Reprojection of the Conjugate Directions in the ABS Class Part I

József Abaffy

Óbuda University, Institute of Applied Mathematics
H-1034 Budapest, Bécsi út 96/b, Hungary
abaffy.jozsef@nik.uni-obuda.hu

Abstract. In the paper we introduce a modification of the Kahan-Parlett "twice is enough" [20] algorithm for conjugate direction algorithms. We apply the developed algorithm for the computation of conjugate directions in three subclasses of the ABS methods. In this part of the paper we give theoretical results and a preliminary test result as well. In the second part of our paper we test some elements of Subclass S2, while in the third part Subclass S6 and S7 will be examined. In this last part we give the conclusions regarding the reprojection of conjugate directions in the ABS classes

Keywords: twice is enough, reprojection of conjugate direction methods, ABS methods

1 Introduction

A very important question in Linear Algebra is the error propagation of algorithms and monitoring the error during the execution. There are many works in this field. We refer to [3], [5], [7], [8], [9], [10], [11], [13], [14], [16], [20], [22], [23] , [24] and others.

The reprojection technic can be applied in every class of the ABS methods. This reprojection improves accuracy in general, but the rounding errors could vanish these improvements. Therefore, it is very important to set up conditions for the reprojection. For orthogonal vectors the "twice is enough" method was developed by Parlett and Kahan [20]. The reprojection technic cannot be applied trivially to the conjugate direction methods like Lánczos [18], [19], Hestenes Stiefel [15] and others.

In this paper we develop a reprojection algorithm to conjugate directions and we give the necessary theoretical results to apply it to different subclasses (S2 and S6) of the ABS class.

We consider the classical conjugate gradient problem. A is supposed to be symmetric positive definite matrix throughout this paper.

We have to mention that we do not consider A^TA or AA^T conjugate direction methods, such as the CG algorithm applied to normal equations, Craig's method, CR, GCR, ORTHOMIN, ORTHODIR, GMRES methods and so on. For these methods see, for example [12], [1]. These problems and algorithms will be considered in a following paper.

2 Theoretical background

In this section we give the conjugate reprojection version of the Parlett and Kahan (PK) [20] method using the ABS class [1].

2.1 Parlett-Kahan type reprojection of conjugate directions in the ABS class

First we present the scaled ABS class which will be used later frequently.

Let us consider the following scaled system

$V^TAx = V^Tb$

where $A \in \Re^{m,n}$, $V \in \Re^{m,m}$ is an arbitrary non-singular matrix, $b \in \Re^m$ and $x \in \Re^n$.

The class of the scaled ABS algorithm

Algorithm 1. *Step 1 Set* $x_1 \in \Re^n$, $H_1 = I \in \Re^{n,n}$ *where I is the unit matrix,* $i = 1$, *and* $iflag = 0$.

Step 2 Let $v_i \in \Re^n$ *be arbitrary save that* $v_1, ..., v_i$ *be linearly independent. Compute the residual error vector* $r_i = Ax - b$. *If* $r_i = 0$, *stop* x_i *solves the system. Otherwise, compute the scalar* $\tau_i = v_i^T r_i$ *and the vector* $s_i = H_iA^Tv_i$.

Step 3 If $s_i \neq 0$, *go to Step 4; if* $s_i = 0$ *and* $\tau_i = 0$, *set* $x_{i+1} = x_i$, $H_{i+1} = H_i$, $iflag = iflag + 1$, *and if* $i < m$, *go to Step 6; otherwise, stop; if* $s_i = 0$ *and* $\tau_i \neq 0$, *set* $iflag = -i$ *and stop.*

Step 4 Compute the search direction p_i *by*

$p_i = H_i^Tz_i$

where $z_i \in \Re^n$ *is arbitrary saving for* $z_i^TH_iA^Tv_i \neq 0$.

Step 5 Update the approximate of the solution by

$x_{i+1} = x_i - \alpha_i p_i$

where the step size α_i *is given by*

$$\alpha_i = \frac{\tau_i}{v_i^T A p_i}$$

if $i = m$*, stop and* x_{i+1} *is the solution of the equations.*

Step 6 Update the matrix H_i *by*

$$H_{i+1} = H_i - \frac{H_i A^T v_i * w_i^T H_i}{w_i^T H_i A^T v_i} \tag{1}$$

where w_i *is arbitrary but the denominator must be non-zero.*

Step 7 Set $i = i+1$ *and go to Step 2.*

The properties of this algorithm can be found in [1]. Further we do not use the index i only if it is necessary.

We shall use two formulas of the H^T projection matrix. One of them can be obtained from (1) and the other is

$$\overline{H}^T = I - W * Q^{-T} * V^T * A \tag{2}$$

where W and V contain $w_1, ..., w_i$ and projection vectors $v_1, ..., v_i$ respectively computed and $Q^{-T} = (W^T A^T V)^{-T}$ until the actual step. For (2) see formula (7.20) or (7.59) of [1].

Now we are able to present the ABS PK type conjugate direction method. First we define the error vectors. Let the error vector $e' = x' - p$ satisfy $e'^T A e' \leq \varepsilon z^T A z$ where x' is the approximation of $p = \overline{H}^T z$ and $e" = x" - p$ satisfy $e"^T A e" \leq \varepsilon z^T A z$ where $x"$ is the approximation of $p = \overline{H}^T x'$ and ε is some tiny positive ε independent of z and A.

Let κ be any fixed value in the range $[1/(0.83-\varepsilon), 0.83/\varepsilon]$. Using the notation of the algorithm "twice is enough" we give.

Algorithm 2. *ABS Conjugate Direction of Parlett Kahan type (ABS_CD_PK)*

Case 1. If $x'^T A x' > z^T A z/\kappa$ *accept* $x = x'$ *and* $e = e'$*, otherwise compute* $x" = \overline{H}^T x'$ *to get* $x"$ *with error* $e"$ *and go to Case 2.*

Case 2. If $x"^T A x" \geq x'^T A x'/\kappa$ *accept* $x = x"$ *and* $e = e"$.

Case 3. If $x"^T A x" < x'^T A x'/\kappa$ *accept* $x = 0$ *and* $e = -p$.

As in the different subclasses the projection matrix H is calculated with different formulas and the theorems which ensure the accuracy of the ABS_CD_PK algorithm will be given there.

2.2 The class of the conjugate direction ABS algorithm (S2)

In this section, we study the S2 subclass of scaled ABS algorithm. Instead of the original equation $Ax = b$, where $A \in \Re^{m,n}$, $b \in \Re^m$, $x \in \Re^n$ consider the scaled equations

$$V^T Ax = V^T b \tag{3}$$

where $V = (v_1, \cdots, v_m) \in \Re^{m,n}$ is a non-singular matrix. The subclass S2 generates conjugate directions is defined by the formula

$v_i = p_i$

Note that we still have two arbitrary vectors z_i and w_i.

We recall Theorem 8.6 of [1] which state that the S2 subclass generates conjugate directions.

Theorem 1. *Let A be symmetric and positive definite. Then the subclass S2 where $v_i = p_i$ generates A conjugate search vectors and the iterate x_{i+1} minimizes over the linear variety $x_1 + Span(p_1, ..., p_i)$ the convex quadratic function*

$F(x) = (x - x^*)^T A(x - x^*)$

where x^* is the unique minimum point of $F(x)$.

Note that it is a special case of Theorem 7.17 in [1].

Now we prove a theorem which shows the effect of the reprojection with ABS_CD_PK.

Theorem 2. *The vector x computed by the ABS_CD_PK algorithm ensures that*

$e^T Ae^T \leq \varepsilon z^T Az^T + O(\varepsilon^2)$

and

$\left|p_0^T Ax\right| \leq \kappa\varepsilon p_0^T Ap_0 x^T Ax + O(\varepsilon^2)$.

Proof. We present those steps of the proof only which use the H projection matrix. The other parts of the proof are the same as in [20].

Case 1.

$e^T Ae = e'^T Ae' \leq \varepsilon z^T Az$

$\left|p_0^T Ax\right| = \left|p_0^T Ax'\right| = \left|p_0^T A(e' - p)\right| = \left|p_0^T Ae' - p_0 Ap\right| = \left|p_0^T Ae'\right|$ because of the conjugacy the second term is zero. Now, by applying the Cauchy–Schwartz inequality we get $\left|p_0^T Ae'\right| \leq \left\|p_0^T A^{1/2}\right\| \left\|A^{1/2} e'\right\| = \left(p_0^T A^{1/2} A^{1/2} p_0\right) \left(e'^T A^{1/2} A^{1/2} e'\right)$

$= \left(p_0^T Ap_0\right) \left(e'^T Ae'\right) \leq \left(p_0^T Ap_0\right) \varepsilon \left(z^T Az\right) =$

$\left(p_0^T Ap_0\right) \varepsilon\kappa \left(x'^T Ax'\right) = \varepsilon\kappa \left(p_0^T Ap_0\right) \left(x^T Ax\right)$ because of the true branch of Case 1.

Case 2.

$\left|p_0^T Ax\right| = \left|p_0^T Ax"\right| = \left|p_0^T A(e"+p)\right| = \left|p_0^T Ae"+p_0^T Ap\right| = \left|p_0^T Ae"\right|$

as the second term is zero because of the conjugacy

$= \left(p_0^T A^{1/2}A^{1/2}p_0\right)\left(e"^T A^{1/2}A^{1/2}e"\right) \leq \left(p_0^T Ap_0\right)\varepsilon\left(x'Ax'\right)$

and again from the true branch we get

$\leq \left(p_0^T Ap_0\right)\varepsilon\left(x"Ax"\right) \leq \kappa\varepsilon\left(p_0^T Ap_0\right)(xAx)$

On the other hand

$$\left(e^T Ae\right) = (x"-p)^T A\,(x"-p) \tag{4}$$

where $p = \left(I - W*Q^{-T}*P^T*A\right)x'$ therefore

$x"-p = e"+p+\left(I-W*Q^{-T}*P^T*A\right)x'-p$

$= e"+\left(I-W*Q^{-T}*P^T*A\right)(e'+p)-p$ and because of the conjugacy

$= e"+p+\left(I-W*Q^{-T}*P^T*A\right)e'-p = e"+\overline{H}e'$. Substituting it in (4) we get

$\left(e^T Ae\right) = \left(e"+\overline{H}e'\right)^T A\left(e"+\overline{H}e'\right)$

$= e"^T Ae"+e'^T\overline{H}Ae"+e"A\overline{H}e'+e'^T\overline{H}A\overline{H}e'$

$\leq \frac{\varepsilon}{\kappa}z^T Az+\|e'\|\left\|\overline{H}\right\|\|A\|\|e"\|+\|e"\|\|A\|\left\|\overline{H}\right\|\|e'\|+$

$\|e'\|\left\|A\overline{H}\right\|\|A\|\left\|\overline{H}H\right\|\|e'\|.$

Suppose that $\left\|\overline{H}\right\| \leq K$ then

$\leq \frac{\varepsilon}{\kappa}z^T Az+K\|A\|\|e'\|\|e"\|+K\|A\|\|e'\|\|e"\|+K^2\|A\|^2\|e'\|^2 \leq$

$\frac{\varepsilon}{\kappa}z^T Az+2K\|A\|\varepsilon z^T Az*\varepsilon x'^T Ax'+K^2\|A\|^2\varepsilon^2(z^T Az)^2.$

As now $x'^T Ax' \leq \frac{1}{\kappa}z^T Az$ we can continue

$\leq \frac{\varepsilon}{\kappa}z^T Az+2K\|A\|\frac{\varepsilon^2}{\kappa}\left(z^T Az\right)^2+\varepsilon^2K^2\|A\|^2\left(z^T Az\right)^2 =$

$\frac{\varepsilon}{\kappa}z^T Az+2K\|A\|\varepsilon^2\left(z^T Az\right)^2\left(\frac{1}{\kappa}+K\|A\|\right) = \frac{\varepsilon}{\kappa}z^T Az+O(\varepsilon^2)$

$\leq \varepsilon z^T Az+O(\varepsilon^2)$

as $\kappa > 1$ will be suggested to use.

Case 3. As $\left|p_0^T Ax\right| = 0$, it is enough to prove that $b^T p = b^T a = 0$, where $a = (I-W*Q^{-T}*P^T*A)e'$ and $b^T = e'^T(W*Q^{-T}*V^T*A)$. Indeed,

$b^T p = e'^T(W*Q^{-T}*V^T*A)*\left(I-W*Q^{-T}*P^T*A\right)x'$

$= e'^T(W*Q^{-T}*V^T*A-W*Q^{-T}*V^T*A*W*Q^{-T}*P^T*A)x' = 0.$

The proof for the case $b^T a = 0$ is similar to $b^T p = 0$. □

Note that the term which contains ε^2 can influence the estimation if $\|A\|$ is big. This phenomena will be observed during the tests of different algorithms.

Consider now the symmetric matrix projection case.

Symmetric matrices H_i are obtained for example with $H_1 = I$ where I is the unit matrix and w_i given by

$$w_i = \frac{Ap_i}{\|H_i A p_i\|_2^2} \tag{5}$$

In this case (5) is well defined.

Theorem 3. *If* $q_i = \frac{H_i A^T p_i}{\|H_i A p_i\|_2}$. *Then* $q_i^T q_j = 0$ *for* $i, j = 1, ..., n$

Proof. Let $j < i$ be. Then

$$q_i^T q_j = \frac{p_i^T H_i^T H_j A^T p_j}{\|H_i A p_i\|_2^2} = p_i^T H_i^T A^T p_j \, \|H_i A p_i\|_2^2 = \frac{p_i^T H_i A^T p_j}{\|H_i A p_i\|_2^2} = 0$$

because H_i is symmetric matrix $Null(H_i) = \left\{A^T p_1, ..., A^T p_{i-1}\right\}$ and the denominator is different from zero. The same argument is valid for the case $j > i$. □

Let $Q_i = [q_1, ..., q_i]$ be then we can obtained a block form of the projection matrix H_{i+1}

$$H_{i+1} = H_1 - Q_i Q_i^T. \tag{6}$$

It is important to note that the conjugate directions $p_i, i = 1, ..., n$ are generated by orthogonal column vectors of Q_i. Now we can only choose vectors z_i arbitrary. As the matrix update (8.24) of [1] takes an important role in some algorithms we present it now:

$$H_{i+1} = H_i - \frac{H_i A^T p_i p_i^T}{p^T A p_i} \tag{7}$$

where we used the idempotency of H_i. We present the chosen cases both for the symmetric and non-symmetric matrix projection cases in PART II of our paper.

3 The Hegedűs-Bodócs (HB) class of biorthogonalization algorithms (S6)

In this section we consider the subclass S6 of the ABS class. The HB biorthogonalization algorithms was first published in [14]. Recently a more detailed paper in this topic was published [13]. The main results of this section is Theorem 8.30 of [1] which proves how the HB algorithms constitute a part of the ABS class.

Theorem 4. *Consider the HB recursions with basis vectors* s_i, q_i *satisfying condition*

$s_i^T S_i A Q q_i \neq 0$

for all possible i , where

$$S_i^T = I - \sum_{j=1}^{i-1} \frac{Au_j v_j^T}{v_j^T Au_j}$$

and

$$Q_i = I - \sum_{j=1}^{i-1} \frac{u_j v_j^T A}{v_j^T Au_j}$$

where

$v_j = S_j s_j$

and

$u_j = Q_j q_j$

for $j = 1,...,i-1$. *Consider the following parameter choices in the scaled ABS class:* $H_1 = I$, v_i *and* z_i *given by*

$v_i = S_i^T s_i$

$z_i = Q_i q_i$

and w_i *a multiple of* z_i. *Then these parameter choices are well defined and moreover the following identity is true*

$p_i = Q_i q_i$.

Note that

$H_i^T z_i = z_i$.

therefore, based on the theoretical results the reason of the multiplication z_i by the projection matrix H_i^T is to have the possibility of the reprojections. As we show in our next paper the reprojection gives much better accuracy for the HB conjugate algorithms too.

It is important to note, that in this paper we suppose that the matrix A is positive definite symmetric matrix, consequently $p_i = Q_i q_i = S_i^T s_i$ that is the arbitrary vectors $v_i = p_i$ are defined as in the previous section. It means that Theorem 2 is valid for the Subclass S6 too.

Note also that the vectors z_i are still arbitrary.

In all algorithms listed below we also inserted the ABS versions to simplify the implementation. Many different versions of the HB algorithms follow from Theorem 8.30 of [1]. In the following definitions, for the sake of brevity, we leave out the index i wherever it is possible.

Algorithm p=H_ABS(v,u,Repr) (p is the conjugate direction)

$ABSv = v$

$ABSz = u$

$ABSw = ABSz$

$p = H^T * ABSz$

$s = HA^T p$

$if\ abs(s) < 3eps\ then$ % linear dependency

$\quad disp('the\ matrix\ A\ is\ singular')$

$\quad stop$

$endif$

$if \quad Repr == 1\ \ then$ %Reprojection is needed if Repr equals to one

$\quad p = H^T p$

end

$ptp = ABSv * Ap$

$pp = p/ptp$

$H = H - \frac{HA^T * ABSv * p^T}{p^T * A^T * ABSv}$.

Now we consider the following cases:

A) Hestenes–Stiefel algorithm in S6 (HBHSABS). The algorithm is defined by formulas (8.124) , (8.125), and the vectors s_i and q_i are defined by (8.135) and (8.136) in [1].

Algorithm P=H_HS_ABS(A,b,Repr,ReprHB,HB)

where A, b define the linear system, $Repr, ReprHB$ and HB are control parameters, see below.

Step 1 Initialize: Choose $S_1 = Q_1 = C_1 = K_1 = E$ where E is the n-dimensional unit matrix.

Let $\nu = \tau = 1$ be.

Compute

$r_1 = b - A * x$;

$s_1 = r_1$;

$q_1 = r_1$;

Step 2 (cycle for the dimension)

for i=1,...,n

$v_i = S_i^T s_i; \qquad u_i = Q_i q_i$

if $ReprHB == 1$ (Reprojection if $ReprHB$ equals to one)

$v_i = S_i^T v_i \qquad u_i = Q_i u_i$

endif

if $HB == 1$ (use the original version of the HS method in [13]

$P(:,i) = \frac{u_i}{norm(u,2)}$ (store the conjugate direction vector)

else

call **p=H_ABS(v,u,Repr)**

$P(:,i) = \frac{p_i}{norm(p,2)}$ (store the conjugate direction vector)

endif.

Step 3 Compute S_{i+1}, and Q_{i+1} by

$S_{i+1} = S_i - \frac{Au_i * v_i^T}{v_i^T Au_i} \qquad Q_{i+1} = Q_i - \frac{u_i * v_i^T A}{v_i^T Au_i}$

Compute the next arbitrary s_{i+1} and q_{i+1} vectors

$s_{i+1} = s_i - \frac{\mu_i s_i^T Cs_i}{v_i^T Au_i} Au_i \qquad q_{i+1} = q_i - \frac{\tau_i q_i^T Kq_i}{v_i^T Au_i} A^T v_i$

endfor.

B) Version of the HS method (S6CioccoHSDM). The algorithm is defined by formulas (3.3), (3.4) and (2.15) of [13].

Algorithm P=H_HSDM_ABS(A,b,Repr,HB)

Step 1 Initialize: Choose the positive definite Hermitian matrices $C = K = E$ as preconditioners where E is the n-dimensional unit matrix. Let x be an arbitrary vector which is not the solution of the linear system of equations. As C and K are unit matrices they are omitted from the formulas below.

Compute

$r_1 = b - A * x;$

$v_1 = r_1$

$u_1 = r_1$

$q_1 = r_1;$

$x = x + \frac{v_1^T r_1}{v_1^T Au_1} u_1.$

Step 2

for i= 1 : n

if $HB == 1$ (use the original version of the HS method in [13]

$P(:,i) = \frac{u_i}{norm(u_i,2)}$ (store the conjugate direction vector)

else

call **p=H_ABS(v,u,Repr)**

$P(:,i) = \frac{p_i}{norm(p_i,2)}$ (store the conjugate direction vector)

endif

$r_{i+1} = r_i - \frac{r_{i1}^T * r_i}{v_i^T A u_i} A u_i$ $\qquad$ $q_{i+1} = q_i - \frac{q_i^T * q_i}{v_i^T A u_i} A^T v_i$

$v_{i+1} = r_{i+1} + \frac{r_{i+1}^T r_{i+1}}{r_i^T r_i} v_i$ $\qquad$ $u_{i+1} = q_{i+1} + \frac{q_{i+1}^T q_{i+1}}{q_i^T * q_i} u_i$

$x = x + \frac{v_{i+1}^T * r_{i+1}}{v_{i+1}^T A u_{i+1}} u_{i+1}$

endfor.

The next algorithm is an alternative numerical formulation of the previous one that is of H_HSDM_ABS. It is defined by formulas (2.2), (3.1) and (3.2) of (S6CioccoHSDMM).

Algorithm P=H_HSDMM_ABS(A,b,ReprHB,Repr,HB)

Step 1 Initialize: Define $PL = E$ and $PR = E$ where E is the n-dimensional unit matrix. Let x be an arbitrary vector which is not a solution of the linear system of equations. Compute

$r = b - A * x$

$rABS = -r$

$q = r$;

Step 2 (cycle for the dimension)

for i = 1 : n

$r = PLr$ $\qquad$ $q = PR^T q$

$v = PL^T r$ $\qquad$ $u = PRq$

if $ReprHB == 1$

$v = PL^T v$ $\qquad$ $u = PRu$

end

if $HB == 1$ (use the original version of the HS method in [13]

$P(:,i) = \frac{u_i}{norm(u_i,2)}$ (store the conjugate direction vector)

else

call **p=H_ABS(v,u,Repr)**

$P(:,i) = \frac{p_i}{norm(p_i,2)}$ (store the conjugate direction vector)

endif.

Step 3 update the matrices

$PL = PL - \frac{Auv^T}{v^T Au}$ $PR = PR - \frac{uv^T A}{v^T Au}$

end.

Note that the difference between the two algorithms from above is the reorthogonalization possibility in the second one. We shall have better accuracy in the solution with this reorthogonalization.

C) Lánczos type recursion in HB (S6CioccoLancz). The algorithm is defined by formulas (8.124) , (8.125), and the vectors s_i and q_i are defined by (8.139) and (8.140) in [1]. It is enough to define the basis vectors.

Algorithm H_Lánczos_ABS(A,b,Repr,HB)

Step 1 Initialize: Choose $S_1 = Q_1 = C_1 = K_1 = E$ where E is the n-dimensional unit matrix. As C_1 and K_1 are unit matrices they are omitted from the formulas below.

Let $\nu = \tau = 1$ be. Similarly we omit *nu* and τ from the formulas.

Compute

$r_1 = b - A * x$;

$s_1 = r_1$;

$q_1 = r_1$.

Step 2 (cycle for the dimension)

for i=1,...,n

$v_i = S_i^T s_i$; $u_i = Q_i q_i$

if $ReprHB == 1$ (reprojection if $ReprHB$ equals to one)

$v_i = S_i^T v_i$ $u_i = Q_i u_i$

endif

if $HB == 1$ (use the original version of the HS method in [13]

$P(:,i) = \frac{u_i}{norm(u,2)}$ (store the conjugate direction vector)

else

call **p=H_ABS(v,u,Repr,)**

$P(:,i) = \frac{p_i}{norm(p,2)}$ (store the conjugate direction vector)

endif.

Step 3 Compute S_{i+1}, and Q_{i+1} by

$S_{i+1} = S_i - \frac{Au_i * v_i^T}{v_i^T Au_i}$ $Q_{i+1} = Q_i - \frac{u_i * v_i^T A}{v_i^T Au_i}$

s_{i+1}, and q_{i+1} by

$$s_{i+1} = s_i - \frac{s_i^T q_i}{v_i^T A u_i} A^T v_i \qquad q_{i+1} = q_i - \frac{q_i^T q_i}{v_i^T A u_i} A u_i$$

endfor.

D) Method (S6Ciocco HSRM) defined by formulas (3.8), (3.9), (3.10) and (5.1) of [13]

Algorithm H_HSRM_ABS(A,b,Repr,HB)

Step 1 Initialize: Choose $PR = E, PQ = E$ where E is the n-dimensional unit matrix.

$v_1 = b - A * x$

$C = v_1^T v_1 E$

$K = v_1^T v_1 E.$

Step 2 (cycle for the dimension)

for k= 1 : n

if $k == 1$

$v_k = b - A * x \qquad rr_k = v_k$

$u_k = v_k$

if $HB == 1$

$P(:,k) = u_i / norm(u_i, 2)$

endif

else

$u_k = PQ u_k$

if $HB == 1$

$P(:,k) = u_i / norm(u_i, 2)$

endif

endif.

Step 3

$x = x + \frac{v_k^T v_k}{v_k^T A u_k} u_k$

$\lambda_i = v_k^T * v_k \qquad \varphi_i = \lambda_i$

$q_k^T = \frac{v_k^T APQ}{\varphi_i} \qquad PQ = PQ - \frac{K * q_k q_k^T}{q_k^T * K * q_k}$

if $HB == 0$

call **p=H_ABS(v,u,Repr,)**

$P(:,k) = \frac{p_i}{norm(p,2)}$ (store the conjugate direction vector)

endif

endfor.

E) Method (S6Ciocco LDM) defined by (3.32), (3.33) and (5.1) of [13]

Algorithm H_LDM_ABS(A,b,Repr,HB)

Step 1

for k = 1 : n

if $k == 1$

$r = b - A * x$ $q = r$

$v = q$ $u = r$

$au = A * u$ $av = A^T * v$

$alp = v\prime * au$ $bet = q\prime * r; sig = bet/alp$

$x = x + sig * u$

if $HB == 1$

$P(:,k) = u/norm(u,2)$

else

call **p=H_ABS(v,u,Repr,)**

$P(:,k) = \frac{p_k}{norm(p,2)}$ (store the conjugate direction vector)

end

else.

Step 2 Preparation for the next iteration

$r = r - sig * au$ $q = q - sig * av$

$bn = q\prime * r$ $rat = bn/bet$

$v = q + rat * v$ $u = r + rat * u$

$au = A * u$ $av = A^T * v$

$alp = v\prime * au$ $bet = bn; sig = bet/alp$

$x = x + sig * u$

if $HB == 1$

$P(:,k) = u/norm(u,2)$

else

call **p=H_ABS(v,u,Repr,)**

$P(:,k) = \frac{p_k}{norm(p,2)}$ (store the conjugate direction vector)

endif

endfor.

F) Finally algorithm (S6HS) is defined by $ABSv_i = ABSu_i = ABSz_i = u_i$ where $ABSv_i$,$ABSu_i$ and $ABSz_i$ are the ABS class free parameter vectors.

Remark. In subclass S7 if we choose $v_i = Ar_i$ then the residual vectors r_i are A conjugate. The reprojection of the projection vectors does not give direct effect of the residual vectors. Therefore, we think that the accuracy of the solution would not grow very much. We present the test results of Subclass S6 and S7 in the third part of our paper.

4 Original algorithms

We implemented the original Hestenes–Stiefel and Lanczos methods as well.

1) Hestenes–Stiefel method (HSCGMoriginal). See in [15] or page 125 of [1].

Algorithm HS

Step 1 Initialize. Choose x_1. Compute $r_1 = Ax_1 - b$. Stop if $r_1 = 0$, otherwise set $p_1 = r_1$ and $i = 1$.

Step 2. Update x_i by

$$x_{i+1} = x_i - \frac{p_i^T r_i}{p_i^T A p_i}.$$

Step 3. Compute the residual r_{i+1}. Stop if $r_{i+1} = 0$.

Step 4 Compute the search vector p_{i+1} by

$$p_{i+1} = r_{i+1} - \frac{p_i^T A r_{i+1}}{p_i^T A p_i} p_i.$$

Step 5 Increment the index i by one and go to Step 2.

2) Lánczos method (Lanczosoriginal). See [18], [19] or page 126 of [1].

Algorithm Lanczos

Step 1. Initialize. Choose x_1. Compute $r_1 = Ax_1 - b$. Stop if $r_1 = 0$, otherwise set $p_1 = r_1$, $p_0 = 0$ and $i = 1$.

Step 2. Update the estimate of the solution by

$$x_{i+1} = x_i - \frac{p_i^T r_i}{p_i^T A p_i}.$$

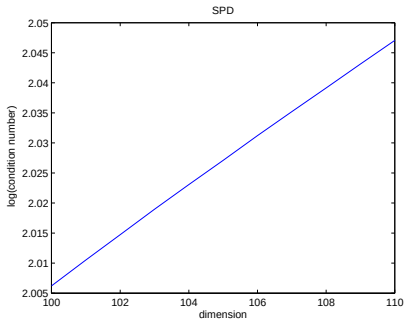
Step 3. Compute the residual r_{i+1}. Stop if $r_{i+1} = 0$. Step 4. Compute the search vector p_{i+1} by

$$p_{i+1} = Ap_i - \frac{p_i^T A^2 p_i}{p_i^T A p_i} p_i - \frac{p_{i-1}^T A p_i}{p_{i-1}^T A p_{i-1}} p_{i-1}$$

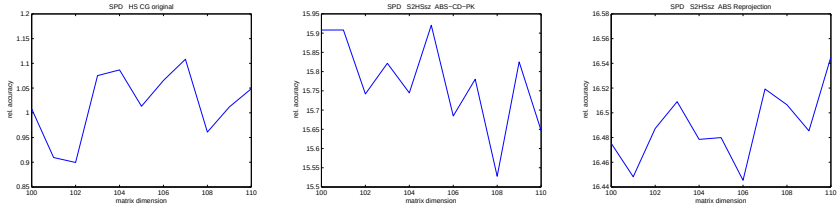
Step 5. Increment the index i by one and go to Step 2.

5 Preliminary test results

In this section we only show how the ABS_CD_PK algorithm works. We leave the intensive testing to the second and third part of the paper. To see the differences among the originals, the ABS_CD_PK conjugate directions method and the unconditional reprojection in the ABS methods we give an example. The algorithms were implemented in MATLAB version R2007b. The coefficient matrix is made by randomly generated Symmetric Positive Definite matrix (SPD) by the MATLAB rand function in [0,1]. Also the solutions of the constructed linear systems were generated randomly (by rand function) in [0,1]. The next figure shows the log(condition number) versus the considered dimension of the SPD problems



We chose the original Hestenes Stiefel, then the ABS_CD_PK algorithm with the Hestenes Stiefel method and the unconditional reprojection case in the ABS S2 subclass. The $\kappa = 100$ was chosen for the ABS_CD_PK algorithm which was suggested in [20]. The x axis shows the dimension while the y axis represents $y = -\log_{10}(yB)$ where $yB = \max abs(P^T AP - diag(P^T AP))/norm(A)$, where $norm(A)$ is the Frobenius norm of A.



where HS CG is the original Hestenes Stifel method see in [15] or page 125 of [20] for example. The name S2HSsz ($z_i = r_i$, $w_i = A^T p_i$) is the ABS symmetric version of it.

The norms of residuals in the solutions in case Hestenes Stiefel original are 3.826e-014 1.096e-013 6.628e-014 1.253e-013 6.889e-014 4.082e-014 7.418e-014 6.628e-

014 8.988e-014 5.64e-014 5.27e-014.

While in case S2HSsz ABS_CD_PK are 3.905e-013 4.353e-013 4.696e-013 4.187e-013 4.203e-013 4.264e-013 5.457e-013 4.942e-013 5.631e-013 6.169e-013 5.155e-013

The numbers of the reprojections are 31 35 44 38 38 41 41 35 49 38 44.

The numbers of linear dependency (ABS) are 10 17 19 22 10 16 10 8 17 14 13.

Finally in case of the unconditionally reprojecting method (ABS reprojection) the norms of residuals in the solutions are 4.092e-013 3.666e-013 5.04e-013 4.535e-013 4.49e-013 3.998e-013 5.749e-013 5.13e-013 4.951e-013 5.876e-013 5.498e-013

and the number of linear dependency (ABS) are 19 15 10 14 8 11 11 11 11 16 13.

The figures show the usefulness of the reprojection algorithm.

It can be seen that the S2HSsz ABS_CD_PK algorithm gives very good accurate results with much less number of computations than reprojections in every step.

Conclusion
In the paper we developed Parlett-Kahan's "twice is enough" algorithm for conjugate gradient case in the ABS class. The preliminary example shows the validity of our results. In the next two parts of the paper we intensively test many algorithms in the ABS class.

Acknowledgement
I would like to thank to my PhD student, Szabolcs Blága for his valuable help in writing this paper and to Csaba Hegedűs for the discussion on Subclass S6.

References

[1] Abaffy, J., and Spedicato, E., "*ABS Projection Algorithms: Mathematical Techniques for Linear and Nonlinear Equations"*, Ellis Horwood Limited, John Wiley and Sons, Chichester, England, (1989).

[2] Abaffy, J., Fodor, Sz., "*Reorthogonaliztaion methods in ABS classes*" under preparation

[3] Abdelmalek, N. N. "Round off error analysis for Gram-Schmidt method and solution of linear least squares problems" BIT 11 (1971) pp. 345-368

[4] Björck, A, "Solving linear least squares problems by Gram-Schmidt orthogonlaization" BIT 7 (1967) pp. 1-21

[5] Björck, A. and Paige, C. "Loss and recapture of orthogonality in the modified Gram-Schmidt algorithm", SIAM J. Matrix Anal. Appl. 13(1) (1992) pp. 176-190.

[6] Broyden, C.G. and Vespucci, M.T. "*Krylov Solvers for Linear Algebraic Systems*", Elsevier, (2004) ISBN 0-444-51474-0

[7] Galántai, A. and Hegedüs, C. J., "Jordan's principal angles in complex vector spaces", Numerical Linear Algebra with Applications 13 (2006) pp. 589-598 , http://dx.doi.org/10.1002/nla.491

[8] Giraud L., Langou J. and Rozloznik M. "On the round-off error analyisis of the Gram-Schmidt algrithm with reorthogonalization, CERFACS Technical Report No. TR/PA/02/33 (2002) pp. 1-11

[9] Giraud L., Langou J. and Rozloznik M. "The loss of orthogonality in the Gram-Schmidt orthogonalization process", Computers and Mathematics with Applications, Vol. 51 (2005) pp. 1069-1075,

[10] Golub, G. and van Loan, "*Matrix Computations*", 3rd ed. John Hopkins Univ. Press, Baltimore, MD (1966)

[11] Daniel, J.W, Gragg, W.B., Kaufman L. and Stewart G.W. "Reorthogonalization and Stable Algorithms for Updating the Gram-Scmidt QR Factorization", Mathematics of Computation Vol 30 No 136 (1976) pp. 772-795

[12] Dennis, J.E. JR and Turner, K. :"Generalized Conjugate Directions", Linear Algebra and its Application Vol 88/89 (1987) pp.187-209

[13] Hegedűs, C.J. : "Generation of Conjugate Directions for Arbitrary Matrices and Solution of Linear Systems", Proceedings of the NATO ASI Conference on Computer Algoirthms for Solving Linear Algebraic Systems, In Contributed papers of the NATO Advanced Study Institute Conference, Computer Algorithms for Solving Linear Algebraic Equations: The State of Art. (Sept. 9-22, 1990, Il Ciocco, Castelvecchio Pascoli, Tuscany, Italy.) (Eds. E. Spedicato and M. T. Vespucci), University of Bergamo, Bergamo, Italy, (1991) pp. 26-49.

[14] Hegedűs, C.J., and Bodócs, L." General Recursions for A-Conjugate Vector Pairs", Report No. 1982/56 Central Research Institute of Physiscs, Budapest (1982)

[15] Hestenes, M.R. and Stiefel, E.: "Methods of Conjugate Gradients for Solving Linear Systems" J. Res.Natlr. Bur. Stand. Vol 49 (1952) pp. 409-436

[16] Higham, N. J. *"Accuracy and Stabiility of Numerical Algorithms",* SIAM, Philadelphia, (1996)

[17] Hoffmann, W. "Iterative Algorithms for Gram_Schmidt orthogonalization" Computing Vol 41 (1989) pp. 335-348

[18] Lánczos, C. "An Iteration Method for the solution of the Eigenvalue Problem of Linear Differential and Integral Operators", J. Res.Natlr. Bur. Stand. Vol 45 (1950) pp. 255-282

[19] Lánczos, C. "Solution of Systems of Linear Equations by Minimized Iterations", J. Res.Natlr. Bur. Stand. Vol 49 (1952) pp. 33-53

[20] Parlett, B.N. "*The symmetric Eigenvalue Problem"*, Englewood Cliffs, N. J. Prentice-Hall (1980)

[21] Rice, J. R. "Experiments on Gram-Schmidt orthogonalization", Math. Comp. 20 (1966) pp. 325-328,

[22] Smoktunowicz, A. Barlow, J. L. and Langou, J. "A note on the error analysis of classical Gram-Schmidt", Numer. Math. 105/2, (2006) pp. 299-313,

[23] Wilkinson J. H. "*Rounding Errors in Algebraic Processes*", Prentice-Hall (1963)

[24] Wilkinson J. H. "*The Algebraiic Eigenvalue Problem",* Oxford University Press (1965)

[25] Zhang Liwei, Xia Zunquan and Feng Enmin, "*Introduction to ABS Methods in Optimization"*, Dalian University of Technology Press, (in chinese) (1998)