Boletín de Matemáticas
Nueva Serie, Volumen XIV No. 1 (2007), pp. 1–13

A NUMERICAL EXPERIMENT WITH HUANG ALGORITHM

HAMID ESMAEILI (*)

Abstract. It is believed that the Huang method is the best one to solve a system of linear equations in the class of ABS methods. Having presented 10 versions of the Huang method and compared them numerically, we will compare the best version of the Huang method with LU (along with partial pivoting) and QR (through Householder transformations) methods. Numerical results show that all three methods yield approximately similar output in well-conditioned problems while the Huang method works more effectively in ill-conditioned problems.

Key words and phrases. ABS Methods, Huang Algorithm, Ill-Conditioned Systems.

Resumen. Se cree que el método Huang es el mejor de los métodos ABS para resolver un sistema de ecuaciones lineales. Habiendo presentado 10 versiones del método Huang y evaluándolas numéricamente, compararemos la mejor versión del método Huang con los métodos LU (a través de pivotaje parcial) y QR (por medio de transformaciones de Householder). Las pruebas numéricas muestran que los tres métodos producen resultados similares en problemas bien condicionados, mientras que el método Huang trabaja más efectivamente en problemas mal condicionados.

Palabras claves. Métodos ABS, Algoritmo Huang , sistemas mal condicionados.

2000 Mathematics Subject Classification: 65F05, 65F20, 65F30.

(*) H. Esmaeili. Department of Mathematics, Bu-Ali Sina University, Hamedan, Iran. E-mail: esmaeili@basu.ac.ir.

1. Introduction

In many numerical methods for mathematical programming problems, it is often required to solve a system of linear equations which is usually consistent and, therefore, solvable. However, the exact nature of the system is not known *a priori*. The system may be determined and admits a unique solution or it may be underdetermined and admits infinite solutions. In either case, the system may contain some redundant equations. Therefore, it is highly desirable to develop a program which can be employed for all cases. Such a program has to perform the following functions: (a) determining whether the system is consistent or not, if the consistency of the equations is not assured beforehand; (b) determining the rank of the system and selecting a set of linearly independent equations that represents the original consistent system; and (c) determining the unique solution if the system is determined or determining a particular solution and some additional data, so that an expression can be formed for the general solution, if the system is underdetermined.

Consider the system of linear equations

$$a_i^T x = b_i, \qquad i = 1, 2, \ldots, m, \tag{1}$$

where $a_i \in \mathcal{R}^n$ and $m \leq n$. Let $A = (a_1, \ldots, a_m)^T \in \mathcal{R}^{m \times n}$ and $b = (b_1, \ldots, b_m)^T \in \mathcal{R}^m$. Thus, we can write the system (1) in the form of

$$Ax = b. \tag{2}$$

For the general solution of a consistent system such as that in (2), a convenient expression can be obtained by making use of the generalized inverse of the matrix A. The generalized inverse of A is the $n \times m$ matrix A^+ determined uniquely by [6]

$$AA^+A = A, \quad A^+AA^+ = A^+, \quad (AA^+)^T = AA^+, \quad (A^+A)^T = A^+A.$$

It is proved that $A^{+T} = A^{T+}$ and $rank(A^+) = rank(A)$. The system (2) is consistent if and only if

$$AA^+b = b. \tag{3}$$

In this case, the general solution of the system can be expressed by

$$x = x^* + Hy, \tag{4}$$

where y is an arbitrary vector, x^* is the minimum modulus solution, and H is an $n \times n$ symmetric matrix. The x^* and H are given by the following

$$x^* = A^+ b, \qquad H = I - A^+ A. \tag{5}$$

The matrix H is the symmetric projection matrix, which takes any vector y into the null space of the matrix A. Its rank is $n - r$, where $r = rank(A)$. When $r = n$, H reduces to zero and x^* becomes the unique solution of the system.

Therefore, a program fulfilling (a)-(c) can be established by computing the generalized inverse. There are several methods available towards this end [6]. However, it is worth noting that all of these methods depend critically on the correct choice of the numerical rank, that is, the rank of A as determined by a particular numerical method. Erroneous results are obtained when one uses a numerical rank greater than the theoretical rank of A or when one uses a numerical rank less than but too close to the theoretical rank of an ill-conditioned matrix A.

Another approach can be used to obtain the general solution (4). In this approach, we select first r linearly independent rows of the matrix A. Let this matrix be denoted by $\overline{A}$, and the corresponding component of the vector b by $\overline{b}$. Then, we consider the equation $\bar{A}x = \bar{b}$, where $\bar{A}$ is $r \times n$ and $\bar{b}$ is $r \times 1$. For this matrix $\bar{A}$, $\bar{A}\bar{A}^T$ is nonsingular and its inverse $(\bar{A}\bar{A}^T)^{-1}$ exists. Furthermore,

$$\bar{A}^+ = \bar{A}^T (\bar{A}\bar{A}^T)^{-1}. \tag{6}$$

Substituting (6) into (5), we obtain

$$x^* = \bar{A}^T (\bar{A}\bar{A}^T)^{-1} \bar{b}, \qquad H = I - \bar{A}^T (\bar{A}\bar{A}^T)^{-1} \bar{A}. \tag{7}$$

Once x^* is obtained, it is substituted into the remaining equations of the original system. If they are satisfied, the system is consistent and the general solution is given by (4) and (7). If any of them is violated, the system is inconsistent and has no solution. This approach is conceptually simple. However, as in the previous case, the determination of the rank is a subtle task. Besides, the matrix $\bar{A}\bar{A}^T$ may be ill-conditioned and its inverse becomes very inaccurate if the rows of the matrix $\bar{A}$ are nearly linearly dependent.

In this paper, a class of direct methods, called ABS class (standing for Abaffy, Broyden and Spedicato methods), is present which has all (a)-(c) properties concerning a general linear system of equations. Section 2 recalls the class of ABS methods and provides some of its properties. In Section 3, Huang algorithm [5,2] in ABS class and some of its other properties have been presented. This algorithm is an implicit LQ decomposition type method, and, in exact arithmetic, is equivalent to the Gram-Schmidt orthogonalization procedure. Furthermore, in this section, 10 versions of Huang algorithm are presented. In Section 4, the above versions are numerically compared and the best one (in our set of test problems) is selected. Then, a numerical experiment is given among the best version of Huang algorithm (version 7), QR method (using Householder transformations), and LU algorithm (with partial pivoting). Numerical results show that all 3 methods gives almost the same results, while for ill-conditioned systems, version 7 of Huang algorithm gives the best ones.

2. ABS Methods and Some of Its Properties

In this section, ABS method is introduced. ABS method is a class of methods of direct type for solving a system of m linear equations in n variables, full rank or deficient rank, determined or underdetermined; whenever the solution is not unique, by solving we intend that a particular solution is computed and a representation is given of the linear variety containing all the solutions. It has some selective parameters, so every particular choice results in a particular method. Indeed, it is shown [2] that all available methods of direct type for solving a linear system can be put in ABS class. In particular, ABS class implicitly includes LU, LL^T, QR decompositions, and conjugate directions methods.

Consider the linear system (2). ABS algorithm is a finite procedure based upon taking m steps along m search vectors, constructed using a certain deflection matrix. The process is made up of the following steps:

[1] Let x_1 be an arbitrary vector in $\mathcal{R}^n$ and H_1 an arbitrary n by n nonsingular matrix. Set $i = 1$ and $r_i = 0$.

[2] Compute the vector $s_i = H_i a_i$, and the scalar $t_i = a_i^T x_i - b_i$.

[3] *If* ($s_i = 0$ and $t_i = 0$), *then* let $x_{i+1} = x_i$, $H_{i+1} = H_i$, $r_{i+1} = r_i$ and *go to* step (7) (the ith equation is redundant). *If* ($s_i = 0$ and $t_i \neq 0$), *then* Stop (the ith equation and hence the system is incompatible).

[4] $\{s_i \neq 0\}$ Compute the search direction $p_i = H_i^T z_i$, where $z_i \in R^n$ is an arbitrary vector satisfying $z_i^T s_i \neq 0$. Compute

$$\alpha_i = \frac{t_i}{a_i^T p_i}$$

and take

$$x_{i+1} = x_i - \alpha_i p_i. \tag{8}$$

[5] {Updating H_i} Update H_i to H_{i+1} by

$$H_{i+1} = H_i - \frac{H_i a_i w_i^T H_i}{w_i^T H_i a_i}, \tag{9}$$

where $w_i \in R^n$ is an arbitrary vector satisfying $w_i^T s_i \neq 0$.

[6] Let $r_{i+1} = r_i + 1$.

[7] *If* $i = m$ *then* Stop (x_{m+1} is a solution) *else* let $i = i + 1$ and *go to* step (2).

Remark 2.1.

- If $s_i = 0$ and $t_i = 0$, then the ith equation is redundant.
- If $s_i = 0$ and $t_i \neq 0$, then the ith equation and hence the system is incompatible.
- In the above algorithm, r_{i+1} denotes the rank of $A_i = (a_1, \ldots, a_i)$.
- If the system (2) is compatible, then the general solution is given by

$$x = x_{m+1} + H_{m+1}^T q,$$

where $q \in R^n$ is arbitrary.

In what follows, we list certain properties of ABS methods [2,1]. For simplicity, it is assumed that $rank(A) = m$.

- The vectors $H_i a_j$, $i \leq j$, are linearly independent.
- For $i < j$, we have $H_i a_j = 0$. Therefore, $Range(A_{i-1}) = Null(H_i)$.
- $H_i a_i \neq 0$ if and only if a_i is linearly independent of $a_1, \ldots, a_{i-1}$.
- $rank(H_i) = n - i + 1$. If $H_i a_i \neq 0$, then $rank(H_{i+1}) = rank(H_i) - 1$.
- Every row of H_i corresponding to a nonzero component of w_i is linearly dependent on other rows.

- The matrix $W_i = (w_1, \ldots, w_i)$ has full column rank and $Null(H_{i+1}^T) = Range(W_i)$.
- The matrix $P_i = (p_1, \ldots, p_i)$ has full column rank. The matrix $L_i = A_i^T P_i$ is a nonsingular lower triangular matrix.
- For all i, $1 \le i \le m$, the vector x_{i+1} is a particular solution for the first i equations. Moreover,

$$x = x_{i+1} + H_{i+1}^T q, \qquad q \in \mathcal{R}^n,$$

  is the general solution for those equations.
- The updating formula H_i can be written as

$$H_{i+1} = H_1 - H_1 A_i (W_i^T H_1 A_i)^{-1} W_i^T H_1,$$

  where $W_i^T H_1 A_i$ is strongly nonsingular (the determinants of all of its main principal submatrices are nonzero).

Remark 2.2. Using the fifth property, Gu [3], Spedicato & Zhu [7] modified ABS algorithm such that the matrices H_i are rectangular with full row rank.

3. Huang Algorithm and Its Versions

An important algorithm in ABS class is called Huang algorithm, introduced by Huang [5] in a paper which led to the development of ABS class. The Huang algorithm is obtained by choosing

$$H_1 = I, \qquad z_i = a_i, \qquad w_i = a_i$$

for parameters of ABS algorithm, so that the updating formula (9) for H-matrices is read as

$$H_{i+1} = H_i - \frac{p_i p_i^T}{a_i^T p_i}\,. \tag{10}$$

We note that, in this case, the H-matrices are symmetric.

In addition to ABS properties, Huang algorithm has some other interesting properties listed below. For simplicity, it is assumed that $rank(A) = m$.

- Vectors p_i, $i = 1, \ldots, m$, are corresponding to vectors obtained by Gram-Schmidt orthogonalization process applied to the rows of A.

- If x_1 is a multiple of a_1, then x_{i+1}, $i = 1, \ldots, m$, is the least squares solution for the first i equations. Moreover,

  $$\|x_i\|_2 \leq \|x_{i+1}\|_2, \qquad \|x_{i+1} - x^+\|_2 \leq \|x_i - x^+\|_2,$$

  where $x^+ = x_{m+1}$ is the least squares solution of (2).
- For every i, $1 \leq i \leq m$, we have

  $$H_i^T = H_i, \qquad H_i^2 = H_i, \qquad Null(H_i) = Range(P_{i-1}).$$

- Since

$$a_i^T p_i = a_i^T H_i a_i = a_i^T H_i H_i a_i = p_i^T p_i, \tag{11}$$

  then quantities of $a_i^T p_i$ are positive real numbers.

To proceed, we consider several versions of the Huang algorithm, corresponding to various parameter choices and alternative formulations to compute p-vectors and update H-matrices. In the next section, we compare these versions numerically and determine the best one.

Version 1.

This is the standard Huang algorithm which can be stated as the following:

$$p_i = H_i a_i, \qquad H_{i+1} = H_i - p_i u_i^T, \qquad u_i = \frac{p_i}{a_i^T p_i}.$$

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.

Version 2.

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 by setting the elements below the diagonal of index (i, j) equal to the elements of index (j, i), $j < i$.

Version 3.

This method differs form the version 1 in the fact that, according to (11), we can replace $a_i^T p_i$ with $p_i^T p_i$:

$$p_i = H_i a_i, \qquad H_{i+1} = H_i - p_i u_i^T, \qquad u_i = \frac{p_i}{p_i^T p_i}\,.$$

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.

Version 4.

This method differs from the previous one by the fact that symmetry is forced as in version 2.

Note. According to (10), the matrix H_i can be written as follows:

$$H_i = I - \sum_{j=1}^{i-1} \frac{p_j p_j^T}{a_j^T p_j}\,. \tag{12}$$

Version 5.

In this method, we do not use H-matrices. Regarding (11) and (12), the search vector p_i can be obtained from

$$p_i = a_i - \sum_{j=1}^{i-1} \frac{p_j^T a_i}{p_j^T p_j}\, p_j. \tag{13}$$

We note that (13) is the same Gram-Schmidt orthogonalization method (without normalization) applied to the rows of A. Then, in exact arithmetic, Huang method is equivalent to Gram-Schmidt orthogonalization method. However, numerical experiments show that Huang method is stabler than Gram-Schmidt.

Version 6.

Here, according to (12), we can compute p_i vectors by

$$p_i = a_i - \sum_{j=1}^{i-1} \frac{p_j^T a_i}{a_j^T p_j}\, p_j\,. \tag{14}$$

Version 7.

Since

$$H_i p_i = H_i H_i p_i = H_i a_i = p_i,$$

then, by (12), we have

$$H_i = \left(I - \frac{p_{i-1} p_{i-1}^T}{p_{i-1}^T a_{i-1}} \right) H_{i-1}. \tag{15}$$

Therefore, by putting $p_j^i = H_i a_j$, we can obtain the following recurrence relation to compute p vectors:

$$p_i = p_i^i,$$

where

$$\begin{cases} p_j^1 = a_j & j = 1, \ldots, m \\ p_j^{i+1} = p_j^i - \dfrac{a_i^T p_j^i}{a_i^T p_i} p_i & j = i+1, \ldots, m, \; i = 1, \ldots, m-1. \end{cases}$$

Here, in the ith stage, we must store $m - i$ vectors. In particular, this version is adequate for pivoting.

Version 8.

In this version the vector p_i is obtained by $p_i = p_i^i$, where

$$\begin{cases} p_1^j = a_j & j = 1, \ldots, m \\ p_i^{j+1} = p_i^j - \dfrac{a_j^T p_i^j}{a_j^T p_j} p_j & j = 1, \ldots, i-1, \; i = 2, \ldots, m. \end{cases}$$

We note that $p_i^j = H_j a_i$.

Version 9.

In version 8, we have

$$a_j^T p_i^j = a_j^T H_j H_j a_i = p_j^T p_i^j, \qquad a_j^T p_j = a_j^T H_j a_j = a_j^T H_j H_j a_j = p_j^T p_j.$$

Hence, we can get the following version:

$$p_i = p_i^i,$$

where

$$\begin{cases} p_1^j = a_j & j = 1, \ldots, m \\ p_i^{j+1} = p_i^j - \dfrac{p_j^T p_i^j}{p_j^T p_j}\, p_j & j = 1, \ldots, i-1, \ i = 2, \ldots, m. \end{cases}$$

This version is, in fact, the stabilized Gram-Schmidt method applied to the rows of A.

Version 10.

Another version can be obtained by substituting $a_j^T p_j$ with $p_j^T p_j$ in the denominator of version 9, that is, in fact, a version of the stabilized Gram-Schmidt method by itself:

$$p_i = p_i^i,$$

where

$$\begin{cases} p_1^j = a_j & j = 1, \ldots, m \\ p_i^{j+1} = p_i^j - \dfrac{p_j^T p_i^j}{a_j^T p_j}\, p_j & j = 1, \ldots, i-1, \ i = 2, \ldots, m. \end{cases}$$

4. Numerical Experiments

In this section, the various versions of Huang method are numerically compared. After determining the best version, we can compare it numerically with LU decomposition (with partial pivoting) and QR decomposition (via the Householder transformations).

Versions 1 to 10 of the Huang algorithm are compared with systems which have the following matrix coefficients:

$$a_{ij} = \max\{i, j\} \tag{16}$$

$$a_{ij} = (i + j - 1)^{-1} \tag{17}$$

$$a_{ij} = |i - j| \tag{18}$$

$$a_{ij} = a_{i-1,j} + a_{i,j-1}, \quad a_{i1} = a_{j1} = 1. \tag{19}$$

Every system is of the order 10 or 17. For every system, we take 5 exact solutions, the components of which are a random number belonging to $[1, 1000]$.

Then, we compute the right hand side accordingly. So, we solve 40 consistent systems for every version. (Thus, 400 systems are solved.)

The criterion to compare versions is as follows: suppose that x and $\overline{x}$ are exact solution and computed solution, respectively. Also, assume that

$$(\mu_{10})_{ij} = \max\left\{\frac{|x_k - \overline{x}_k|}{|x_k|}, \quad k = 1, \ldots, 10\right\}, \quad 1 \le i \le 10, \quad 1 \le j \le 20,$$

is the quantity according to the jth system of order 10 which is solved by version i. If $(\mu_{10})_{ij} \le 5 \times 10^{-t}$, then all components of the computed solution by version i for the system j have at least t significant digits. Similarly, we define $(\mu_{17})_{ij}$ for $1 \le i \le 10$ and $1 \le j \le 20$. Take

$$\mu_i = \left(\sum_{j=1}^{20}(\mu_{10})_{ij} + \sum_{j=1}^{20}(\mu_{17})_{ij}\right), \quad i = 1, \ldots, 10.$$

Note that, if $\mu_{i_1} > \mu_{i_2}$, then the computed solution by version i_2 has components with significant digits, more than those of version i_1. Values of μ_is corresponding to any version (up to 7 digits) are as follows:

$$\begin{aligned}
\mu_1 &= 1.304702, & \mu_6 &= 1.342882,\\
\mu_2 &= 1.099594, & \mu_7 &= 1.096178,\\
\mu_3 &= 1.419204, & \mu_8 &= 1.096178,\\
\mu_4 &= 1.433135, & \mu_9 &= 1.359440,\\
\mu_5 &= 1.492797, & \mu_{10} &= 1.513501.
\end{aligned}$$

We observe that

$$\mu_7 = \mu_8 < \mu_2 < \mu_1 < \mu_6 < \mu_9 < \mu_3 < \mu_4 < \mu_5 < \mu_{10}.$$

Since version 6 has a better result than version 5, we can take version 6 as a modification of the standard Gram-Schmidt method. Moreover, it is expected that $\mu_7 = \mu_8$ since versions 7 and 8 are different only due to the order of computations. Therefore, the Huang method without explicitly using H-matrices (versions 7 and 8) gets some better results than those of stabilized Gram-Schmidt (version 9). The best version of the Huang method (regarding the correctness of the computed solution) is version 7. In what follows, by Huang method we mean version 7.

Now, we compare the Huang method with the LU method (with partial pivoting) and the QR method (with Householder transformations [4]). Since our main goal is merely to compare the accuracy of the computed solution using the above algorithms, then, for simplicity, only two exact solutions are used with following components:

$$x_k = 1 \quad \forall\, k, \tag{20}$$

$$x_k = k \quad \forall\, k. \tag{21}$$

The coefficient matrix, dimension of system, and criterion are the same as before. The computing platform was a PC with a PIV processor at 2.2 MHz and 512 Mb RAM. PASCAL language programming was used. Results have been shown in table 1, where its entries denote error in computed solution. The first 4 rows of table 1 are results according to systems with order 10 and the other 4 rows are the results for systems of order 17. Table 1 shows that all 3 methods gets almost similar results for well-conditioned systems (systems which matrix coefficients are (16) or (18)), while for ill-conditioned systems (which matrix coefficients are (17) or (19)) the Huang method get good results. In the general case, CPU time of the Huang method is greater than that of QR method which, in turn, is also greater than that of LU method.

Type of matrix	Exact solution of type (20)			Exact solution of type (21)		
	LU	QR	Huang	LU	QR	Huang
(16)	4.4E-11	4.1E-10	3.5E-10	1.6E-10	2.1E-10	1.0E-11
(17)	1.4E0	1.5E0	1.9E-4	2.3E0	1.9E-2	4.2E-4
(18)	2.9E-10	1.1E-10	4.4E-11	1.7E-10	0.6E-10	0.8E-10
(19)	1.1E-4	1.2E-4	1.0E-11	0.9E-4	1.3E-5	1.0E-11
(16)	4.7E-10	4.3E-10	0.7E-9	0.6E-9	1.5E-9	1.0E-11
(17)	2.0E+2	2.0E+2	1.3E0	1.0E+2	1.4E+2	0.6E0
(18)	1.0E-11	2.0E-10	3.0E-10	1.0E-11	0.7E-8	1.2E-9
(19)	4.4E+3	2.8E+3	1.0E-1	1.6E+4	0.7E+4	1.0E0

Table 1.

Conclusion

In this article, we presented ten versions of the Huang method in ABS class and compared them numerically. Then, we presented the numerical comparisons of the best version obtained for the Huang method with LU (along partial pivoting) and QR (using Householder transformations) methods. Numerical results show that all three methods arrive at approximately similar results when applied on well-conditioned problems while, for ill-conditioned problems, the Huang method works better. To achieve exact results, 400 systems of linear equations have been solved.

References

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

[2] J. Abaffy and E. Spedicato, *ABS Projection Algorithms: Mathematical Techniques for Linear and Nonlinear Equations*, Ellis Horwood, Chichester, 1989.

[3] S. Gu, *The simplified ABS algorithm*, Report 1/98, University of Bergamo, 1998.

[4] W. W. Hager, *Applied Numerical Linear Algebra*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[5] H. Y. Huang, *A direct method for the general solution of a system of linear equations*, JOTA 16, 1975, 429-445.

[6] G. H. Golub, C. F. Van Loan, *Matrix computations*, Jhon Hopkins University Press, 1989.

[7] E. Spedicato and M. Zhu, *Reformulation of ABS algorithm via full rank Abaffian*, Proceedings EAMA Conference, Seville, 396-403, 1997.

Recibido: Diciembre de 2006. Aceptado para publicación: Junio de 2007