Electronic Transactions on Numerical Analysis.
Volume 29, pp. 97-115, 2008.
Copyright © 2007, Kent State University.
ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

A RANK-ONE UPDATING APPROACH FOR SOLVING SYSTEMS OF LINEAR EQUATIONS IN THE LEAST SQUARES SENSE[*]

A. MOHSEN[†] AND J. STOER[‡]

Abstract. The solution of the linear system $Ax = b$ with an $m \times n$-matrix A of maximal rank $\mu := \min(m, n)$ is considered. The method generates a sequence of $n \times m$-matrices H_k and vectors x_k so that the AH_k are positive semidefinite, the H_k approximate the pseudoinverse of A and x_k approximate the least squares solution of $Ax = b$. The method is of the type of Broyden's rank-one updates and yields the pseudoinverse in μ steps.

Key words. linear least squares problems, iterative methods, variable metric updates, pseudo-inverse

AMS subject classifications. 65F10, 65F20

1. Introduction.

We consider the problem of solving a rectangular system of linear equations $Ax = b$ in the least squares sense. Note that x solves the associated normal equations

$$A^T Ax = A^T b.$$

Here we suppose that the matrix $A \in R^{m\times n}$ has maximal rank $\mu := \min(m, n)$. In the case $m = n$ this amounts to solving a nonsingular system $Ax = b$ with a perhaps nonsymmetric matrix A by means of solving $A^T Ax = A^T b$. When A is symmetric positive definite (s.p.d.), the classical conjugate gradient method (cg) of Hestenes and Stiefel [13] belongs to the most powerful iterative methods. For solving general rectangular systems, a number of cg-type methods have been developed, see for instance [23]. In general solving such systems is more difficult than in the case of s.p.d. A. In this paper, we use rank-one updates to find approximations of the pseudoinverse of A, which, for instance, may be used as preconditioners for solving such systems with multiple right hand sides.

The use of a rank-one update to solve square systems with $A \in R^{n\times n}$ was studied by Eirola and Nevanlinna [8]. They invoked a conjugate transposed secant condition without line search. They proved that their algorithm terminates after at most n iteration steps, and if n steps are needed, then A^{-1} is obtained.

The problem was recently considered by Ichim [14] and by Mohsen and Stoer [17]. In [17], starting from an approximation H_0 for the pseudoinverse A^+, or from A^T when such an approximation is not available, their method uses *rank-2 updates* to generate a sequence of $n \times m$-matrices H_k approximating A^+. Two rank-2 updates were proposed. One is related to the DFP update and the other to the $BFGS$ update. Numerical results showed that both methods of [17], when used for the case $m = n$, give a better accuracy than the other rank-one update methods.

On the other hand, rank-one updating has attractive features in terms of computational requirements (lower number of operations, less storage). In this paper, rank-one updates (of Broyden's type) for matrices H_k are computed so that, in rather general situations, the sequence H_k terminates with the pseudoinverse in no more than μ steps. The properties of

[*] Received October 9, 2006. Accepted for publication September 3, 2007. Published online May 19, 2008. Recommended by M. Hochbruck.

[†] Department of Engineering Mathematics and Physics, Cairo University, Giza 12211, Egypt (amohsen@eng.cu.edu.eg). The work of this author was supported by the Alexander von Humboldt Stiftung, Germany.

[‡] Institut für Mathematik, Universität Würzburg, Am Hubland, D-97074 Würzburg, Germany (jstoer@mathematik.uni-wuerzburg.de).

the algorithm are established. For square systems, $m = n$, also the new method is compared with the methods CGN, CGS and $GMRES$, which try to solve $Ax = b$ directly.

An alternative algorithm utilizing the s.p.d. matrices $D_k := AH_k$ is proposed. It has the advantage that it can monitor the accuracy as $D_k \to I_m$. However, we then have to update two matrices (but note that the D_k are symmetric).

2. Rank-one updating.

We consider the solution of the linear system

$$Ax = b \tag{2.1}$$

in the least squares sense: $\bar{x}$ "solves" (2.1) if $\|b - A\bar{x}\| = \min_x \|b - Ax\|$, that is if $A^T(b - A\bar{x}) = 0$. We assume that $A \in R^{m\times n}$ has maximal rank, $\operatorname{rk} A = \mu := \min(m, n)$. Here and in what follows, $\|x\| := (x, x)^{1/2}$ is the Euclidean norm and $(x, y) := x^T y$ the standard scalar product in R^m. The vector space spanned by the vectors $u_i \in R^m$, $i = 1, 2, \dots, k$, is denoted by

$$[[u_1, u_2, \dots, u_k]].$$

We note already at this point that all results of this paper can be extended to complex linear systems with a complex matrix $A \in C^{m\times n}$. One only has to define the inner product in C^m by

$$(x, y) := x^H y.$$

Also the operator ^{T} then has to be replaced by ^{H} and the word "symmetric" by "Hermitian".

We call an $n \times m$-matrix H A-related if AH is symmetric positive semidefinite (s.p.s.d.) and $x^T AHx = 0$ implies that $x^T A = 0$ and $Hx = 0$. Clearly, $H := A^T$ is A-related, the pseudoinverse A^+ (in the Moore Penrose sense) is A-related (this can be shown using the singular value decomposition of A and A^+); if A is s.p.d. then $H = I$ is A-related. It is easily verified that any matrix H of the form $H = UA^T$, where $U \in R^{n\times n}$ is s.p.d., is A-related.

This concept will be central since our algorithms aim to generate A-related matrices H_k which approximate A^+. It also derives its interest from the special case when A is a nonsingular, perhaps nonsymmetric, square matrix. It is easily verified that a matrix H is A-related if and only if AH is symmetric positive definite. This means that H can be used as a right preconditioner to solve $Ax = b$ by means of solving the equivalent positive definite system $AHy = b$, $x = Hy$, using the *cg*-algorithm. Then the preconditioner H will be the better the smaller the condition number of AH is. So the algorithms of this paper to find A-related matrices H_k with good upper and lower bounds on the nonzero eigenvalues of AH_k can be viewed as algorithms for computing good preconditioners H_k in the case of a full-rank A.

The following proposition gives another useful characterization of A-related matrices:

PROPOSITION 2.1. *Let P be any unitary $m \times m$-matrix satisfying*

$$PA = \begin{bmatrix} \tilde{A} \\ 0 \end{bmatrix},$$

where $\tilde{A}$ is a $\mu \times n$-matrix of rank $\mu = \min(m, n) = \operatorname{rk} A$. Define for an $n \times m$- matrix H the $n \times \mu$-matrix $\tilde{H}$ and $\tilde{U}$ by

$$[\tilde{H}\ \ \tilde{U}] := HP^T.$$

Then H is A-related if and only if the following holds

$$\tilde{U} = 0 \text{ and } \tilde{A}\tilde{H} \text{ is s.p.d.,}$$

and then

$$PAHP^T = \begin{bmatrix} \tilde{A}\tilde{H} & 0 \\ 0 & 0 \end{bmatrix}.$$

Proof. 1. Suppose that H is A-related. Then the matrix

$$PAHP^T = \begin{bmatrix} \tilde{A}\tilde{H} & \tilde{A}\tilde{U} \\ 0 & 0 \end{bmatrix}$$

is s.p.s.d. . Hence $\tilde{A}\tilde{U} = 0$ and $\tilde{A}\tilde{H}$ is s.p.s.d.

Suppose $\tilde{U} \neq 0$. Then there is a vector y_2 with $\tilde{U}y_2 \neq 0$. Define $x := P^T y$, where $y^T = [y_1^T \;\; y_2^T]$, $y_1 := 0$. Then

$$x^T AHx = y^T PAHP^T y = [0 \;\; y_2^T] \begin{bmatrix} \tilde{A}\tilde{H} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ y_2 \end{bmatrix},$$

but

$$Hx = HP^T y = \begin{bmatrix} \tilde{H} & \tilde{U} \end{bmatrix} \begin{bmatrix} 0 \\ y_2 \end{bmatrix} \neq 0,$$

contradicting the A-relatedness of H. Hence $\tilde{U} = 0$. Now suppose that $\tilde{A}\tilde{H}$ is only positive semidefinite but not positive definite. Then there is a vector $y_1 \neq 0$ such that $\tilde{A}\tilde{H}y_1 = 0$. But then $y_1^T \tilde{A} \neq 0$, because $\tilde{A}$ has full row rank, and

$$[y_1^T \;\; 0] \begin{bmatrix} \tilde{A}\tilde{H} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ 0 \end{bmatrix} = 0.$$

Hence the corresponding

$$x := P^T \begin{bmatrix} y_1 \\ 0 \end{bmatrix}$$

satisfies

$$x^T AHx = [y_1^T \;\; 0] \begin{bmatrix} \tilde{A}\tilde{H} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ 0 \end{bmatrix} = 0,$$

but

$$x^T A = [y_1^T \;\; 0] \begin{bmatrix} \tilde{A}\tilde{H} & 0 \\ 0 & 0 \end{bmatrix} \neq 0,$$

again contradicting the A-relatedness of H.

2. Conversely, suppose that P, A and H satisfy the conditions of the proposition and assume $x^T AHx = 0$. Then with $y^T = [y_1^T, \, y_2^T] := x^T P^T$,

$$x^T AHx = [y_1^T \;\; y_2^T] \begin{bmatrix} \tilde{A}\tilde{H} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = 0,$$

so that $y_1 = 0$. But then

$$x^T A = y^T P A = [0 \;\; y_2^T] \begin{bmatrix} \tilde{A} \\ 0 \end{bmatrix} = 0$$

and likewise

$$Hx = HP^T y = [\tilde{H} \;\; 0] \begin{bmatrix} 0 \\ y_2 \end{bmatrix} = 0.$$

Hence H is A-related. □

In the case $\mu = m \leq n$, we can choose $P := I$ and the proposition reduces to the following corollary, which has a simple direct proof.

COROLLARY 2.2. *If the matrix A has full row rank, then H is A-related iff AH is a symmetric positive definite matrix.*

Proof. If AH is s.p.d., then $x^T AHx = 0$ gives $x = 0$. Conversely, if H is A-related, then $x^T AHx = 0$ implies $A^T x = 0$ and, therefore, $x = 0$ since A^T has full column rank. Hence the s.p.s.d. matrix AH is s.p.d.. □

We noted above that any matrix H of the form $H = UA^T$, where $U \in R^{n\times n}$ is s.p.d., is A-related. The converse is also true:

PROPOSITION 2.3. *A matrix $H \in R^{n\times m}$ is A-related iff it has the form $H = UA^T$, where $U \in R^{n\times n}$ is s.p.d..*

Proof. We prove the result only for the most important case of a full column rank matrix A, $m \geq n = \mu$.
Let H be A-related. Since $x^T AHx = 0$ implies $A^T x = 0$ and $Hx = 0$, $A^T x = 0$ is equivalent to $Hx = 0$, that is H and A^T have the form $H = UA^T$, $A^T = VH$ for some matrices U, $V \in R^{n\times n}$. As A^T has full column rank $n = \mu$, so has H. This implies that U is nonsingular and $V = U^{-1}$. Since $AH = AUA^T$ is s.p.s.d. and $AUA^T = (AUA^T)^T = AU^T A^T$, and therefore by the nonsingularity of AA^T

$$A^T AUA^T A = A^T AU^T A^T A \Longrightarrow U = U^T,$$

showing that U is symmetric, nonsingular and positive semidefinite, and therefore a s.p.d. matrix. □

The methods for solving (2.1) will be iterative. The new iterate x_{k+1} is related to the previous iterate by

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

where the search direction p_k is determined by the residual $r_k = b - Ax_k$ via an A-related $n \times m$ matrix H_k of rank μ by

$$p_k = H_k r_k.$$

We will assume $p_k \neq 0$, because otherwise

$$Ap_k = AH_k r_k = 0 \Rightarrow (AH_k r_k, r_k) = 0,$$

so that, by the A-relatedness of H_k, $A^T r_k = 0$, that is, x_k is a (least-squares) solution of (2.1).

We also note that $p_k \neq 0$ implies $Ap_k \neq 0$, because otherwise, again by A-relatedness,

$$0 = (Ap_k, r_k) = (AH_k r_k, r_k) \Rightarrow H_k r_k = p_k = 0.$$

Hence the scalar products (Ap_k, r_k), (Ap_k, Ap_k) will be positive.

Now, the stepsize α_k is chosen to minimize the Euclidean norm $\|r_{k+1}\|$. Thus,

$$r_{k+1} = r_k - \alpha_k Ap_k = r_k - z_k,$$

where

$$\alpha_k = (Ap_k, r_k)/(Ap_k, Ap_k) > 0.$$

Hence α_k is well defined and

$$\begin{aligned} (r_{k+1}, z_k) &= 0 = (r_{k+1}, AH_k r_k), \\ (r_{k+1}, r_{k+1}) &= (r_k, r_k) - |(r_k, Ap_k)|^2/(Ap_k, Ap_k) \\ &< (r_k, r_k). \end{aligned}$$

We update H_k using a rank-one correction

$$H_{k+1} = H_k + u_k v_k^T, \qquad u_k \in R^n, \quad v_k \in R^m.$$

Upon invoking the secant condition

$$H_{k+1} z_k = y_k,$$

we get the update formula of Broyden type [3],

$$\begin{aligned} H_{k+1} &= H_k + (y_k - H_k z_k) v_k^T/(v_k, z_k) \\ &= H_k + u_k v_k^T/(v_k, z_k), \qquad u_k := y_k - H_k z_k, \end{aligned} \tag{2.2}$$

provided that $(v_k, z_k) \neq 0$. The updates (2.2) are invariant with respect to a scaling of v_k. It can be shown (see [10]) that for $m = n$ the matrix H_{k+1} is nonsingular iff H_k is nonsingular and $(v_k, H_k^{-1} y_k) \neq 0$. If $m = n$ and A is s.p.d., then it is usually recommended to start the method with $H_0 := I$ and to use $v_k := u_k = y_k - H_k z_k$, which results in the symmetric rank-one method (SRK1) of Broyden [21] and leads to symmetric matrices H_k. However, it is known that SRK1 may not preserve the positive definitness of the updates H_k. This stimulated the work to overcome this difficulty [22], [16], [24], [15].

When A is nonsymmetric, the good Broyden (GB) updates result from the choice $v_k := H_k^T y_k$, while the bad Broyden (BB) updates take $v_k := z_k$. For $\alpha_k := 1$ and solving linear systems, Broyden [4] proved the local R-superlinear convergence of GB (that is, the errors $\epsilon_k := \|x_k - x^*\| \leq \theta_k$ are bounded by a superlinearly convergent sequence $\theta_k \downarrow 0$). In Moré and Trangenstein [18], global Q-superlinear convergence (i.e. $\lim_{k\to\infty} \epsilon_{k+1}/\epsilon_k = 0$) for linear systems is achieved using a modified form of Broyden's method. For $A \in R^{n\times n}$, Gay [10] proved that the solution using update (2.2) is found in at most $2n$ steps. The generation of $\{v_k\}$ as a set of orthogonal vectors was considered by Gay and Schnabel [11]. They proved termination of the method after at most n iterations.

The application of Broyden updates to rectangular systems was first considered by Gerber and Luk [12]. The use of the Broyden updates (2.2) for solving non-Hermitian square systems was considered by Deuflhard et al. [7]. For both GB and BB, they introduced an error matrix and showed the reduction of its Euclidean norm. For GB they provided a condition on the choice of H_0 ensuring that all H_k remain nonsingular. They also discussed the choice $v_k := H_k^T H_k z_k$, But in this case, no natural measure for the quality of H_k was available, which makes the method not competitive with GB and BB. For GB and BB, they showed the importance of using an appropriate line search to speed up the convergence.

In this paper, we start with an A-related matrix H_0 and wish to make sure that all H_k remain A-related. Hence the recursion (2.2) should at least preserve the symmetry of the AH_k. This is ensured iff

$$v_k = Au_k,$$

or equivalently

$$\begin{aligned} u_k &= y_k - H_k z_k = (I - H_k A) y_k = H_k(\alpha_k r_k - z_k) = H_k t_k, \\ v_k &= A(y_k - H_k z_k) = (I - AH_k) z_k = AH_k t_k, \end{aligned} \tag{2.3}$$

where t_k is the vector $t_k := \alpha_k r_k - z_k$.

However, the scheme (2.2), (2.3) may break down when $(v_k, z_k) = 0$ and, in addition, it may not ensure that AH_{k+1} is again A-related. We will see that both difficulties can be overcome.

3. Ensuring positive definiteness and A-relatedness.

For simplicity, we drop the iteration index k and replace $k+1$ by a star *. We assume that H is A-related and $p = Hr \neq 0$, so that $(AHr, r) > 0$ and $(Ap, Ap) > 0$. Following Kleinmichel [16] and Spedicato [24], we scale the matrix H by a scaling factor $\gamma > 0$ before updating. This leads to

$$H^* = \gamma H + uv^T/(v, z),$$

where $u = y - \gamma Hz = H(\alpha r - \gamma z) = Ht$, $v = Au = AHt$ and $t = \alpha r - \gamma z$. Therefore,

$$H^* = \gamma H + Ht(AHt)^T/(AHt, z), \tag{3.1}$$

$$AH^* = \gamma AH + AHt(AHt)^T/(AHt, z). \tag{3.2}$$

Then, introducing the abbreviations

$$\beta_1 := (AHr, r), \quad \beta_2 := (AHz, z), \quad \beta^* := (AHr^*, r^*),$$

we find

$$\beta_1 > 0, \quad \beta^* \geq 0, \quad \beta_2 \geq \beta_1 > 0,$$

$$\alpha = \frac{(Ap, r)}{(Ap, Ap)} = \frac{(AHr, r)}{(AHr, AHr)} > 0,$$

and an easy calculation, using $(r^*, z) = 0$, $z = r - r^*$, shows

$$\beta_2 = \beta_1 + \beta^*$$

and the following formula for the denominator in (3.2),

$$(v, z) = (AHt, z) = (\alpha AHr - \gamma AHz, z) = \alpha\beta_1 - \gamma\beta_2 = (\alpha - \gamma)\beta_1 - \gamma\beta^*. \tag{3.3}$$

Since H is A-related, Proposition 2.1 can be applied. Hence, there exist a unitary $m\times m$-matrix P and $\mu\times n$-matrices $\tilde{A}$, $\tilde{H}^T$, such that

$$PA = \begin{bmatrix}\tilde{A}\\ 0\end{bmatrix}, \quad HP^T = [\tilde{H}\;\; 0], \quad PAHP^T = \begin{bmatrix}\tilde{A}\tilde{H} & 0\\ 0 & 0\end{bmatrix}$$

.

$$\tilde{A}\tilde{H} \text{ is symmetric positive definite.}$$

Then (3.1), (3.2) preserve the block structure of HP^T and $PAHP^T$. Replacing t by

$$\tilde{t} = \begin{bmatrix}\tilde{t}_1\\ \tilde{t}_2\end{bmatrix} := Pt, \quad \tilde{t}_1 \in R^{\mu},$$

we find

$$[\tilde{H}^*\;\; 0] := H^*P^T = \gamma[\tilde{H}\;\; 0] + \frac{\tilde{H}\tilde{t}_1\,\tilde{t}_1^T[\tilde{A}\tilde{H}\;\; 0]}{(AHt, z)},$$

$$\begin{bmatrix}\tilde{A}\tilde{H}^* & 0\\ 0 & 0\end{bmatrix} := PAH^*P^T = \gamma\begin{bmatrix}\tilde{A}\tilde{H} & 0\\ 0 & 0\end{bmatrix} + \frac{\begin{bmatrix}\tilde{A}\tilde{H}\\ 0\end{bmatrix}\tilde{t}_1\,\tilde{t}_1^T[\tilde{A}\tilde{H}\;\; 0]}{(AHt, z)}.$$

Hence, by Proposition 2.1, AH^* is A-related if and only if

$$\tilde{A}\tilde{H}^* = \gamma\tilde{A}\tilde{H} + \frac{\tilde{A}\tilde{H}\tilde{t}_1(\tilde{A}\tilde{H}\tilde{t}_1)^T}{(AHt, z)} \tag{3.4}$$

is positive definite.

Since $\tilde{A}\tilde{H}$ is s.p.d. and $\gamma > 0$, $(\gamma\tilde{A}\tilde{H})^{1/2}$ is well defined and $\tilde{A}\tilde{H}^*$ satisfies

$$(\gamma\tilde{A}\tilde{H})^{-1/2}(\tilde{A}\tilde{H}^*)(\gamma\tilde{A}\tilde{H})^{-1/2} = I + \frac{(\tilde{A}\tilde{H})^{1/2}\tilde{t}_1\,\tilde{t}_1^T(\tilde{A}\tilde{H})^{1/2}}{\gamma(AHt, z)} =: U. \tag{3.5}$$

The matrix U has the eigenvalues 1 (with multiplicity $\mu - 1$) and, with multiplicity 1, the eigenvalue

$$\varphi(\gamma) := 1 + \frac{1}{\gamma}\,\frac{(\tilde{A}\tilde{H}\tilde{t}_1, \tilde{t}_1)}{(AHt, z)}.$$

But as is easily seen, $(\tilde{A}\tilde{H}\tilde{t}_1, \tilde{t}_1) = (AHt, t)$, so that by (3.3),

$$\varphi(\gamma) = \frac{1}{\gamma}\,\frac{\alpha(\alpha-\gamma)\beta_1}{\alpha\beta_1 - \gamma\beta_2}. \tag{3.6}$$

Hence, H^* will be A-related, iff $\gamma > 0$ satisfies $\alpha\beta_1 - \gamma\beta_2 \neq 0$ and

$$\frac{\alpha(\alpha-\gamma)\beta_1}{\alpha\beta_1 - \gamma\beta_2} > 0,$$

that is iff

$$0 < \gamma < \frac{\alpha\beta_1}{\beta_2} = \alpha\frac{\beta_1}{\beta_1 + \beta*} \quad \text{or} \quad \gamma > \alpha. \tag{3.7}$$

In view of the remark before Corollary 4.5 (see below), the choice $\gamma = 1$ seems to be favorable. By (3.7), this choice secures that H^* is A-related unless α, β_1, β^* satisfy

$$1 \le \alpha \le 1 + \frac{\beta^*}{\beta_1}. \tag{3.8}$$

Next we derive bounds for the condition number of $\tilde{A}\tilde{H}^*$ and follow the reasoning of Oren and Spedicato [20]. Let $\tilde{D} := \tilde{A}\tilde{H}$ and $\tilde{D}^* := \tilde{D}\tilde{H}^*$. Then by (3.4),

$$\tilde{D}^* = \gamma\tilde{D} + \tilde{D}\tilde{t}_1(\tilde{D}\tilde{t}_1)^T/(AHt, z),$$

where we assume that $\tilde{D}$ is s.p.d. and γ satisfies (3.7), so that also $\tilde{D}^*$ is s.p.d. Then the condition number $\kappa(\tilde{D})$ with respect to the Euclidean norm is given by

$$\kappa(\tilde{D}) = \frac{\lambda_{\max}(\tilde{D})}{\lambda_{\min}(\tilde{D})} = \frac{\max_{x\neq 0} x^T\tilde{D}x/x^Tx}{\min_{x\neq 0} x^T\tilde{D}x/x^Tx}.$$

Then by (3.5) for any $x \neq 0$,

$$\begin{aligned}\frac{x^T\tilde{D}^*x}{x^Tx} &= \frac{x^T(\gamma\tilde{D})^{1/2}U(\gamma\tilde{D})^{1/2}x}{x^T\gamma\tilde{D}x}\,\frac{x^T\gamma\tilde{D}x}{x^Tx}\\ &\le \lambda_{\max}(U)\,\gamma\lambda_{\max}(\tilde{D}),\end{aligned}$$

and, similarly,

$$\frac{x^T\tilde{D}^*x}{x^Tx} \ge \lambda_{\min}(U)\,\gamma\lambda_{\min}(\tilde{D}).$$

Now, the eigenvalues of U are 1 and $\varphi(\gamma)$ is given by (3.6). Hence,

$$\begin{aligned}\lambda_{\max}(\tilde{D}^*) &\le \max(1, \varphi(\gamma))\,\gamma\,\lambda_{\max}(\tilde{D}),\\ \lambda_{\min}(\tilde{D}^*) &\ge \min(1, \varphi(\gamma))\,\gamma\,\lambda_{\min}(\tilde{D}),\end{aligned}$$

so that, finally,

$$\kappa(\tilde{D}^*) \le \Phi(\gamma)\kappa(\tilde{D}),$$

where

$$\Phi(\gamma) := \frac{\max(1, \varphi(\gamma))}{\min(1, \varphi(\gamma))}.$$

Note that $\tilde{D}^*$ depends on γ, $\tilde{D}^* = \tilde{D}^*(\gamma)$. Unfortunately, the upper bound for $\kappa(\tilde{D}^*)$ just proved is presumably too crude; in particular it does not allow us to conclude that $\kappa(\tilde{D}^*) < \kappa(\tilde{D})$ for certain values of γ. Nevertheless, one can try to minimize the upper bound by minimizing $\Phi(\gamma)$ with respect to γ on the set Γ of all γ satisfying (3.6). The calculation is tedious though elementary. With the help of MATHEMATICA one finds for the case $\beta^* > 0$, that is $\beta_2 > \beta_1$, that Φ has exactly two local minima on Γ, namely

$$\begin{aligned}\gamma_+ &:= \alpha\big(1 + \sqrt{\beta^*/\beta_2}\,\big) > \alpha,\\ \gamma_- &:= \alpha\big(1 - \sqrt{\beta^*/\beta_2}\,\big) < \alpha\frac{\beta_1}{\beta_2}.\end{aligned} \tag{3.9}$$

Both minima are also global minima, and they satisfy

$$\Phi(\gamma_+) = \Phi(\gamma_-) = \frac{\beta_1\sqrt{\beta^*/\beta_2} + \beta^*\big(1 + \sqrt{\beta^*/\beta_2}\,\big)}{\beta_1\sqrt{\beta^*/\beta_2} - \beta^*\big(1 - \sqrt{\beta^*/\beta_2}\,\big)} > 1.$$

4. The algorithm and its main properties. Now, let H_0 be any A-related starting matrix, and x_0 a starting vector. Then the algorithm for solving $Ax = b$ in the least squares sense and computing a sequence of A-related matrices H_k is given by:

Algorithm: *Let A be a $m \times n$-matrix of maximal rank, $b \in R^m$, H_0 be A-related, and $x_0 \in R^n$ a given vector.*

```
For $k = 0, 1, \dots$
    1. Compute the vectors $r_k = b - Ax_k$, $p_k := H_k r_k$.
       If $p_k = 0$ then stop.
    2. Otherwise, compute
```

$$\alpha_k := (Ap_k, r_k)/(Ap_k, Ap_k),$$
$$x_{k+1} := x_k + \alpha_k p_k, \quad y_k := x_{k+1} - x_k,$$
$$r_{k+1} := r_k - Ay_k, \quad z_k := r_k - r_{k+1}.$$

```
    3. Define
```

$$\beta_1 := (AH_k r_k, r_k) = (Ap_k, r_k), \quad \beta^* := (AH_k r_{k+1}, r_{k+1}).$$

```
    4. Set $\gamma_k := 1$ if (3.8) does not hold.
       Otherwise compute any $\gamma_k > 0$ satisfying (3.7), say by using (3.9),
```

$$\gamma_k := \begin{cases} \gamma_+, & \textit{if } \beta^* > 0, \\ \alpha_k(1 + \text{eps}), & \textit{if } \beta^* = 0, \end{cases}$$

```
       where eps is the relative machine precision.
       Compute $u_k := y_k - \gamma_k H_k z_k$, $v_k := Au_k$ and
```

$$H_{k+1} := \gamma_k H_k + u_k v_k^T/(v_k, z_k).$$

```
       Set $k := k + 1$ and goto 1.
```

We have already seen that step 2 is well-defined if $p_k \neq 0$, because then $(Ar_k, r_k) > 0$ and $Ap_k \neq 0$, if H_k is A-related. Moreover, $p_k = 0$ implies $A^T r_k = A^T(b - Ax_k) = 0$, i.e., x_k is a least-squares solution of (2.1), and the choice of γ_k in step 2 secures also that H_{k+1} is A-related. Hence, the algorithm is well-defined and generates only A-related matrices H_k. This already proves part of the main theoretical properties of the algorithm stated in the following theorem:

THEOREM 4.1. *Let A be real $m \times n$-matrix of maximal rank $\mu := \min(m, n)$, H_0 an A-related real $n \times m$-matrix and $x_0 \in R^n$. Then the algorithm is well-defined and stops after at most μ steps: There is a smallest index $l \leq \mu$ with $p_l = 0$, and then the following holds:*

$$||Ax_l - b|| = \min_x ||Ax - b||,$$

and in particular $Ax_l = b$ if $m \leq n$. Moreover

1. *H_j is A-related for $0 \leq j \leq l$.*
2. *$H_k z_i = \gamma_{i,k} y_i$ for $0 \leq i < k \leq l$, where*

$$\gamma_{i,k} := \begin{cases} \gamma_{i+1}\gamma_{i+2}\cdots\gamma_{k-1} & \textit{for } i < k-1, \\ 1 & \textit{for } i = k-1. \end{cases}$$

3. *$(r_k, z_i) = 0$ for $0 \leq i < k \leq l$.*
4. *$(z_k, z_i) = 0$ for $i \neq k \leq l$.*

5. $z_k \neq 0$ *for* $0 \leq k < l$.

Proof. Denote by (P_k) the following properties:
1) H_j is A-related for $0 \leq j \leq k$.
2) $H_j z_i = \gamma_{i,j} y_i$ for $0 \leq i < j \leq k$.
3) $(r_j, z_i) = 0$ for $0 \leq i < j \leq k$.
4) $(z_j, z_i) = 0$ for $0 \leq i < j < k$.
5) $z_j \neq 0$ for $0 \leq j < k$.

By the remarks preceding the theorem, the algorithm is well-defined as long as $p_k \neq 0$ and the matrices H_k are A-related. (P_k), 4) and 5) imply that the k vectors z_i, $i = 0, 1, \ldots, k-1$, are linearly independent and, since $z_i = Ay_i \in \mathcal{R}(A)$ and dim $\mathcal{R}(A) = \mu$, property (P_k) cannot be true for $k > \mu$. Therefore, the theorem is proved, if (P_0) holds and the following implication is shown:

$$(P_k),\ p_k \neq 0 \implies (P_{k+1}).$$

1) (P_{k+1}), 1) follows from the choice of γ_k that H_{k+1} is well-defined and A-related if is H_k.

2) To prove (P_{k+1}), 2) we first note that

$$H_{k+1} z_k = y_k$$

by the definition of H_k. In addition for $i < k$, (P_k) and the definition of $\gamma_{i,k}$ imply

$$H_{k+1} z_i = \gamma_k H_k z_i = \gamma_k \gamma_{i,k} y_i = \gamma_{i,k+1} y_i,$$

since $(z_k, z_i) = 0$ and

$$(AH_k z_k, z_i) = (z_k, AH_k z_i) = (z_k, A\gamma_{i,k} y_i) = (z_k, \gamma_{i,k} z_i) = 0.$$

This proves (P_{k+1}), 2).

3) Since $(r_{k+1}, z_k) = 0$, and for $i < k$,

$$(r_{k+1}, z_i) = (r_{i+1} + \sum_{j=i+1}^{k} z_j, z_i) = 0,$$

because of (P_k), 3), 4). This proves (P_{k+1}), 3).

4) By (P_{k+1}), 3) we have for $i < k$

$$(z_i, z_k) = (z_i, r_k - r_{k+1}) = 0.$$

Hence (P_{k+1}), 4) holds.

It was already noted that $p_k \neq 0$ implies $z_k \neq 0$. Hence $(P_{k+1}$, 5) holds.
This completes the proof of the theorem. □

REMARK 4.2. In the case $m \geq n = \mu$, the algorithm finds the solution $\bar{x} :=$ arg $\min_x$ $\|Ax - b\|$, which is *unique*. In the case $m = \mu < n$, the algorithm finds only *some* solution $\bar{x}$ of the many solutions of $Ax = b$, usually not the interesting particular solution of smallest Euclidean norm. This is seen from simple examples, when for instance the algorithm is started with a solution x_0 of $Ax = b$ that is different from the least norm solution: the algorithm then stops immediately with $\bar{x} := x_0$, irrespective of the choice of H_0.

The least squares, minimal norm solution x^* of $Ax = b$ can be computed as follows: let $\bar{b}$ be any solution of $Ax = b$ (computed by the algorithm) and compute y^* as the least squares solution of $A^T y = \bar{b}$ again by the algorithm (but with A replaced by A^T and b by $\bar{b}$). Then $x^* = A^T y^*$ is the least squares solution of $Ax = b$ with minimal norm.

We also note that the algorithm of this paper belongs to the large ABS-class of algorithms (see Abaffy, Broyden and Spedicato [5]). As is shown in [25], also the minimal norm solution of a $Ax = b$, $A \in R^{m\times n}$, $m = \mu < n$, can be computed by invoking suitable algorithms of the ABS-class twice.

COROLLARY 4.3. *Under the hypotheses of Theorem 4.1, the following holds. If the algorithm does not stop prematurely, that is if $\mu = \min(m,n)$ is the first index with $p_l = 0$, $l = \mu$, then*

$$\begin{aligned} AH_\mu &= ZDZ^{-1}, \quad \text{if } \mu = m \le n,\\ H_\mu A &= YDY^{-1}, \quad \text{if } \mu = n \le m, \end{aligned}$$

with the matrices

$$\begin{aligned} Z &:= [z_0 \;\; z_1 \;\; \dots \;\; z_{\mu-1}],\\ Y &:= [y_0 \;\; y_1 \;\; \dots \;\; y_{\mu-1}],\\ D &:= \operatorname{diag}(\gamma_{0,\mu}, \gamma_{1,\mu}, \gamma_{\mu-1,\mu}). \end{aligned}$$

That is, if $\gamma_k = 1$ for all k, then $D = I$ so that H_μ is a right-inverse of A if $\mu = m \le n$, and a left-inverse if $\mu = n \le m$.

Proof. Part 2. of Theorem 4.1 and the definitions of D, Z, Y imply

$$H_\mu Z = YD, \quad AY = Z. \tag{4.1}$$

The μ columns $z_i \in R^\mu$ of Z are linearly independent by parts 4. and 5. of Theorem 4.1 (they are even orthogonal to each other), so that by $AY = Z$ also the columns $y_i \in R^n$ of the matrix Y are linearly independent.

Hence, for $\mu = m \le n$, Z^{-1} exists and by (4.1)

$$AH_\mu Z = ZD \;\Rightarrow\; AH_\mu = ZDZ^{-1},$$

and for $\mu = n \le m$, Y^{-1} exists, which implies by (4.1)

$$H_\mu Z = H_\mu AY = YD \Rightarrow H_\mu A = YDY^{-1}. \qquad \square$$

As a byproduct of the corollary we note

$$\begin{aligned} AH_\mu A &= \begin{cases} ZDZ^{-1}A & \text{if } m \le n\\ AYDY^{-1} & \text{if } m \ge n \end{cases},\\ H_\mu AH_\mu &= \begin{cases} H_\mu ZDZ^{-1} & \text{if } m \le n\\ YDY^{-1}H_\mu & \text{if } m \ge n \end{cases}. \end{aligned}$$

If $D = I$ (i.e., $\gamma_k = 1$ for all k) these formulae reduce to

$$AH_\mu A = A, \quad H_\mu AH_\mu = H_\mu.$$

Since then, for $\mu = n \le m$, in addition, both $H_\mu A = I_m$ and AH_μ are symmetric, H_μ must be the Moore-Penrose pseudoinverse A^+ of A.

REMARK 4.4. Consider the special case $\mu = m = n$ of a nonsingular matrix A. Then, AH_n is s.p.d. and, by the corollary, $AH_n = ZDZ^{-1}$, so that the eigenvalues of AH_n are just the diagonal elements of D. Hence its condition number is $\kappa(AH_n) = \kappa(D)$. Since the algorithm chooses the scaling parameter $\gamma_k = 1$ as often as possible, AH_n will presumably have a relatively small condition number even if $D \neq I$.

Theorem 4.1, part 3. implies a minimum property of the residuals:

COROLLARY 4.5. *For $k < l$, the residuals satisfy*

$$\|r_{k+1}\| = \min_{\lambda_i} \|b - A(x_0 + \sum_{i=0}^{k} \lambda_i p_i)\|.$$

Proof. By the algorithm

$$\alpha_i A p_i = \alpha_i A H_i r_i = z_i,$$

so that

$$r_{k+1} = r_0 - \sum_{i=0}^{k} \alpha_i A p_i = r_0 - \sum_{i=0}^{k} z_i.$$

Hence

$$\Phi(\lambda_0, \lambda_1, \ldots, \lambda_k) := \|b - A(x_0 + \sum_{i=0}^{k} \lambda_i p_i)\|^2 = \|r_0 - \sum_{i=0}^{k} \frac{\lambda_i}{\alpha_i} z_i\|^2$$

is a convex quadratic function with minimum $\lambda_i = \alpha_i$, $i = 0, 1, \ldots, k$, since

$$\Phi'_{\lambda_i}(\alpha_0, \ldots, \alpha_k) = \frac{1}{\alpha_i}(r_{k+1}, z_i) = 0, \qquad i = 0, \ldots, k,$$

by Theorem 4.1, part 3. □

We note that the space $[[z_0, z_1, \ldots, z_k]]$ is closely related to the Krylov space

$$K_{k+1}(AH_0, r_0) := [[r_0, AH_0 r_0, \ldots, (AH_0)^k r_0]] \subset R^m$$

generated by the matrix AH_0 and the initial residual r_0:

THEOREM 4.6. *Using the notation of Theorem 4.1, the following holds for $0 \leq k < l$*

$$\begin{aligned} r_k &\in [[r_0, AH_0 r_0, \ldots, (AH_0)^k r_0]] = K_{k+1}(AH_0, r_0), \\ z_k &\in [[AH_0 r_0, \ldots, (AH_0)^{k+1} r_0]] = AH_0 K_{k+1}(AH_0, r_0), \\ [[z_0, z_1, \ldots, z_k]] &= [[AH_0 r_0, \ldots, (AH_0)^{k+1} r_0]], \end{aligned}$$

that is z_0, z_1, …, z_k is an orthogonal basis of $AH_0 K_{k+1}(AH_0, r_0)$ and

$$\dim AH_0 K_{k+1}(AH_0, r_0) = k + 1.$$

Proof. The assertions are true for $k = 0$, since

$$z_0 = Ay_0 = \alpha_0 AH_0 r_0 \in [[AH_0 r_0]].$$

Now

$$AH_k z_k \in [[AH_0 r_0, \ldots, (AH_0)^{k+2} r_0]], \qquad 0 \leq k < l,$$

$$r_{k+1} = r_k - z_k, \quad z_k \alpha_k A H_k r_k \in [[AH_k r_k]], \quad \alpha_k \neq 0.$$

For any vector $u \in R^m$

$$\begin{aligned} AH_{k+1}u &\in [[AH_k u]] + [[z_k]] + [[AH_k z_k]] \\ &\vdots \\ &\in [[AH_0 u]] + [[AH_0 r_0, \ldots, (AH_0)^{k+2} r_0]], \end{aligned}$$

$$[[z_0, z_1, \ldots, z_k]] \subset [[AH_0 r_0, \ldots, (AH_0)^{k+1} r_0]],$$

$$\dim\, [[AH_0 r_0, \ldots, (AH_0)^{k+1} r_0]] \leq k+1. \qquad \Box$$

5. Comparison with other methods and numerical results. The numerical tests are concentrated to the cases $m = n + 1$ and $m = n$ for the following two reasons: The first is that it is harder to solve least squares problems with $n \leq m$ with n close to m than those with $n \ll m$. The second is that in the case $m = n$, which is equivalent to solving a linear equation $Ax = b$ with a nonsingular and perhaps nonsymmetric matrix A, there are many more competing iterative methods, such as CGN, CGS, $GMRES$ than the rank-one ("$RK1$") method of this paper and the rank-2 methods ("$RK2$") of [17] that solve $Ax = b$ only indirectly by solving $A^T Ax = A^T b$. In all examples we take $\log_{10} \|r_k\|$ as measure of the accuracy.

EXAMPLE 5.1. Here we use that the algorithm of this paper is defined also for complex matrices A. As in [17] we consider rectangular complex tridiagonal matrices $A = (a_{jk}) \in C^{m \times n}$ that depend on two real parameters q and f and are defined by

$$a_{jk} := \begin{cases} 1 + i\,q, & \text{if } j = k, \\ -i\,f, & \text{if } k = j+1, \\ i\,f, & \text{if } k = j-1, \\ 0, & \text{otherwise.} \end{cases} \quad 1 \leq j \leq m,\ 1 \leq k \leq n.$$

In particular, we applied $RK1$ (starting with $x_0 := 0$ and $H_0 := A^H$) to solve the least squares problem with $m = 31$, $n = 30$, $q = 0.1$ and $f = 1$. The iteration was stopped as soon as

$$\mathrm{re}_k := \|r_k\| \|r_0\| \leq \epsilon, \quad \epsilon := 10^{-3}.$$

$RK1$ stopped after 24 iterations. Figure 5.1 (plotting $\log_{10} \mathrm{re}_k$ versus k) shows the typical behavior observed with cg-type methods (cf. Theorem 4.1).

At termination, $RK1$ produced only an approximation H^* of the pseudoinverse of A, since $RK1$ stopped already after $24 < 30 = n$ iterations. When $H_0 := H^*$ was then used to solve by $RK1$ the system $Ax = b'$ with a new right-hand side $b' \neq b$ (but the same starting vector $x_0 = 0$ and precision $\epsilon = 10^{-3}$), $RK1$ already stopped after 9 iterations (see the plot of Figure 5.2).

Other choices of the parameters q and f lead to least squares systems $Ax = b$, where $RK1$ (started with $H_0 := A^T$) did not stop prematurely: it needed n iterations and thus terminated with a final H^* very close to A^+. Not surprisingly, $RK1$ (started with $H_0 := H^*$) then needed only one iteration to solve the system $Ax = b'$ with a different right-hand side $b' \neq b$. .

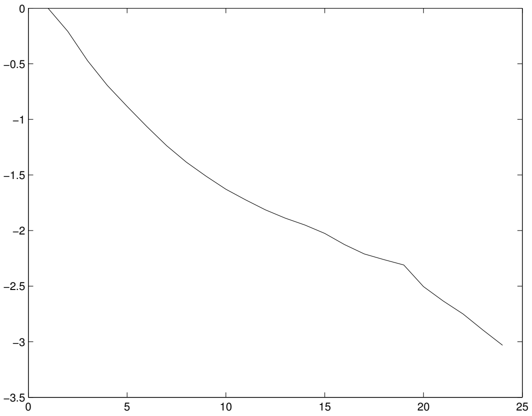

FIG. 5.1. *No preconditioning*

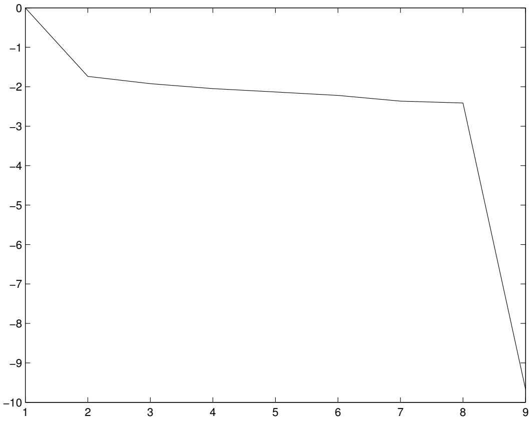

FIG. 5.2. *With preconditioning*

The test results of [17] for $RK2$ were similar. Since $RK1$ needs only half the storage and number of operations as $RK2$, $RK1$ is preferable.

EXAMPLE 5.2. This major example refers to solving square systems $Ax = b_i$, $i = 1, 2, \dots,$ with many right hand sides b_i. It illustrates the use of the final matrices $H^{(i)}$ produced by $RK1$ when solving $Ax = b_i$ as starting matrix $H_0 := H^{(i)}$ of $RK1$ for solving $Ax = b_{i+1}$.

We consider the same evolution equation as in [17],

$$u_t + a \cdot u_x + b \cdot u_y = u_{xx} + u_{yy} + f(x, y, t), \qquad 0 \le t \le T, \quad 0 \le x, y \le 1,$$

where

$$f(x, y, t) := e^{-\lambda t}[(-\lambda + 2\pi^2) \sin \pi x \, \sin \pi y + \pi(a \cos \pi x \, \sin \pi y + b \sin \pi x \, \cos \pi y)],$$

with the initial and boundary conditions

$$\begin{aligned} u(x, y, 0) &= \sin \pi x \, \sin \pi y, \qquad 0 \le x, y \le 1, \\ u(x, y, t) &= 0, \quad \text{for } t > 0,\ (x, y) \in \partial[0, 1] \times [0, 1]. \end{aligned}$$

Its exact solution is

$$u(x,y,t) = e^{-\lambda t} \sin \pi x \, \sin \pi y.$$

The space-time domain is discretized by a grid with space step $\Delta x = \Delta y = h = 1/N$ and time step $\Delta t = \tau > 0$. Let us denote the discretized solution at time level $n\,\tau$ by U, and at level $(n+1)\tau$ by V. Application of the Crank-Nicolson technique using central differences to approximate the first order derivatives and the standard second order differences to approximate the second derivatives yields the following scheme (we use the abbreviations $\beta := \tau/(2h^2)$, $\gamma := \tau/(4h)$) for the transition $U \rightarrow V$:

$$\begin{aligned}(1{+}4\beta)V_{ij} &+ V_{i+1,j}(a\gamma - \beta) + V_{i-1,j}(-a\gamma - \beta) + V_{i,j+1}(b\gamma - \beta) + \\ &+ V_{i,j-1}(-b\gamma - \beta) = \\ = (1 - 4\beta)U_{ij} &+ U_{i+1,j}(-a\gamma + \beta) + U_{i-1,j}(a\gamma + \beta) + U_{i,j+1}(-b\gamma + \beta) + \\ &+ U_{i,j-1}(b\gamma + \beta) + \frac{1}{2}[f(i\,h, j\,h, (n+1)\tau) + f(i\,h, j\,h, n\tau)],\end{aligned}$$

where $2 \le i,j \le N-1$. This scheme is known to be stable and of order $O(\tau^2, h^2)$. V is obtained form U by solving a linear system of equations $AV = C(U)$ with a nonsymmetric penta-diagonal matrix A and a right hand side $C(U)$ depending linearly on U. This linear system may be solved directly by using an appropriate modification of Gauss elimination. However, to demonstrate the use of our updates, we solve these equations for given U iteratively starting for the first time step with $H_0 := A^T$. For the next time steps, we compare the results for the rank-two method in [17] with the rank-one method of this paper.

For $N = 35$ (corresponding to more than 1000 unknowns at each time level) and $\Delta t = \tau = 0.01$, $a = 10$, $b = 20$, $\lambda = 1$ the solution of the problem up to the time $t = 5 \times \Delta t$ requires the following numbers of iterations in order to achieve a relative residual norm $<$ 0.0001:

RK1:	158	123	98	91	62
RK2:	158	123	109	87	74

These results show a comparable number of iterations for both methods. But again, the storage and the number of operations for $RK1$ is half that required for $RK2$.

The next examples also refer to the solution of systems $Ax = b$ with a square nonsingular $n \times n$-matrix A. Here we follow the ideas of Nachtigal et al. in [19], who compared three main iterative methods for the solution of such systems. These are CGN, $GMRES$ and CGS. Examples of matrices were given for which each method outperforms the others by a factor of size $O(\sqrt{n})$ or even $O(n)$. These examples are used to compare our algorithm ("$RK1$") with these three methods. In all examples $n = 40$, except Examples 5.3 and 5.6, where $n = 400$. We take $\log_{10} \|r_k\|$ as a measure of the accuracy. Note that $RK1$ solves $Ax = b$ only indirectly via the equivalent system $A^TAx = A^Tb$..

EXAMPLE 5.3. In this example all methods make only negligible progress until step n. Here

$$A := \text{diag}\,(1, 4, 9, \ldots, n^2),$$

is diagonal but represents a normal matrix with troublesome eigenvalues and singular values. The results show that both $GMRES$ and $RK1$ are leading.

EXAMPLE 5.4. This is an example where both CGN and $RK1$ are leading. Here A is taken as the unitary $n \times n$-matrix

$$A := \begin{bmatrix} 0 & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ 1 & & & & 0 \end{bmatrix}.$$

EXAMPLE 5.5. Here

$$A := \mathrm{diag}\,(d_1, d_2, \ldots, d_n)$$

where $\{d_j\}$ denote the set of Chebyshev points scaled to the interval $[1, \kappa]$, where

$$\begin{aligned} e_j &:= \cos\frac{(j-1)\pi}{n-1}, \\ d_j &:= 1 + (e_j + 1)(\kappa - 1)/2, \\ \kappa &:= \left(\frac{1 + \varepsilon^{1/(2\sqrt{n})}}{1 - \varepsilon^{1/(2\sqrt{n})}}\right)^2, \quad \varepsilon = 10^{-10}. \end{aligned}$$

A has condition number $O(n)$ and smoothly distributed entries. In this example CGS wins and both CGN and $RK1$ are bad.

EXAMPLE 5.6. Here A is the block diagonal matrix

$$A := \begin{bmatrix} M_1 & & & \\ & M_2 & & \\ & & \ddots & \\ & & & M_{n/2} \end{bmatrix}$$

with

$$M_j := \begin{bmatrix} 1 & j-1 \\ 0 & 1 \end{bmatrix}, \qquad j = 1, 2, \ldots, n/2,$$

which ensures a bad distribution of singular values. Here both CGS and $GMRES$ are leading, but $RK1$ outperforms CGN.

EXAMPLE 5.7. A is chosen as in Example 5.6, but with

$$M_j := \begin{bmatrix} 1 & j-1 \\ 0 & -1 \end{bmatrix}, \qquad j = 1, 2, \ldots, n/2.$$

This choice of M_j causes failure of CGS in the very first step. $GMRES$ is leading and $RK1$ outperforms CGN.

EXAMPLE 5.8. A is taken as in Example 5.6, but with

$$M_j := \begin{bmatrix} e_j & \gamma_j \\ 0 & \kappa/e_j \end{bmatrix},$$

where $\gamma_j := (\kappa^2 + 1 - e_j^2 - \kappa^2/e_j^2)^{1/2}$ and e_j and κ are defined as in Example 5.3. This leads to fixed singular values and varying eigenvalues.

Here $RK1$ and CGN lead and both are much better than CGS and $GMRES$.

EXAMPLE 5.9. A is taken as in Example 5.6, but with

$$M_j := \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

This matrix is normal and has eigenvalues $\pm i$ and singular value 1. Here both $RK1$ and CGN require only one iteration, $GMRES$ requires two while CGS fails.

The numerical results for Examples 5.1–5.7 are summarized in the next table listing the number of iterations needed for convergence (tested by $\log_{10} \|r_k\| \leq -10$). The number of iterations was limited to 50, and if a method failed to converge within this limit then the final accuracy is given by the number following \, e.g., \e-2 indicates that the final accuracy was $\|r_{50}\| \approx 10^{-2}$.

Example	CGN	CGS	GMRES	RK1
5.1	\e-6	\e-0	40	40
5.2	1	40	40	1
5.3	\e-4	20	38	50
5.4	\e-2	2	2	40
5.5	\e-2	Fails	3	40
5.6	2	20	38	2
5.7	1	Fails	2	1

We note that the residuals $\|r_k\|$ behave erratically for CGN. Its performance may be improved if one uses $\bar{r}_0 := A^T r_0$ rather than $\bar{r}_0 := r_0$ as taken in [19]. $RK1$ has common features with CGN: both perform excellently in Examples 5.2, 5.6 and 5.7, but $RK1$ avoids the bad performance of CGN in Examples 5.4, 5.5. $RK1$ never exceeded the maximum number of iterations.

No preconditioning was tried. It is obvious that if Examples 5.1 and 5.3 are scaled (to achieve a unit diagonal), one reaches the solution immediately within one iteration.

The comparison between these methods may include the number of matrix vector multiplications, of arithmetical operations, storage requirements and general rates of convergence. However, we chose only the number of iterations and the reduction of the norm of the residual as in [19] as indicators of performance. After comparing different iterative algorithms, Nachtigal et al. [19] concluded that it is always possible to find particular examples where one method performs better than others: There is not yet a universal iterative solver which is optimal for all problems.

6. Implementational issues and further discussion. There are several ways to realize the Algorithm. We assume that $H_0 = A^T$ and that A is sparse, but H_k ($k > 0$) is not. Then a direct realization of the Algorithm requires storage of a matrix $H \in R^{n\times m}$, and in each iteration, computation of 2 new matrix-vector products of the type $H_k w$ (neglecting products with the sparse matrices A and $H_0 = A^T$) and updating of H_k to H_{k+1} by means of the dyadic product $u_k v_k^T \in R^{n\times m}$ in Step 4 of the Algorithm. This requires essentially $3nm$ arithmetic operations (1 operation = 1 multiplication + 1 addition).

The arithmetic expense can be reduced by using the fact that any H_k has the form $H_k = U_k A^T$ (see Propositon 2.3), where $U_k \in R^{n\times n}$ is updated by

$$U_{k+1} = \gamma_k U_k + u_k u_k^T/(Au_k, z_k)$$

and each product $H_k w$ is computed as $U_k(A^T w)$. This scheme requires storage space for a matrix $U \in R^{n\times n}$ and only $3n^2$ operations/iteration, which is much less than $3nm$ if $n \ll m$.

Finally, if the Algorithm needs only few iterations, say at most $k \ll n$, one may store only one set of vectors u_i, $i \leq k$, explicitly (not two as with the rival method $RK2$ of [17]) and use the formula (supposing that $H_0 = A^T$)

$$H_{k+1} = \gamma_0\gamma_1\cdots\gamma_k \left(I + \sum_{i=0}^{k} \frac{u_i u_i^T}{(Au_i, z_i)\gamma_i} \right) A^T. \tag{6.1}$$

when computing products like $H_k z_k$ and $H_{k+1} r_{k+1}$ in the Algorithm. Using (6.1) to compute $H_k w$ requires essentially the computation of 2 matrix-vector products of the form

$$\bar{U}_k^T p, \quad \bar{U}_k q$$

with the matrix

$$\bar{U}_k := [u_0, \ldots, u_{k-1}] \in R^{n \times k},$$

which requires $2kn$ operations to compute a product $H_k w$. This again is half the number of operations as for the rank-2 methods in [17].

Our new update is connected to the previous one in the same way the SRK1 update is related to the DFP-update as shown by Fletcher [9] and Brodlie et al. [2]. We may define general updating expressions by

$$H_{k+1} = \gamma H_k + u_k (S u_k)^T / (S u_k, z_k).$$

This encompasses the case when A is s.p.d., which requires $S = I$ and $H_0 = I$, as well as the more general case in which $S = A$: this would require $H_0 = A^T$ or a better starting matrix, which secures that AH_0 is s.p.d. .

An alternative procedure is to update both H_k and the matrices $D_k := AH_k$ using

$$D_k^* = D_k + v_k v_k^T / (v_k, z_k)$$

This allows us to monitor the accuracy of the approximate inverse H_k by checking the size of $D_k - I$. For reasons of economy, one could monitor only the diagonal or some row of D_k.

The algorithm may be appropriate for solving initial value problems for partial differential equations as shown for rank-two updates in [17] and, for rank-one updates, in Example 5.2. In such problems, we start with the given initial value and iterate to find the solution for the next time level. The solution is considered to be acceptable if its accuracy is comparable to the discretization error of the governing differential equations.The final updated estimate of the inverse obtained for the present time level is then taken as the initial estimate of the inverse for the next time step.

REFERENCES

[1] C. Brezinski, M. Redivo-Zaglia, and H. Sadok, *New look-ahead Lanczos-type algorithms for linear systems,* Numer. Math, 83 (1999), pp. 53–85.

[2] K. W. Brodlie, A. R. Gourlay, and J. Greenstadt, *Rank-one and rank-two corrections to positive definite matrices expressed in product form,* J. Inst. Math. Appl., 11 (1973), pp. 73–82.

[3] C. G Broyden, *A class of methods for solving nonlinear simultaneous equations,* Math. Comput., 19 (1965), pp. 577–593.

[4] C, G. Broyden, *The convergence of single-rank quasi-Newton methods,* Math. Comput., 24 (1970), pp. 365–382.

[5] J. Abaffy, C. G Broyden, and E. Spedicato, *A class of direct methods for linear systems,* Numer. Math. 45 (1984), pp. 361–376.

[6] J. Y. Chen, D. R. Kincaid, and D. M. Young, *Generalizations and modifications of the GMRES iterative method,* Numer. Algorithms, 21 (1999), pp. 119–146.

[7] P. Deuflhard, R. Freund, and A. Walter, *Fast secant methods for the iterative solution of large nonsymmetric linear systems,* Impact Comput. Sci. Eng., 2 (1990), pp. 244–276.

[8] T. Eirola and O. Nevanlinna, *Accelerating with rank-one updates,* Linear Algebra Appl., 121 (1989), pp. 511–520.

[9] R. Fletcher, *A new approach to variable metric algorithms,* Computer J., 13 (1970), pp. 317–322.

[10] D. M. Gay, *Some convergence properties of Broyden's method,* SIAM J. Numer. Anal., 16 (1979), pp. 623–630.

[11] D. M. GAY AND R. B. SCHNABEL, *Solving systems of nonlinear equations by Broyden's method with projected updates,* in Nonlinear Programming, O. L. Mangasarian, R. R. Mayer, and S. M. Robinson, eds., Academic Press, New York, 1973, pp. 245–281.

[12] R. R. GERBER AND F. J. LUK, *A generalized Broyden's method for solving simultaneous linear equations,* SIAM J. Numer. Anal., 18 (1981), pp. 882–890.

[13] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems,* J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–436.

[14] I. ICHIM, *A new solution for the least squares problem,* Int. J. Computer Math., 72 (1999), pp 207–222.

[15] C. M. IP AND M. J. TODD, *Optimal conditioning and convergence in rank one quasi-Newton updates,* SIAM J. Numer. Anal., 25 (1988), pp. 206–221.

[16] H. KLEINMICHEL, *Quasi-Newton Verfahren vom Rang-Eins-Typ zur Lösung unrestringierter Minimierungsprobleme, Teil 1. Verfahren und grundlegende Eigenschaften,* Numer. Math., 38 (1981), pp. 219–228.

[17] A. MOHSEN AND J. STOER, *A variable metric method for approximating inverses of matrices,* ZAMM, 81 (2001), pp. 435–446.

[18] J. J. MORÉ AND J. A. TRANGENSTEIN, *On the global convergence of Broyden's method,* Math. Comput., 30 (1976), pp. 523–540.

[19] N. M. NACHTIGAL, S. C. REDDY, AND L. N. TREFETHEN, *How fast are nonsymmetric matrix iterations?,* SIAM J. Matrix Anal. Appl., 13 (1992), pp. 778–795.

[20] S. S. OREN AND E. SPEDICATO, *Optimal conditioning of self-scaling variable metric algorithms,* Math. Program., 10 (1976), pp. 70–90.

[21] J. D. PEARSON,*Variable metric methods of minimization,* Comput. J., 12 (1969), pp. 171–178.

[22] M. J. D. POWELL, *Rank one method for unconstrained optimization,* in Integer and Nonlinear Programming, J. Abadie, ed., North Holland, Amsterdam, 1970, pp. 139–156.

[23] Y. SAAD AND H. A VAN DER VORST, *Iterative solution of linear systems in the 20th century,* J. Comput. Appl. Math., 123 (2000), pp. 1–33.

[24] E. SPEDICATO, *A class of rank-one positive definite quasi-Newton updates for unconstrained minimization,* Optimization, 14 (1983), pp. 61–70.

[25] E. SPEDICATO, M. BONOMI, AND A. DEL POPOLO, *ABS solution of normal equations of second type and application to the primal-dual interior point method for linear programming,* Technical report, Department of Mathematics, University of Bergamo, Bergamo, Italy, 2007.