Accelerating with Rank-One Updates

Timo Eirola and Olavi Nevanlinna
Institute of Mathematics
Helsinki University of Technology
SF-02150 Espoo, Finland

Submitted by Michael Neumann

ABSTRACT

Consider the iteration $x_{k+1} = x_k + H(b - Ax_k)$ for solving $Ax = b$ (A is $n \times n$ nonsingular). We discuss rank-one updates to improve H as an approximation to A^{-1} during the iteration. The update kills and reduces singular values of $I - AH$ and thus speeds up the convergence. The algorithm terminates after at most n sweeps, and if all n sweeps are needed, then A^{-1} has been computed.

1. DERIVATION OF THE SCHEME

In this note we propose an acceleration scheme for iteration methods for solving linear systems of equations.

Let A be a nonsingular $n \times n$ matrix. Assume given a nonsingular H which approximates the inverse of A. Then the usual iteration for solving

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

can be written as

$$x_{k+1} = x_k + Hr_k, \tag{1.2a}$$

where

$$r_k = b - Ax_k. \tag{1.2b}$$

655 Avenue of the Americas, New York, NY 10010 0024-3795/89/$3.50

The residual satisfies

$$r_{k+1} = Er_k, \tag{1.3}$$

where $E := I - AH$. We shall update H during each step using rank-one corrections. Let us denote the H of (1.2a) by H_{k+1}, and similarly $E_{k+1} = I - AH_{k+1}$. We require

$$E_{k+1} = (I - c_k c_k^*)E_k, \tag{i}$$

where $c_k c_k^*$ is some rank-one projection chosen in such a way that

$$r_{k+1} = E_{k+1} r_k \approx 0. \tag{ii}$$

Consider rank-one updates of H_k:

$$H_{k+1} = H_k + u_k v_k^*. \tag{1.4}$$

We obtain

$$E_{k+1} = E_k - Au_k v_k^*, \tag{1.5}$$

which satisfies (i) with $c_k = Au_k/|Au_k|$ iff

$$v_k = \frac{1}{|Au_k|^2} E_k^* A u_k. \tag{1.6}$$

($|\cdot|$ denotes the Euclidean length.) It remains to choose u_k. If we could require $Au_k = E_k r_k$, then we would have $r_{k+1} = 0$. However, our best knowledge of the inverse of A is H_k, and hence a natural choice is

$$u_k = H_k E_k r_k. \tag{1.7}$$

The algorithm then reads:

1. Read x_0, compute r_0, read H_0, set $k = 0$.
2. Set

$$u_k = H_k(I - AH_k)r_k,$$

$$v_k = \frac{1}{|Au_k|^2}(I - H_k^* A^*)Au_k.$$

3. Set

$$H_{k+1} = H_k + u_k v_k^*,$$

$$x_{k+1} = x_k + H_{k+1} r_k,$$

$$r_{k+1} = b - A x_{k+1}.$$

4. Stop e.g. if r_{k+1} is small enough; otherwise return to step 2.

This form of the algorithm is suitable for understanding and analysis. The actual implementation to save computational work is discussed in chapter 3. It seems that this algorithm is new. At least it is not equivalent to the biconjugate gradient (BCG, [1]), the generalized minimum residual (GMRES, [2]), or the conjugate gradient squared (CGS, [3]) algorithms. Choosing $u_k = H_k r_k$ above leads to an algorithm algebraically equivalent to GMRES.

Note that getting $(I - AH_k)r_k = 0$ and thus $u_k = 0$ above does not cause any problems, since then by omitting the update of H_k we get $r_{k+1} = 0$.

2. BASIC PROPERTIES

The following properties of the algorithm hold for all choices of H_0 and x_0 provided H_k stays nonsingular. Theorem 2.3 will give a sufficient condition for that.

THEOREM 2.1. *Assume H_k does not become singular and $E_k r_k \neq 0$ for $k \leqslant k_0$. Then:*

(a) $E_{k+1} = (I - P_k)E_0$, *where $P_k = \sum_{i=0}^{k} c_i c_i^*$ is the orthogonal projection onto the subspace spanned by $\{Au_0, \dots, Au_k\}$.*
(b) *The Frobenius norm of E_k decreases.*
(c) *The singular values of E_k do not increase.*
(d) *The method gives the solution in at most n steps.*
(e) *One has*

$$r_{k+1} = E_{k+1} E_k r_k. \tag{2.1}$$

Proof: (a): By induction. For $k = 0$ (a) holds trivially with $P_0 = c_0 c_0^*$. Since

$$Au_k = AH_k E_k r_k = E_k AH_k r_k = (I - P_{k-1}) E_0 AH_k r_k,$$

we note that c_k is in the range of $I - P_{k-1}$, i.e., $P_{k-1} c_k = 0$. Thus

$$(I - c_k c_k^*)(I - P_{k-1}) = I - P_{k-1} - c_k c_k^* = I - P_k,$$

where

$$P_k = P_{k-1} + c_k c_k^* = \sum_{i=0}^{k} c_i c_i^*$$

is the orthogonal projection onto the subspace spanned by $\{Au_0, \ldots, Au_k\}$.

(b): $E_{k+1}^* E_{k+1} = E_k^*(I - c_k c_k^*) E_k = E_k^* E_k - E_k^* c_k c_k^* E_k$. Taking the trace gives

$$|E_{k+1}|_F^2 = |E_k|_F^2 - |E_k^* c_k|^2,$$

and since $c_k = (I - E_k) E_k r_k / |(I - E_k) E_k r_k|$, we have

$$|E_k^* c_k| = \sup_{|x|=1} \langle E_k x, c_k \rangle \geqslant \left\langle E_k \frac{r_k - E_k r_k}{|r_k - E_k r_k|}, c_k \right\rangle = \frac{|AH_k E_k r_k|}{|AH_k r_k|} > 0.$$

(c): See [1, p. 270].

(e): Since $c_k = (E_k - E_k^2) r_k / |(E_k - E_k^2) r_k|$, we have

$$\begin{aligned} r_{k+1} &= (I - c_k c_k^*) E_k r_k = (I - c_k c_k^*)\left[E_k r_k - \left(E_k - E_k^2\right) r_k\right] \\ &= (I - c_k c_k^*) E_k^2 r_k = E_{k+1} E_k r_k. \end{aligned}$$

(d): By (a) and because c_k is in the range of E_k we have $\operatorname{rank}(E_{k+1}) = \operatorname{rank}(E_k) - 1$. Thus $E_{k+1} r_{k+1} = 0$ for some $k \leqslant n - 1$. ∎

The following shows the continuity of the algorithm w.r.t. initial choices.

THEOREM 2.2. *If* $\tilde{x}_k = x_k + \delta x_k$ *and* $\tilde{H}_k = H_k + \delta H_k$, *then*

$$|\tilde{H}_{k+1} - H_{k+1}| \leqslant \kappa(A)\{[1 + C_H(|E_k r_k| + |AH_k||r_k|)]|\delta H_k| + C_H|AH_k E_k||\delta x_k|\} + O(d^2)$$

$$|\tilde{x}_{k+1} - x_{k+1}| \leqslant \kappa(A)\{[|r_k| + C_x(|E_k r_k| + |AH_k||r_k|)]|\delta H_k| + [|E_k| + C_x|AH_k E_k|]|\delta x_k|\} + O(d^2),$$

where

$$C_H = \frac{|E_k|}{|AH_k E_k r_k|}, \qquad C_x = \frac{|E_k r_k|}{|AH_k E_k r_k|}, \quad \textit{and} \quad d = \max(|\delta H_k|, |\delta x_k|).$$

Proof: Dropping the index k, since $E_+ = (I - cc^*)E$, we have

$$\begin{aligned}
\tilde{H}_+ - H_+ &= A^{-1}[I - \tilde{E}_+ - (I - E_+)] \\
&= A^{-1}[(I - cc^*)E - (I - \tilde{c}\tilde{c}^*)\tilde{E}] \\
&= A^{-1}[(\tilde{c}\tilde{c}^* - cc^*)E + (I - \tilde{c}\tilde{c}^*)A\,\delta H], \\
\tilde{x}_+ - x_+ &= \tilde{x} + \tilde{H}_+(b - A\tilde{x}) - x - H_+(b - Ax) \\
&= A^{-1}(I - \tilde{c}\tilde{c}^*)(I - A\tilde{H})A\,\delta x \\
&\quad + A^{-1}[(\tilde{c}\tilde{c}^* - cc^*)Er + (I - \tilde{c}\tilde{c}^*)A\,\delta H r].
\end{aligned}$$

First,

$$|\tilde{c}\tilde{c}^* - cc^*| = \sqrt{1 - \langle \tilde{c}, c\rangle^2} = \sqrt{\frac{|A\delta u|^2}{|Au|^2} - \frac{\langle A\tilde{u}, A\,\delta u\rangle^2}{|A\tilde{u}|^2 |Au|^2}} \leqslant \frac{|A\,\delta u|}{|Au|}.$$

Further,

$$\begin{aligned}
A\,\delta u &= A\tilde{H}(I - A\tilde{H})\tilde{r} - AH(I - AH)r \\
&= -AHEA\,\delta x + A[\tilde{H} - \tilde{H}A\tilde{H} - H + HAH]\tilde{r} \\
&= -AHEA\,\delta x + A[\delta HE - (H + \delta H)A\,\delta H](r - A\,\delta x).
\end{aligned}$$

Thus

$$|\tilde{H}_+ - H_+| \leqslant |A^{-1}|\left(\frac{|A\delta u|}{|Au|}|E| + |A||\delta H|\right) + O(d^2)$$

$$\leqslant \kappa(A)\left\{|\delta H| + \frac{|E|}{|Au|}\left[|AHE||\delta x| + (|Er| + |AH||r|)|\delta H|\right]\right\} + O(d^2),$$

$$|\tilde{x}_+ - x_+| \leqslant |A^{-1}||E|A|\delta x| + |A^{-1}|\left(\frac{|A\delta u|}{|Au|}|Er| + |A||r||\delta H|\right) + O(d^2)$$

$$\leqslant \kappa(A)\left\{|r||\delta H| + |E||\delta x| + \frac{|Er|}{|Au|}\left[|AHE||\delta x| + (|Er| + |AH||r|)|\delta H|\right]\right\} + O(d^2).$$

∎

Note that above C_H can be big if $|E_k r_k|$ is small. This is due to the sensitivity of the new eigenvector of AH_{k+1} corresponding to the eigenvalue 1. C_x, however, can only be of size $|(AH_k)^{-1}|$, and when $|E_k r_k|$ is small, x_{k+1} will already be very close to the solution. This partially explains the stability of the algorithm.

COROLLARY 2.1. *Assume x_0 and $H_0 = A^{-1} + B$ with* $\operatorname{rank}(B) = q$ *are such that H_k stays nonsingular. Then the algorithm solves the problem in $k_0 \leqslant q$ steps. Further, starting with $\tilde{x}_0 = x_0$, $\tilde{H}_0 = A^{-1} + B + C$ with small enough $|C|$ implies $|\tilde{r}_{k_0}| \leqslant$* tol.

Proof. $\operatorname{rank}(E_0) = q$. Use the proof of Theorem 2.1(d). The claim for $\tilde{H}_0$ follows from Theorem 2.2. ∎

The next result gives a "safe" case:

THEOREM 2.3. *Assume $AH_0 + (AH_0)^*$ is positive definite. Then H_k does not become singular, and we have the following uniform bounds:*

$$\begin{aligned} |AH_k| &\leqslant \left(1 + |AH_0|^2 - \mu^2\right)^{1/2} \\ \left|(AH_k)^{-1}\right| &\leqslant \mu^{-1}, \end{aligned} \tag{2.2}$$

where

$$\mu^2 := \inf_{|x|=1} \frac{\langle x, AH_0 x\rangle^2}{|x|^2 + |AH_0 x|^2}. \tag{2.3}$$

Proof: First note that by (a) of Theorem 1 we have

$$AH_k = I - E_k = I - (I - P_k)E_0 = P_k + (I - P_k)AH_0.$$

LEMMA. *Let P be an orthogonal projection and $x, y \in \mathbf{R}^n$. Then*

$$\tfrac{1}{2}\left(|x|^2 + |y|^2 + \sqrt{\left(|x|^2 + |y|^2\right)^2 - 4\langle x, y\rangle^2}\right)$$

$$\geqslant |Px|^2 + |(I - P)y|^2$$

$$\geqslant \tfrac{1}{2}\left(|x|^2 + |y|^2 - \sqrt{\left(|x|^2 + |y|^2\right)^2 - 4\langle x, y\rangle^2}\right).$$

Proof. We have

$$|\langle x, y\rangle| = |\langle Px, Py\rangle + \langle (I - P)x, (I - P)y\rangle|$$

$$\leqslant |Px|\,|Py| + |(I - P)x|\,|(I - P)y|.$$

Since $\inf_{as + bt \geqslant r}(s^2 + t^2)$ with $a, b, r \geqslant 0$ equals $r^2/(a^2 + b^2)$, using

$$|(I - P)x|^2 + |Py|^2 = |x|^2 + |y|^2 - |Px|^2 - |(I - P)y|^2,$$

we get

$$m := |Px|^2 + |(I - P)y|^2 \geqslant \frac{\langle x, y\rangle^2}{|(I - P)x|^2 + |Py|^2} = \frac{\langle x, y\rangle^2}{|x|^2 + |y|^2 - m}.$$

Thus

$$m^2 - \left(|x|^2 + |y|^2\right)m + \langle x, y\rangle^2 \leqslant 0,$$

from which the inequalities follow. ∎

Now using this lemma we get for $|x| = 1$

$$|AH_k x|^2 \geqslant \frac{|x|^2 + |AH_0 x|^2}{2}\left(1 - \sqrt{1 - 4\left(\frac{\langle x, AH_0 x\rangle}{|x|^2 + |AH_0 x|^2}\right)^2}\right) \geqslant \mu^2,$$

from which the bound for $|(AH_k)^{-1}|$ follows. The other one is similar. ∎

If the assumption of the previous theorem is not fulfilled we can use the following simple check to avoid producing a singular H_k (e.g. by not updating).

PROPOSITION 2.1. *Assume H_k is nonsingular. Then H_{k+1} is singular if and only if $\langle c_k, E_0 r_k\rangle = 0$.*

Proof. Note first that singular H_k is equivalent to: E_k has eigenvalue 1. Further,

$$E_{k+1}\eta = \eta \quad \Leftrightarrow \quad (I - c_k c_k^*)E_k\eta = \eta$$

and because $c_k = \tau(E_k - E_k^2)r_k$ this is equivalent to

$$E_k\eta - \gamma\big(E_k - E_k^2\big)r_k = \eta \quad \text{i.e.} \quad E_k(\eta + \gamma E_k r_k) = \eta + \gamma E_k r_k,$$

where $\gamma = \tau^2\langle(E_k - E_k^2)r_k, \eta\rangle$. By asssumption this is equivalent to $\eta = -\gamma E_k r_k$, i.e.

$$1 = \tau^2\Big\langle\big(E_k - E_k^2\big)r_k, \; -E_k^2 r_k\Big\rangle$$

and since $\tau^{-2} = |(E_k - E_k^2)r_k|^2$ this happens exactly when $\langle c_k, E_k r_k\rangle = 0$. Further,

$$\langle c_k, E_k r_k\rangle = \big\langle (I - P_{k-1})c_k, (I - P_{k-1})E_0 r_k\big\rangle = \langle c_k, E_0 r_k\rangle. \qquad \blacksquare$$

PROPOSITION 2.2. *The algorithm is invariant under unitary transformation of coordinates.*

Proof. Let T be nonsingular, $A = T\tilde{A}T^{-1}$, $x_k = T\tilde{x}_k$, $b = T\tilde{b}$; then for $\tilde{r}_k = \tilde{b} - \tilde{A}\tilde{x}_k$ we find $r_k = b - Ax_k = T(\tilde{b} - \tilde{A}\tilde{x}_k) = T\tilde{r}_k$. Also, given $\tilde{H}_{k+1} = T^{-1}H_{k+1}T$, we obtain from $\tilde{x}_{k+1} = \tilde{x}_k + \tilde{H}_{k+1}\tilde{r}_k$ that $x_{k+1} = T\tilde{x}_{k+1}$ satisfies $x_{k+1} = x_k + H_{k+1}r_k$. Hence we only have to check the updating of H_k.

Assume therefore that $\tilde{H}_k = T^{-1}H_kT$. Then from $\tilde{u}_k = (I - \tilde{H}_k\tilde{A})\tilde{H}_k\tilde{r}_k$ we obtain $T\tilde{u}_k = (I - H_kA)H_kr_k$, i.e., we always have $u_k = T\tilde{u}_k$. However,

$$\tilde{v}_k = \frac{1}{|\tilde{A}\tilde{u}_k|^2}\big(I - \tilde{H}_k^*\tilde{A}^*\big)\tilde{A}\tilde{u}_k = T^{-1}v_k$$

for all $\tilde{u}_k$ exactly when T is unitary. Then, finally, again for unitary T only,

$$\tilde{u}_k \tilde{v}_k^* = T^{-1} u_k (T^{-1} v_k)^* = T^{-1} u_k v_k^* T,$$

which completes the proof. ■

3. ON IMPLEMENTATION

In practice the matrix H_0 is seldom available, only a routine is provided which performs the multiplication by it, e.g. H_0 could be the inverse of an incomplete LU-factorization of A. Thus we cannot add the rank-one updates to it (this would also destroy the possible sparsity), but we store the corresponding vectors separately. Also to save computational work it turns out to be more efficient to save the pairs (u_k, c_k) instead of (u_k, v_k).

We have implemented the algorithm in the following form:

At step k we have $x_k, r_k, u_0, \dots, u_{k-1}, c_0, \dots, c_{k-1}$. Then compute

1° $\alpha_i = \langle c_i, r_k - AH_0 r_k \rangle$ for $i = 0, \dots, k-1$

$$\eta = H_0 r_k + \sum_{i=0}^{k-1} \alpha_i u_i, \quad \xi = r_k - A\eta$$

$$\left(\text{or} \quad \xi = r_k - AH_0 r_k - \sum_{i=0}^{k-1} \alpha_i c_i \right)$$

2° $\beta_i = \langle c_i, \xi - AH_0 \xi \rangle$ for $i = 0, \dots, k-1$

$$u_k = \tau \left(H_0 \xi + \sum_{i=0}^{k-1} \beta_i u_i \right), \quad c_k = A u_k$$

$$\left(\text{or} \quad c_k = \tau \left(AH_0 \xi + \sum_{i=0}^{k-1} \beta_i c_i \right) \right)$$

where τ is such that $|c_k| = 1$

3° $x_{k+1} = x_k + \eta + u_k \langle c_k, \xi \rangle$

$$r_{k+1} = \xi - c_k \langle c_k, \xi \rangle$$

When $k = 0$ the computation of alphas and betas and the sums are omitted. The alternatives in 1° and 2° are used, when it is cheaper to compute a linear combination of $k+1$ vectors than multiply one by A. It is seen that at step k this version needs two multiplications by H_0, four (two) multiplications by A, $2k$ scalar products, and two (four) linear combinations of $k+1$ vectors.

To limit the memory requirement and/or computational work in GMRES it has shown to be useful to keep only a fixed number of vectors in the memory and throw some away during the iteration [4]. Here the same strategy can also be applied and for example the rules of throwing away the oldest ones or those with least contribution have turned out to be effective.

The preliminary experiments and comparisons with BCG, CGS, and GMRES have shown to the authors that this method can be an efficient alternative to those. The results of these experiments will be reported in more detail in a forthcoming paper.

REFERENCES

1 Fletcher, R., Conjugate gradient methods for indefinite systems, *Numerical Analysis*, G. A. Watson, ed., Lect. Notes in Math 506, Springer-Verlag, Berlin, 1976, pp. 73–89.

2 Saad, Y. and Schultz, M. H., GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 7:856–869 (1986).

3 Sonneveld, P., Wesseling, P., and de Zeeuw, P. M., Multigrid and conjugate gradient methods as convergence acceleration techniques, *Multigrid Methods for Integral and Differential Equations*, D. J. Paddon and H. Holstein, eds., Clarendon Press, Oxford, 1985, pp. 117–167.

4 Tismenetsky, M. and Efrat, I., A Modified Generalized Minimal Residual Algorithm for Nonsymmetric Linear Systems, IBM Israel, Technical Report 88.219, March 1987.

5 Wilkinson, J. H., *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

Received 22 January 1988; final manuscript accepted 13 January 1989