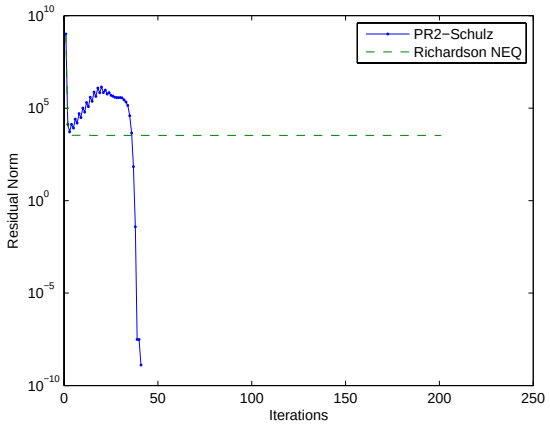

Figure 5: Norm of the residual of PR2-Schulz and Richardson-NEQ for $A = DD12$

criteria. In the first set of experiments we compare the PR2-Schulz method with the Richardson-NEQ method, and stop the iterations when the norm of the residual is sufficiently small, i.e., when

$$\|A^T(Ax_k - b)\|_2 \leq \epsilon.$$

For all the other experiments, we stop the iterations when the relative norm of the residual with respect to the initial residual is sufficiently small, i.e., when

$$\|A^T(Ax_k - b)\|_2 / \|A^T(Ax_0 - b)\|_2 \leq \epsilon.$$

In Figures 4 and 5 we report the norm of the residual per iteration for PR2-Schulz and Richardson-NEQ for $A = DD11$ and $A = DD12$, respectively. We clearly observe that applying the classical Richardson method on the normal equations, as expected, is extremely slow and requires a prohibitive amount of iterations to reduce the norm of the residual to $\epsilon = 10^{-8}$. We can also observe that applying the preconditioning strategy based on Schulz method (PR2-Schulz) drastically reduce the number of required iterations to achieve the same high accuracy. It is worth mentioning that the computational cost per iteration for Richardson-NEQ is mainly two matrix-vector products, and the

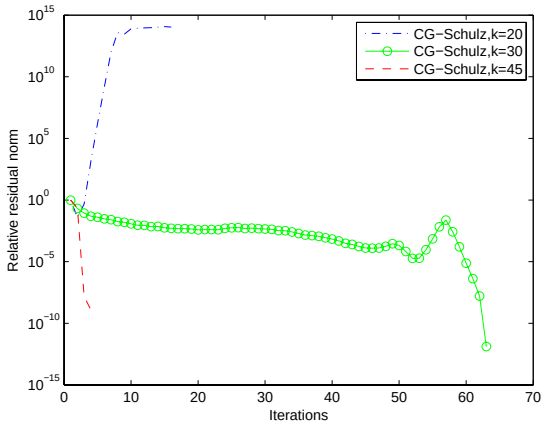

Figure 6: Norm of the relative residual of CG-Schulz for $A = DD13$ with three different values of k to build in advance the preconditioner M_k ($k = 20, 30, 45$).

computational cost per iteration for PR2-Schulz is mainly two rectangular matrix-matrix products (to obtain $M_k A M_k$). In spite of requiring less computational work per iteration, Richardson-NEQ is not a competitive option for accurately solving large dense linear least-squares problems.

Now we would like to consider the CG-Schulz method for solving least-squares problems. In this case, one important issue is the number of previous Schulz iterations required to build a suitable preconditioner in advance. In order to illustrate this important issue, we report in Figure 6 the norm of the relative residual per iteration of CG-Schulz for $A = DD13$ and three different values of k to build in advance the preconditioner M_k, $k = 20, 30, 45$. The most important observation is that the choice of the parameter k is critical. Notice that for the matrix $A = DD13$, $k = 20$ does not produce a sufficiently good approximation to the pseudo-inverse, and as a result, the sequence of relative residual norms generate overflow values and the process is abruptly stopped. On the other hand, $k = 45$ produces an excellent approximation to the pseudo-inverse and then, using M_{45} as a preconditioner, only 5 CG iterations are required. When $k = 30$, the CG-Schulz method converges but requires 62 iterations. Notice that in the CG-Schulz method the main computational cost is required for building the preconditioner M_k and, in that first stage, $2k$ rectangular matrix-matrix products are required, as in the PR2-Schulz method. Once the matrix M_k has been built, the required computational cost in the preconditioned CG method is almost irrelevant since they require only matrix-vector products.

Consequently, in order to compare the behavior of the PR2-Schulz method and the CG-Schulz method, different values of k to build the preconditioner in the CG-Schulz method need to be considered. In Figures 7 and 8 we report the norm of the relative residual per iteration of PR2-Schulz, and CG-Schulz for $A = DD14$ and $A = DD13$, respectively, and two different values of k to build in advance the preconditioner M_k using Schulz method. Notice that it is always possible to find a value of k for which the matrix M_k is good enough to guarantee that CG-Schulz requires very few CG iterations. However, as mentioned before, the computational cost for building such a good approximation to the pseudo-inverse could represent an excessive amount of work. It is also worth mentioning that the PR2-Schulz method represents a robust option that builds the matrix M_k dynamically, avoiding the choice of a critical parameter in advance. It is also interesting to observe, from Figures 7 and 8, that the PR2-Schulz method requires very few iterations to achieve low accuracy (e.g. $\epsilon = 10^{-4}$), and seems to be the best option in that case.

To close this section we would like to notice that using CG-Schulz with $k = 0$, where M_0 is given by (6), is exactly the same as using CG directly on the normal equations (2). This option has produced very poor results on all the considered test matrices. Therefore, based on our numerical experience,

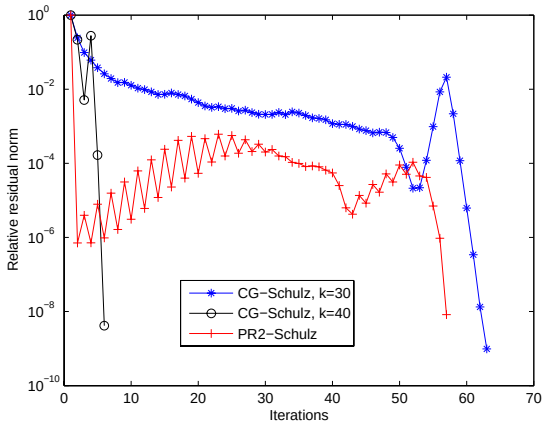

Figure 7: Norm of the relative residual of PR2-Schulz, and CG-Schulz for $A = DD13$ with two different values of k to build in advance the preconditioner M_k $(k = 30, 40)$.

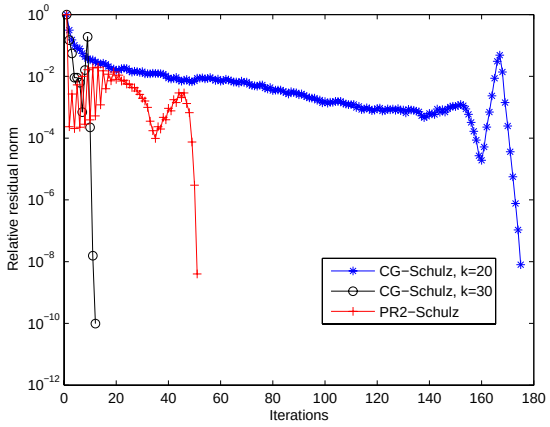

Figure 8: Norm of the relative residual of PR2-Schulz, and CG-Schulz for $A = DD14$ with two different values of k to build in advance the preconditioner M_k $(k = 20, 30)$.

CG on the normal equations is not a competitive option for dense ill-conditioned linear least-squares problems.

6 Final remarks

The main contribution of this work is the use of Schulz method to build rectangular preconditioners that can be embedded into some classical and well understood iterative schemes to accelerate the convergence when solving dense linear least-squares problems.

The Schulz preconditioning scheme was combined with an extension of Richardson's method (PR2-Schulz), producing a robust option for large and dense problems. It was also combined with the conjugate gradient method (CG-Schulz), and in this case the number of Schulz iterations needs to be defined in advance, which turns out to be a critical choice. For some choices of the number of preliminary Schulz iterations, the CG-Schulz method could be very competitive and convenient, and for some others it could not work at all. The optimal value of the number of Schulz iterations, in terms

of the required computational work, for the CG-Schulz method is a key issue that deserves further investigation.

The most significant computational cost in PR2-Schulz and also in CG-Schulz is the two rectangular matrix-matrix products per iteration. In order to reduce this cost to only one matrix-matrix product per iteration, we have explored a couple of options. Using the expression for $M_{k-1}A$ in (5) it follows that for any vector w

$$\begin{aligned} M_k w &= 2M_{k-1}w - \left(I - (I - M_0 A)^{2^{k-1}}\right) M_{k-1} w \\ &= M_{k-1}w + (I - M_0 A)^{2^{k-1}} w. \end{aligned} \tag{20}$$

To find a recursive expression of $d_{k+1} = M_{k+1} r^{(k+1)}$ for all k, we combine $r^{(k+1)} = r^{(k)} - \lambda_k A M_k r^{(k)}$ and equation (20), where $P_k \equiv (I - M_0 A)^{2^k}$, to obtain

$$\begin{aligned} d_{k+1} &= M_k r^{(k+1)} + P_k M_k r^{(k+1)} \\ &= M_k \left(r^{(k)} - \lambda_k A M_k r^{(k)}\right) + P_k M_k \left(r^{(k)} - \lambda_k A M_k r^{(k)}\right) \\ &= M_k r^{(k)} - \lambda_k M_k r^{(k)} + P_k M_k r^{(k)} + \lambda_k P_k^2 M_k r^{(k)} \\ &= (1 - \lambda_k)\, d_k + P_k d_k + \lambda_k P_k^2 d_k. \end{aligned}$$

A second and simpler option is based on the direct calculation of $B_k = M_k A$ and $z_k = M_k b$ independently instead of $M_k r_k$, using the following recursions: $B_{k+1} = B_k(2I - B_k)$, and $z_{k+1} = (2I - B_k) z_k$. We have run the same experiments reported in the previous section, with these two less expensive options, obtaining the same number of iterations of CG-Schulz and PR2-Schulz whenever the condition number of the matrix A is not too ill-conditioned. However, for very ill-conditioned problems, they have shown some numerical instability at the last iterations. For that reason we are not reporting with these two variants. Exploring the possibility of stabilizing these schemes is an interesting line of future research.

The extension of the Schulz preconditioning strategy for sparse problems is a very difficult task. On one hand, the matrix-matrix products in Schulz method produce significant fill-in from the beginning of the process, and therefore, the sparsity in the matrix $M_k A$ would be lost almost immediately. On the other hand, if we introduce a dropping strategy in Schulz method to preserve the sparsity pattern, then the orthogonality condition in Theorem 2.3 is lost. This orthogonality condition is the most important feature of the combined preconditioned schemes discussed in this work.

References

[1] M. Baboulin, L. Giraud, S. Gratton, and J. Langou. Parallel tools for solving incremental dense least squares problems: application to space geodesy. *Journal of Algorithms & Computational Technology*, 3:117–133, 2009.

[2] M. Benzi, L. Giraud, and G. All. Sparse approximate inverse preconditioning for dense linear systems arising in computational electromagnetics. *Numerical Algorithms*, 16:1–15, 1997.

[3] A. Björck. *Numerical methods for least squares problem.* SIAM, Philadelphia, 1996.

[4] C. Brezinski. Variations on Richardson's method and acceleration. *Bull. Soc. Math. Belg.*, pages 33–44, 1996.

[5] C. Brezinski. *Projection methods for systems of equations.* North Holland, Amsterdam, 1997.

[6] K. Forsman, W. Gropp, L. Kettunen, D. Levine, and J. Salonen. Solution of dense systems of linear equations arising from integral equation formulations. *Antennas and propagation magazine*, 37:96–100, 1995.

[7] R. W. Freud, G. H. Golub, and N. Nachtigal. Iterative solution of linear systems. *Acta Numerica*, 1:57–100, 1992.

[8] G. Golub and R. Plemmons. Large scale geodetic least squares adjustment by dissection and orthogonal decomposition. Technical report, Computer Sciences Department, Stanford University, 1979.

[9] B. Héron, F. Issard-Roch, and C. Picard. *Analyse numérique*. Dunod, Paris, 1999.

[10] A. S. Householder. *The Theory of Matrices in Numerical Analysis*. Dover Publications, Inc., New York, 1975.

[11] T. Kariya and H. Kurata. *Generalized Least Squares*. John Wiley and Sons Ltd., 2004.

[12] W. La Cruz and M. Raydan. Residual iterative schemes for large-scale nonsymmetric positive definite linear systems. *Computational and applied mathematics*, 49:151–173, 2008.

[13] M. Monsalve and M. Raydan. Newton's method and secant methods: A longstanding relationship from vectors to matrices. *Portugaliae Mathematica*, 68:431–475, 2011.

[14] G. Opfer and G. Schober. Richardson's iteration for nonsymmetric matrices. *Linear Algebra Appl*, 58:343–361, 1984.

[15] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, 2003.

[16] Y. Saad and M. Sosonkina. Enhanced preconditioners for large sparse least squares problem. Technical report, umsi-2001-1, Minnesota Supercomputer Institute, University of Minnesota, 2001.

[17] Y. Saad and H.A. Van der Vorst. Iterative solutions of linear systems in the 20th century. Technical report, Minnesota Supercomputer Institute, 1999.

[18] G. Schulz. Iterative berechnung del reziproken matrix. *Z. Angew. Math. Mech.*, 13:57–59, 1933.

[19] Y. Yan. Sparse preconditioned iterative methods for dense linear systems. *SIAM J. Sci. Comp*, 15:1190–1200, 1994.

[20] D.M. Young. On Richardson's method for solving linear systems with positive definite matrices. *J. Math Phys*, 32:243–255, 1954.