

Applications of Mathematics

Emilio Spedicato; Maria Teresa Vespucci

Variations on the Gram-Schmidt and the Huang algorithms for linear systems: A numerical study

Applications of Mathematics, Vol. 38 (1993), No. 2, 81–100

Persistent URL: <http://dml.cz/dmlcz/104537>

Terms of use:

© Institute of Mathematics AS CR, 1993

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

VARIATIONS ON THE GRAM-SCHMIDT AND THE HUANG
ALGORITHMS FOR LINEAR SYSTEMS: A NUMERICAL STUDY

EMILIO SPEDICATO, MARIA TERESA VESPUCCI, Bergamo

(Received August 22, 1990)

Summary. In this paper we compare the numerical performance on a set of ill conditioned problems of several algorithms for linear systems based upon the explicit QR factorization and the implicit LQ factorization associated with the Huang and the modified Huang algorithms in the ABS class. The results indicate that the modified Huang algorithm is generally more accurate than the Huang algorithm and competitive with commercial codes based upon the QR factorization with Householder or Givens reflections. The best version of the modified Huang algorithm performs similarly, as theoretically expected, to the doubly iterated Gram-Schmidt method of Daniel et al., applied on the rows to generate search vectors.

Keywords: ABS methods, Huang algorithm, QR algorithm, Gram-Schmidt orthogonalization

AMS classification: 65F05

1. INTRODUCTION

In a series of recent papers (Abaffy, Broyden and Spedicato (1984), Abaffy and Spedicato (1985), see also the monograph of Abaffy and Spedicato (1989)), a general class of algorithms of direct type, here referred to as the ABS class, has been introduced for solving the determined or undetermined linear system of equations

$$(1) \quad a_i^T x = b^T e_i, \quad 1 \leq i \leq m, \quad m \leq n, \quad x, a_i \in \mathbb{R}^n, \quad b, e_i \in \mathbb{R}^m$$

or, in matrix form

$$(2) \quad Ax = b.$$

In (1) e_i is the i -th unit vector in \mathbb{R}^m and in (2) $A = A_m^T$ where for $i = 1, \dots, m$

$$(3) \quad A_i = (a_1, \dots, a_i).$$

The ABS class of algorithms (extended to nonlinear systems in Abaffy, Galantai and Spedicato (1987a) and Abaffy and Galantai (1986) and to nonlinear optimization in Abaffy, Galantai and Spedicato (1987b)) is defined, in exact and sequential arithmetic, by the following procedure:

(A) Let x_1 be an arbitrary vector in \mathbb{R}^n . Let H_1 be an arbitrary n by n nonsingular matrix. Let v_1 be an arbitrary nonzero in \mathbb{R}^n . Set $i = 1$.

(B) Compute the vector $s_i \in \mathbb{R}^n$ by

$$(4) \quad s_i = H_i A^T v_i$$

and the scalar τ_i by

$$(5) \quad \tau_i = v_i^T (A x_i - b).$$

If $s_i \neq 0$, go to (C). If s_i and τ_i are zero, set $x_{i+1} = x_i$, $v_{i+1} = v_i$, $H_{i+1} = H_i$ and go to (G). If s_i is zero but τ_i is nonzero, stop (system incompatible).

(C) Define the search vector $p_i \in \mathbb{R}^n$ by

$$(6) \quad p_i = H_i^T z_i$$

where $z_i \in \mathbb{R}^n$ is arbitrary save that $z_i^T s_i \neq 0$.

(D) Update the estimate x_i of the solution by

$$(7) \quad x_{i+1} = x_i - \alpha_i p_i$$

where

$$(8) \quad \alpha_i = \frac{\tau_i}{z_i^T s_i}.$$

If $i = m$ stop (x_{m+1} solves the system).

(E) Update the matrix H_i by

$$(9) \quad H_{i+1} = H_i - s_i w_i^T H_i,$$

where $w_i \in \mathbb{R}^n$ is arbitrary save that

$$(10) \quad w_i^T s_i = 1.$$

- (F) Define the scaling vector $v_{i+1} \in \mathbb{R}^m$ as an arbitrary vector in \mathbb{R}^m subject to linear independence from v_1, \dots, v_i .
- (G) Increment the iteration index i by one and go to (B).

An important algorithm in the ABS class is the so called Huang algorithm, introduced by Huang (1975) in a paper which led to the development of the ABS class. The Huang algorithm is obtained by the choices $H_1 = I$, $v_i = e_i$ and

$$(11) \quad z_i = a_i, \quad w_i = \frac{a_i}{a_i^\top H_i a_i}$$

so that formulas (8), (9) read

$$(12) \quad \alpha_i = \frac{a_i^\top x_i - b^\top e_i}{a_i^\top p_i}$$

and

$$(13) \quad H_{i+1} = H_i - \frac{p_i p_i^\top}{a_i^\top p_i}$$

In this paper we consider several versions of the Huang algorithm, corresponding to various parameter choices and alternative formulations for the stepsize and the search vectors. We compare these algorithms with three commercial codes, based upon explicit QR factorization via Householder or Givens rotations, and with four algorithms based upon the stabilized or the reorthogonalized Gram-Schmidt orthogonalization procedure applied to the columns or to the rows of the matrix. The results indicate that the so called modified Huang algorithm is usually more accurate than the Huang algorithm and competitive with the considered commercial codes based upon the QR factorization. The best version of the modified Huang algorithm performs similarly to the algorithm where the search vectors are obtained by twice iterating the standard Gram-Schmidt algorithm on the rows of A and updating x_i via the ABS formula

$$(14) \quad x_{i+1} = x_i - \frac{a_i^\top x_i - b^\top e_i}{a_i^\top p_i} p_i.$$

2. PROPERTIES AND FORMULATION OF THE HUANG AND THE MODIFIED HUANG ALGORITHMS

In exact arithmetic the Huang algorithm has a number of remarkable properties that are recalled here, assuming, for the sake of simplicity, that A has full rank. For proofs and further discussions, see Abaffy and Spedicato (1989).

- (A1) The denominator $a_i^T H_i a_i = a_i^T p_i$ is strictly positive.
- (A2) The search vectors are orthogonal and coincide with the vectors obtained by applying the Gram-Schmidt orthogonalization procedure to the rows of A .
- (A3) If x_1 is an arbitrary multiple of a_1 (for instance $x_1 = 0$), then for $i = 1, \dots, n$, the vector x_{i+1} is the solution of minimal Euclidean norm of the first i equations of the given system and, moreover, the following inequalities are satisfied (in Euclidean norm):

$$(15) \quad \|x_i\| \leq \|x_{i+1}\|,$$

$$(16) \quad \|x_{i+1} - x^+\| \leq \|x_i - x^+\|,$$

where $x^+ = x_{m+1}$ is the solution of the system of minimal Euclidean norm.

- (A4) The matrices H_i are orthogonal projectors.

The modified Huang algorithm has been firstly introduced in the paper by Abaffy and Spedicato (1983) corresponding to the following parameter choices in the ABS class; $H_1 = I$, $v_i = e_i$ and

$$(17) \quad z_i = H_i a_i = s_i, \quad w_i = \frac{s_i}{s_i^T s_i},$$

implying that

$$(18) \quad \alpha_i = \frac{a_i^T x_i - b^T e_i}{s_i^T s_i},$$

and

$$(19) \quad H_{i+1} = H_i - \frac{s_i p_i^T}{s_i^T s_i}.$$

Among the properties of the modified Huang algorithm, we quote the following.

- (B1) The vector $w_i = s_i / s_i^T s_i$ is the vector of minimal Euclidean norm, which satisfies the projection condition (10). It is also the vector of minimal Euclidean norm, which solves the linear system, consisting of (10) and the equation $(A_{i-1})^T w_i = 0$, which defines optimally conditioned updates for H_i , in the sense defined by Spedicato (1987) and Deng and Spedicato (1988).

- (B2) In the Huang update the denominator $a_i^\top H_i a_i$ may become negative owing to the round-off errors. In the modified Huang update the denominator can never become negative.
- (B3) The termination condition for the ABS algorithm is equivalent to the matrix condition

$$(20) \quad V^\top AP = L,$$

where $V = (v_1, \dots, v_m)$, $P = (p_1, \dots, p_m)$ and L is nonsingular lower triangular. In presence of round-off the matrix L generally has nonzero elements above the diagonal. Yang (1988) shows, under certain assumptions, that, if the size of these elements is ε , when the Huang algorithm is used, then their size is ε^2 if the modified Huang algorithm is used.

- (B4) The eigenvalues of H_{i+1} are zero, with multiplicity i , and one, with multiplicity $n - i$. Let ε be the size of the interval where, owing to round-off, the perturbed zero eigenvalues of the matrix H_i are contained. Then, if the Huang update is used, this interval can be amplified by a factor σ or also σ^2 , where $\sigma = \|a_i\|/\|H_i a_i\|$ may be large for ill conditioned problems, depending on precisely which formula is used for the update of H_i . When the modified Huang update is used, either there is no amplification or the amplification factor is only σ , depending on precisely which formula is used for the update of H_i . See Broyden (1989) for an analysis of this phenomenon.

A number of alternative formulas for the computation of the search vectors are discussed in the monograph by Abaffy and Spedicato (1989). In our experiments we have considered, in addition to the formulas given above, the following formulations for the Huang and the modified Huang algorithms.

(A) Formulas using an i by i matrix G_i

These formulas are based upon the identity, here written for $H_1 = I$ and $v_i = e_i$

$$(21) \quad H_{i+1} = I - A_i G_i W_i^\top,$$

where $W_i = (w_1, \dots, w_i)$ and G_i is the LU decomposable matrix, defined by

$$(22) \quad G_i^{-1} = (W_i)^\top A_i.$$

In the case of the Huang algorithm the above formula gives

$$(23) \quad p_{i+1} = a_{i+1} - A_i G_i (A_i)^\top a_{i+1},$$

and G_i is updated by

$$(24) \quad G_{i+1} = \bar{G}_i + uu^T$$

where \bar{G}_i is G_i bordered with an extra null row and column, $u \in \mathbb{R}^{i+1}$ is defined by

$$(25) \quad u = \delta \begin{bmatrix} G_i A_i^T a_{i+1} \\ -1 \end{bmatrix}$$

and the scalar δ satisfies

$$(26) \quad \frac{1}{\delta} = a_{i+1}^T a_{i+1} - a_{i+1}^T A_i G_i (A_i)^T a_{i+1}.$$

In the case of the modified Huang algorithm the search vector is computed by

$$(27) \quad p_{i+1} = \bar{p}_{i+1} - A_i G_i (A_i)^T \bar{p}_{i+1},$$

where \bar{p}_{i+1} is p_{i+1} computed by (23).

(B) Formulas using order i vectors

These formulas have the following form, assuming that $H_1 = I$ and $v_i = e_i$,

$$(28) \quad p_i = z_i - \sum_{j < i} s_j r_j^T z_i,$$

where the $i - 1$ vectors $s_j, r_j \in \mathbb{R}^n$ have to be stored and are given by the relations $r_1 = a_1$, $s_1 = w_1$ and for $i > 1$

$$(29) \quad r_i = a_i - \sum_{j < i} r_j s_j^T a_i$$

$$(30) \quad s_i = w_i - \sum_{j < i} s_j r_j^T w_i.$$

In the case of the Huang algorithm we have identically $p_i = r_i$, $s_i = \frac{p_i}{a_i^T p_i}$, hence we obtain

$$(31) \quad p_i = a_i - \sum_{j < i} \frac{p_j^T a_i}{p_j^T a_j} p_j.$$

Note that formula (31) corresponds to one of the iterations of the standard Gram-Schmidt orthogonalization procedure.

Another form of this procedure is obtained noting that $p_i^T a_i = a_i^T H_i a_i = a_i^T H_i^2 a_i = p_i^T p_i$, from idempotency of H_i , hence

$$(32) \quad p_i = a_i - \sum_{j < i} \frac{p_j^T a_i}{p_j^T p_j} p_j.$$

In Huang (1975) it is argued that formula (31) is more stable than formula (32). This fact is to some extent confirmed by our numerical experiments.

When we consider the modified Huang algorithm, the obtained formulas are

$$(33) \quad p_i = r_i - \sum_{j < i} \frac{r_j^T r_i}{r_j^T r_j} p_j,$$

$$(34) \quad r_i = a_i - \sum_{j < i} \frac{p_j^T a_i}{r_j^T r_j} r_j.$$

Noting that in exact arithmetic p_i and r_i are identical, formulas (33) and (34) can be changed into the following:

$$(35) \quad p_i = r_i - \sum_{j < i} \frac{p_j^T r_i}{p_j^T p_j} p_j$$

$$(36) \quad r_i = a_i - \sum_{j < i} \frac{p_j^T a_i}{p_j^T p_j} p_j.$$

Formulas (35), (36) require the storage of only the vectors p_1, \dots, p_i . They are equivalent to twice iterating the standard Gram-Schmidt iteration (32), as suggested by Daniel et al. (1976). A further variation, in the spirit of the quoted Huang's argument, can be obtained by substituting $p_j^T p_j$ in (35) and (36) with $a_j^T p_j$.

(C) Formulas using $n - i$ vectors

If $H_1 = I$, $v_i = e_i$ and $w_i = \frac{z_i}{z_i^T H_i a_i}$, then the search vector can be defined by

$$(37) \quad p_i = u_i^i$$

where $u_j^1 = z_j$ for $j = 1, \dots, m$ and for $i > 1$ the vectors u_j^i are updated for $j = i + 1, \dots, m$ by

$$(38) \quad u_j^{i+1} = u_j^i - \frac{a_i^\top u_j^i}{a_i^\top u_i^i} u_i^i.$$

The Huang algorithm can be written using (37) and (38) by setting $u_j^1 = a_j$ for $j = 1, \dots, m$. The resulting algorithm is related to a class of algorithms for linear systems described by Sloboda (1978), see also Abaffy (1988).

(D) Formulas using i vectors

When $v_i = e_i$, the termination condition (20) is equivalent to the equation $a_j^\top p_i = 0$, $j < i$, that can be solved by the ABS method for the search vector p_i , approximated through a sequence p_i^1, \dots, p_i^i . If $H_1 = I$, $p_i^1 = z_i$ and we use for the first $i - 1$ equations of the i th system the search vectors used in solving the $(i - 1)$ th system, then the ABS iteration has the form

$$(39) \quad p_i^{j+1} = p_i^j - \frac{a_j^\top p_i^j}{a_j^\top p_j} p_j, \quad j = 1, \dots, i - 1$$

and

$$(40) \quad p_i = p_i^i.$$

Notice that relation (39) and (38) generate identical iterates, differing only in order of computation.

In the case of the Huang update $z_i = a_i$. Noting also that $p_i^j = H_j z_i = H_j a_i$, we have the identity $a_j^\top p_i^j = a_j^\top H_j a_i = a_j^\top H_j H_j z_i = p_j^\top p_i^j$; also we have $a_j^\top p_j = a_j^\top H_j a_j = a_j^\top H_j H_j p_j = p_j^\top p_j$. Hence (39) can be written in the form

$$(41) \quad p_i^{j+1} = p_i^j - \frac{p_j^\top p_i^j}{p_j^\top p_j} p_j, \quad j = 1, \dots, i - 1$$

with $p_i^1 = a_i$.

Formula (41) is identical with the stabilized Gram-Schmidt orthogonalization formula applied to the rows of A . A variation of this formula can be obtained by substituting $p_j^\top p_j$ with $a_j^\top p_j$.

3. THE EXPERIMENTED ALGORITHMS

We have implemented and tested eight versions of the Huang algorithm and nine versions of the modified Huang algorithm. We have also implemented four algorithms based upon the stabilized and the doubly iterated unstabilized Gram-Schmidt procedure applied either to the columns or to the rows of the matrix. Finally we have considered three commercial codes based upon the QR factorization via Householder or Givens rotations.

The eight versions of the Huang algorithm correspond in the Tables to the following symbols and formulas:

Method HS/ATS

This is the standard Huang algorithm, where $p_i = H_i a_i$, α_i is given by (12), H_i is updated by (13) written in the form

$$(42) \quad H_{i+1} = H_i - p_i u_i^T$$

where $u_i = p_i / a_i^T p_i$. All elements of H_i are explicitly computed by this formula, implying generally a loss of symmetry, since u_i may not be an exact multiple of p_i due to round-off errors.

Method HS/ATS/SYMM

This method differs from the previous one in the fact that symmetry of H_i is forced by applying the update formula only to the elements on and above the diagonal and setting the elements below the diagonal of index (i, j) equal to the elements of index (j, i) , $j < i$.

Method HS/STS

This is the Huang algorithm with α_i computed by (12) and H_i updated by (42) with $u_i = p_i / (p_i^T p_i)$. Again all elements of H_i are explicitly computed by this formula, implying generally a loss of symmetry.

Method HS/STS/SYMM

This method differs from the previous one by the fact that symmetry is forced as in HS/ATS/SYMM.

Method HBE

Here p_i is computed via relations (23) to (26), α_i by (12).

Method HB/ATS

Here p_i is computed by (31), α_i by (12).

Method HB/STS

Here p_i is computed by (32), α_i by (12).

Method HCS

Here p_i is computed via (37), (38), α_i by (12).

The nine versions of the modified Huang algorithm are:

Method MHS/STS

Here z_i , α_i , H_i are computed as in (17), (18), (42), with $u_i = p_i/s_i^T s_i$, and $p_i = H_i^T z_i$. Again all elements of H_i are explicitly computed by (42), implying generally a loss of symmetry.

Method MHS/STS/SYMM

This method differs from the previous one by the fact that symmetry is forced as in HS/ATS/SYMM.

Method MHS/ATP

This method differs from MHS/STS by the fact that $u_i = p_i/a_i^T p_i$.

Method MHST/PTP

In this method α_i is given by

$$(43) \quad \alpha_i = \frac{\tau_i}{p_i^T p_i}$$

and H_i is updated by (42), with $u_i = p_i/p_i^T p_i$.

Method MHST/STS

This method differs from MHST/PTP by the fact that α_i is given by (18).

Method MHBE

Here p_i is computed by (27), α_i by (43).

Method MHB/PTP

Here p_i is computed by (33) and (34), α_i by (43).

Method MHB/ATP

As the previous one, save that α_i is computed by (12).

Method MHB/STS

As the previous one, save that α_i is computed by (18).

Notice that all the Huang and modified Huang algorithms are started with x_1 the zero vector.

The four algorithms using the stabilized or doubly iterated Gram-Schmidt formula are the following:

GSM/COL

In this algorithm a QR factorization of A is obtained by applying the stabilized Gram-Schmidt algorithm to the columns of A (i.e. formula (41) with $p_i^1 = Ae_i$) and then solving the triangular system $Rx = Q^T b$ by back substitution.

GS2/COL

This algorithm is based upon twice iterating the standard Gram-Schmidt iteration (31) on the columns of A to form a QR factorization of A and then proceeding as in GSM/COL.

GSM/ROW

This algorithm computes p_i by the stabilized Gram-Schmidt iteration (41) applied to the rows of A , then updates x_i by (7) with α_i given by (12) (with $x_1 = 0$).

GS2/ROW

This method computes p_i by twice iterating the standard Gram-Schmidt iteration (31), then updates x_i by (7) with α_i given by (12) (with $x_1 = 0$).

Other considered algorithms are the following:

QR HOUSE

This method is the LINPACK implementation (codes SQRDC, SQRSL) of the QR method, where the QR factorization is obtained via Householder rotations.

QR GIVENS

This method is an implementation of the QR method, where the QR factorization is obtained via Givens rotations, available from the N.O.C. library of the Hatfield Polytechnic.

BRENT

This method is the code BRENTM developed by More and Cosnard (1979) for solving nonlinear systems. On linear systems this method has finite termination, being equivalent to the QR method via Householder rotations.

The above described codes have been written in FORTRAN and run on the computer IBM 4361 working in single precision and in double precision. The zero machine has the value $2^{-21} \cong 4 \cdot 10^{-7}$ in single precision; in double precision the value is $2^{-53} \cong 1 \cdot 10^{-16}$.

4. THE TEST PROBLEM

The algorithms have been tested on several families of ill conditioned small dimensional problems. The dimension varies between 4 and 20 and the L_1 condition number, estimated by the IBM ESSL routine DGEICD, varies between $4 \cdot 10^3$ and $8 \cdot 10^{13}$; it is given in Table 1.

The first four test problems are taken from Rutishauser (1968), where they are introduced in the context of determining the matrix with largest P -condition number whose elements are integers not greater than 10 in modulus.

They are called in the Tables as WILSON, WIL-RUT, WIL-MOR, TODD-SL and correspond respectively to the matrices

$$\begin{bmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{bmatrix} \quad \begin{bmatrix} 10 & 1 & 4 & 0 \\ 1 & 10 & 5 & -1 \\ 4 & 5 & 10 & 7 \\ 0 & -1 & 7 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix} \quad \begin{bmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{bmatrix}$$

Additional problems, indicated as WILSON**2, WIL-RUT**2, WIL-MOR**2, TODD-SL**2, WILSON**3, WIL-RUT**3, WIL-MOR**3 and TODD-SL**3 are obtained by taking the second and the third power of the above defined matrices.

Problem PASCAL corresponds to the matrix defined by the recursion formula

$$(43) \quad A_{i,j} = A_{i-1,j} + A_{i,j-1}, \quad i, j = 2, \dots, n$$

with $A_{1,j} = A_{j,1} = 1, j = 1, \dots, n$.

Problem HILBERT corresponds to the integer matrix obtained by multiplying the Hilbert matrix ($A_{i,j} = 1/(i+j-1)$) by the smallest integer such that the elements of the resulting matrix are all integers.

Problem VANDERMONDE uses the Vandermonde matrix ($A_{i,j} = y_i^{j-1}$), where the y_i are distinct small integers.

Problem MICCHELLI uses the (nonsingular) matrix defined by $A_{i,j} = |i - j|$.

Problem ZIELKE uses the matrix defined by formula (21) in Zielke (1986) in terms of Hadamard matrices.

All matrices in the defined problems have integer elements. Overall 33 matrices have been considered. Their L_1 estimated condition numbers are listed in Table 1.

For each matrix 100 problems have been defined by assigning a randomly generated integer solution and computing the corresponding right-hand side. In single precision the random elements of the solution have been generated in the interval $[-50, +50]$. In double precision they are generated in the interval $[-1000, +1000]$. Hence the double precision Tables do not relate to the same problems dealt with in single precision. The size of the involved integer numbers was checked to guarantee that the right hand side was computed by an exact operation among integers.

5. THE NUMERICAL RESULTS

The analysis of the numerical tests is based upon the following quantities all computed in double precision. The symbol $\|\cdot\|$ relates to the Euclidean norm.

E1MIN: the minimum value, with respect to all problems having a given matrix A , of the relative error μ_1 in the solution, defined by

$$(45) \quad \mu_1 = \frac{\|x' - x^+\|}{\|x^+\|}$$

where x' is the computed solution, x^+ the true solution;

E1MAX: the maximum value of μ_1 ;

E1MED: the geometric average value of μ_1 ;

E2MED: the geometric average value of the relative error μ_2 in the solution, defined by

$$(46) \quad \mu_2 = \frac{\|Ax' - b\|}{\|b\|};$$

E3MED: the geometric average value of

$$\mu_3 = \max_i \frac{|x'_i - x_i^+|}{\|x^+\|};$$

E4MED: the geometric average value of

$$\mu_4 = \max_i \frac{\left| \sum_{j=1}^n a_{ij} x'_j - b_i \right|}{\sum_{j=1}^n (|a_{ij}| \cdot |x'_j| + |b_i|)};$$

E5MED: the geometric average value of

$$\mu_5 = \frac{1}{2} \left(\frac{\mu_1}{\text{cond}(A)} + \mu_2 \right);$$

ANGMED: the geometric average value of μ_6 , the modulus of the cosine of the angle β_i between p_i and p_n

$$\mu_6 = \frac{|p_i^\top p_n|}{\|p_i\| \cdot \|p_n\|};$$

ANGMAX: maximum value, for $i = 1, \dots, n-1$, of μ_6 ;

D%: fraction of the number of iterations where the monotonicity condition $\|x_{i+1}\| > \|x_i\|$ is not satisfied;

PAIMED: the geometric average value, for $i = 1, \dots, n-1$, of μ_7 , the modulus of the cosine of the angle Φ_i between a_i and p_n

$$\mu_7 = \frac{|p_n^\top a_i|}{\|p_n\| \cdot \|a_i\|};$$

PAIMAX: the maximum value, for $j = 1, \dots, n-1$, of μ_7 ;

HELMED: the geometric average value of the modulus of all the elements of the matrix H_n ;

HELMAX: the maximum absolute value of the elements of H_n .

Due to space reasons, here we present in Tables 2 and 3 only part of the data upon which the conclusions of this paper are based. The interested reader can find the complete data in a report by Spedicato and Vespucci (1989).

Let us first consider the results pertaining to the Huang algorithm. In terms of accuracy, the worst results are those of algorithms HS/STS and HS/STS/SYMM, while in terms of loss of orthogonality algorithm HBE is the worst. The other algorithms behave similarly, with a marginal superiority of algorithm HCS, with respect to the average values, of algorithm HS/ATS/SYMM, with respect to the maximum values. Algorithm HS/ATS/SYMM gives marginally better results than HS/ATS. Algorithm HB/ATS is usually more accurate than algorithm HB/STS, up to two orders with respect to some parameters (but is worse with respect to some other parameters).

Let us now consider the results pertaining to the modified Huang algorithm. The worst results are given by algorithms MHBE, MHB/ATP, MHB/STS. In terms of the maximum error bad results are also obtained by MHB/ATP, MHB/STS and MHST/PTP. In term of keeping orthogonality among the search vectors, the best results are given by MHST/PTP (and MHST/STS, which generates identical search vectors), while in terms of the orthogonality of p_n with a_1, \dots, a_{n-1} , the best results are given by algorithms MHS/STS and MHS/ATP, particularly when considering the maximum values. In terms of accuracy in the solution the best algorithms are MHS/STS, MHS/ATP and MHST/STS. Of these MHST/STS is marginally less accurate in the approximation of the solution, measured by the first seven accuracy parameters, but gives less failures in satisfying the monotonicity condition $\|x_{i+1}\| > \|x_i\|$. Algorithm MHS/STS/SYMM intriguingly performs worse than MHS/STS. Overall the algorithms of the modified Huang type perform more accurately than the algorithms of the Huang type. This superiority is up to four or five orders in terms of angles, but generally of only two orders in terms of the other parameters.

Considering the algorithms based upon the Gram-Schmidt procedure, we notice that those which work on the columns give much less accurate solutions than those which work on the rows. This fact shows up particularly when considering the maximum errors. The algorithms based upon the doubly iterated standard Gram-Schmidt method are much better than the algorithms based upon the stabilized Gram-Schmidt method, a result which Hoffmann (1989) has also found to be true for ill conditioned problems with respect to a certain measure of orthogonality.

The three commercial codes based upon the QR factorization via Householder or Givens rotations behave similarly, the code BRENT being marginally better.

A comparison among the best algorithms in the considered groupings indicates that the best versions of the modified Huang algorithm, say MHS/STS or MHST/STS are just marginally less accurate than the doubly iterated Gram-Schmidt procedure applied on the rows (algorithm GS2/ROW) and usually better (marginally on the average error, sometimes significantly on the maximum error) than the commercial codes based upon the explicit QR factorization. In Spedicato and Vespucci (1989) tables are given containing detailed results of algorithms HS/ATS/SYMM, MHS/STS, GS2/ROW, BRENT on the considered test problems (runs in single precision). It is seen that on problems with largest condition numbers, algorithms MHS/STS and GS2/ROW perform similarly and significantly better than algorithm BRENT with respect to the errors E1MED and E1MAX.

6. FINAL REMARKS AND CONCLUSIONS

We have presented the results of extensive numerical experiments with algorithms for linear systems based upon methods of the LQ, QR and Huang type. We have obtained the following indications:

- different implementations of the Huang and the modified Huang algorithm behave differently, as already observed in Abaffy and Spedicato (1987);
- the modified Huang algorithms are more accurate than the Huang algorithms, as already shown in the quoted paper of Abaffy and Spedicato and as suggested by the theoretical analysis of Yang (1988) and Broyden (1989);
- the best modified Huang algorithms are essentially as good as the doubly iterated Gram-Schmidt algorithm applied on the rows of the coefficient matrix and coupled with the ABS update formula (14). They are generally more accurate than the stabilized Gram-Schmidt algorithm and the algorithms based upon the QR factorization, implemented in the considered commercial codes.

An important observation which arises from the experiments is that accurate results depend not only from the computation of the search vectors, but also from the computation of the stepsize. This shows up dramatically in the comparison of the algorithms MHST/PTP and MHST/STS. They generate the same set of search vectors but differ in the stepsize formula. Here the use of $p^T p$ in the denominator gives much worse results than the use of $z^T z$. A possible explanation is that while the used update formula for H_i is invariant to the norm of p_i , the update for x_i is not so invariant. Now the reprojection procedure is expected to reduce the norm of p_i , since it tends to annihilate components of p_i in the null space of H_i . A numerical check of the norm of p_i before and after reprojection has shown that the norm was never increased. It was usually only slightly decreased, but sometimes it could be dramatically decreased, especially in the last iterations. If this procedure produces a reprojected search vector of smaller norm than the correct one, a corresponding larger stepsize than the correct one would be obtained, with a possibly larger error in the computed solution. A further indication that such an overshooting actually occurs is given by the fact that the monotonicity condition $\|x_{i+1}\| > \|x_i\|$ is violated less often by algorithm MHST/PTP.

References

- [1] *Abaffy J.*: Equivalence of a generalization of Sloboda's algorithm with a subclass of the generalized ABS algorithm for linear systems, *Quaderno DMSIA 1/88* (1988), University of Bergamo.
- [2] *Abaffy J., Broyden C. G. and Spedicato E.*: A class of direct methods for linear equations, *Numerische Mathematik 45* (1984), 361–376.

- [3] *Abaffy J. and Galántai A.*: Conjugate direction methods for linear and nonlinear systems of algebraic equations, *Colloquia Mathematica Societatis János Bolyai* 50 (1986), 481–502.
- [4] *Abaffy J., Galántai A. and Spedicato E.*: The local convergence of ABS methods for nonlinear algebraic systems, *Numerische Mathematik* 51 (1987a), 429–439.
- [5] *Abaffy J., Galántai A. and Spedicato E.*: Application of ABS class to unconstrained function minimization, *Quaderno DMSIA 14/87* (1987b), University of Bergamo.
- [6] *Abaffy J. and Spedicato E.*: A generalization of the ABS algorithm for linear systems, *Quaderno DMSIA 4/85* (1985), University of Bergamo.
- [7] *Abaffy J. and Spedicato E.*: Numerical experiments with the symmetric algorithm in the ABS class for linear systems, *Optimization* 18(2) (1987), 197–212.
- [8] *Abaffy J. and Spedicato E.*: ABS projection algorithms: mathematical techniques for linear and nonlinear equations, Ellis Horwood, Chichester, 1989.
- [9] *Broyden C. G.*: On the numerical stability of Huang’s update, *Quaderno DMSIA 18/89* (1989), University of Bergamo.
- [10] *Daniel J., Gragg W. B., Kaufman L. and Stewart G. W.*: Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization, *Mathematics of Computation* 30 (1976), 772–795.
- [11] *Deng N. Y. and Spedicato E.*: Optimal conditioning parameter selection in the ABS class through a rank two update formulation, *Quaderno DMSIA 18/88* (1988), University of Bergamo.
- [12] *Hoffmann W.*: Iterative algorithms for Gram-Schmidt orthogonalization, *Computing* 41 (1989), 335–348.
- [13] *Huang H. Y.*: A direct method for the general solution of a system of linear equations, *Journal of Optimization Theory and Applications* 16 (1975), 429–445.
- [14] *More J. J. and Cosnard M. Y.*: Numerical solution of nonlinear equations, *ACM Trans.* 5 (1979), 64–85.
- [15] *Rutishauser H.*: On test matrices, *Progrès en Mathématiques Numériques* (M. Kuntzmann, eds.), Editions de la Faculté de Science de Besançon, 1968.
- [16] *Sloboda F.*: A parallel projection method for linear algebraic systems, *Apl. Mat. Českosl. Akad. Ved* 23 (1978), 185–198.
- [17] *Spedicato E.*: Optimal conditioning parameter selection in the ABS class for linear systems, Report 203, Mathematische Institute, University of Würzburg, 1987.
- [18] *Spedicato E. and Vespucci M. T.*: Variations on the Gram-Schmidt and the Huang algorithms for linear systems: a numerical study, *Quaderno DMSIA 21/89* (1989), University of Bergamo.
- [19] *Yang Z.*: On the numerical stability of the Huang and the modified Huang algorithms and related topics, Collection of reports on the ABS class of algorithms, 5, Department of Applied Mathematics, Dalian University of Technology, 1988.
- [20] *Zielke G.*: Report on test matrices for generalized inverses, *Computing* 36 (1986), 105–162.

TABLE 1: ESTIMATED L_1 CONDITION NUMBERS

NP	PROBLEM	N	COND
1	WILSON	4	$4 \cdot 10^3$
2	WIL-MOR	4	$4 \cdot 10^3$
3	TODD-SL	4	$4 \cdot 10^3$
4	PASCAL5	5	$1 \cdot 10^4$
5	MICCHELLI (A)	20	$2 \cdot 10^4$
6	HILBERT4	4	$2 \cdot 10^4$
7	MICCHELLI (B)	20	$3 \cdot 10^4$
8	ZIELKE	8	$5 \cdot 10^4$
9	WIL-RUTIS	4	$6 \cdot 10^4$
10	PASCAL6	6	$2 \cdot 10^5$
11	HILBERT5	5	$9 \cdot 10^5$
12	TANABE	20	$2 \cdot 10^6$
13	PASCAL7	7	$2 \cdot 10^6$
14	VANDERMONDE (A)	5	$3 \cdot 10^6$
15	WILSON**2	4	$1 \cdot 10^7$
16	WIL-MOR**2	4	$1 \cdot 10^7$
17	TODD-SL**2	4	$1 \cdot 10^7$
18	HILBERT6	6	$2 \cdot 10^7$
19	VANDERMONDE (B)	6	$3 \cdot 10^7$
20	PASCAL8	8	$4 \cdot 10^7$
21	VANDERMONDE (C)	7	$4 \cdot 10^7$
22	HILBERT4**2	4	$4 \cdot 10^8$
23	PASCAL9	9	$5 \cdot 10^8$
24	HILBERT7	7	$9 \cdot 10^8$
25	VANDERMONDE (D)	8	$1 \cdot 10^9$
26	WIL-RUTIS**2	4	$2 \cdot 10^9$
27	PASCAL10	10	$8 \cdot 10^9$
28	HILBERT8	8	$3 \cdot 10^{10}$
29	WILSON**3	4	$4 \cdot 10^{10}$
30	WIL-MOR**3	4	$4 \cdot 10^{10}$
31	TODD-SL**3	4	$4 \cdot 10^{10}$
32	PASCAL11	11	$1 \cdot 10^{11}$
33	WIL-RUTIS**3	4	$8 \cdot 10^{13}$

TABLE 2: AVERAGE ON PROBLEMS (SINGLE PRECISION)

METHOD	E1MIN	E1MED	E1MAX	E2MED	ANGMED
HS/ATS	1.E-02	2.E-01	6.E-01	5.E-05	1.E-03
HS/ATS/SYMM	1.E-02	2.E-01	5.E-01	4.E-05	7.E-04
HS/STS	2.E-01	3.E+02	1.E+03	6.E+00	1.E-03
HS/STS/SYMM	2.E-01	3.E+02	1.E+03	6.E+00	1.E-03
HBE	5.E-02	5.E-01	1.E+00	2.E-03	1.E-02
HB/ATS	1.E-02	3.E-01	7.E-01	1.E-04	1.E-03
HB/STS	3.E-02	2.E-01	5.E-01	8.E-03	2.E-03
HCS	7.E-03	2.E-01	7.E-01	2.E-05	1.E-03
MHS/STS	9.E-04	3.E-02	2.E-01	8.E-07	5.E-07
MHS/STS/SYMM	7.E-03	1.E-01	3.E-01	4.E-05	5.E-05
MHS/ATP	1.E-03	3.E-02	2.E-01	7.E-07	5.E-07
MHST/PTP	2.E-03	9.E-02	6.E-01	5.E-06	7.E-08
MHST/STS	1.E-03	4.E-02	2.E-01	1.E-06	7.E-08
MHBE	3.E-02	4.E-01	1.E+00	6.E-03	4.E-04
MHB/PTP	2.E-03	8.E-02	5.E-01	7.E-06	8.E-07
MHB/ATP	7.E-03	8.E+02	6.E+03	6.E-02	3.E-06
MHB/STS	6.E-03	6.E+01	4.E+02	4.E-03	3.E-06
GSM/COL	1.E+02	7.E+03	2.E+04	2.E-02	5.E-05
GS2/COL	7.E-03	3.E-01	1.E+00	4.E-06	8.E-08
GSM/ROW	2.E-02	3.E-01	6.E-01	3.E-04	8.E-05
GS2/ROW	9.E-04	3.E-02	1.E-01	1.E-06	8.E-08
QR HOUSE	2.E-03	8.E-02	4.E-01	2.E-06	
QR GIVENS	2.E-03	5.E-02	3.E-01	7.E-05	
BRENT	2.E-03	8.E-02	5.E-01	1.E-06	2.E-07

TABLE 3: MAXIMUM ON PROBLEMS (SINGLE PRECISION)

METHOD	E1MIN	E1MED	E1MAX	E2MED	ANGMED
HS/ATS	5.E-01	2.E+00	9.E+00	7.E-03	2.E-01
HS/ATS/SYMM	5.E-01	1.E+00	1.E+00	9.E-04	2.E-01
HS/STS	1.E+00	3.E+05	2.E+06	1.E+06	7.E-01
HS/STS/SYMM	1.E+00	3.E+05	2.E+06	1.E+06	7.E-01
HBE	6.E-01	1.E+00	3.E+00	7.E-02	5.E-01
HB/ATS	5.E-01	5.E+00	1.E+01	1.E-02	3.E-01
HB/STS	7.E-01	1.E+00	4.E+00	2.E+00	5.E-01
HCS	5.E-01	4.E+00	2.E+01	8.E-03	3.E-02
MHS/STS	1.E-01	1.E+00	5.E+00	3.E-06	1.E-04
MHS/STS/SYMM	4.E-01	7.E-01	2.E+00	5.E-02	2.E-02
MHS/ATP	1.E-01	8.E-01	5.E+00	3.E-06	1.E-04
MHST/PTP	3.E+00	7.E+02	4.E+03	6.E+00	2.E-06
MHST/STS	2.E-01	2.E+00	8.E+00	7.E-06	2.E-06
MHBE	5.E-01	2.E+01	2.E+02	5.E+00	1.E-02
MHB/PTP	7.E-01	7.E+01	7.E+02	3.E-02	1.E-04
MHB/ATP	4.E+01	4.E+20	4.E+22	1.E+15	1.E-03
MHB/STS	3.E+01	4.E+13	2.E+15	1.E+08	1.E-03
GSM/COL	1.E+06	4.E+07	2.E+08	4.E+00	3.E-03
GS2/COL	1.E+02	9.E+03	4.E+04	1.E-03	3.E-06
GSM/ROW	5.E-01	2.E+00	7.E+00	1.E-02	3.E-03
GS2/ROW	1.E-01	2.E+00	1.E+01	5.E-06	3.E-06
QR HOUSE	3.E+00	4.E+01	3.E+02	5.E-06	
QR GIVENS	7.E-01	2.E+01	8.E+01	2.E+01	
BRENT	7.E-01	2.E+01	1.E+02	1.E-05	7.E-07

Acknowledgements: Work supported by CNR grant 88.00337.01.

Authors' addresses: Emilio Spedicato, Maria Teresa Vespucci, Department of Mathematics, University of Bergamo, Via Salvecchio 19, 24100 Bergamo, Italy.