Sindh Univ. Res. Jour. (Sci. Ser.) Vol. 48 (3) 639-642 (2016)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Numerical Solution of Symmetric Saddle Point Problems by Updating Procedure using QR Householder Factorization

S. ZEB++, M. YOUSAF

Department of Mathematics, University of Malakand, Chakdara, Dir (Lower), Khyber Pakhtunkhwa, Pakistan

Received 21st February 2015, and Revised 8th June 2016

Abstract: In this article, we proposed an updating QR Householder factorization procedure for the numerical solution of symmetric saddle point problems. We reduce the large structure of the problem into the small subproblem by removing block of rows and columns and calculating its QR factorization. The updating procedure is performed in the R factor of the QR factorization. We illustrate several numerical experiments and showed that the proposed procedure is applicable and forward stable for large scale and ill-conditioned type of saddle point problems.

Keywords: Saddle point problems; QR factorization; Householder reflection; Updating.

1. INTRODUCTION

We consider the following symmetric saddle point problem

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B^T} & \boldsymbol{0} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}, \quad \textbf{or} \quad \boldsymbol{Sz = f}, \qquad (1)$$

where $A \in R^{m\times m}$ is symmetric positive definite matrix, $B \in R^{m\times n}$ has full column rank such that $rank(B) = n \leq m$, B^T represents transpose of the matrix B, $S \in R^{(m+n)\times(m+n)}$, $z = \begin{pmatrix} x \\ y \end{pmatrix} \in R^{m+n}$, and $f = \begin{pmatrix} b \\ c \end{pmatrix} \in R^{m+n}$. Then there exists a unique solution of equation (1). The system (1) arises as a first-order optimality conditions for the equality constrained quadratic programming problem

$$\min J(x) = \frac{1}{2}x^T Ax - b^T x \text{ Subject to } B^T x = c.$$

The name saddle point problem is due to the fact that any solution (x_*, y_*) of (1) is a saddle point for the Lagrangian

$$\boldsymbol{L(x,y) = \frac{1}{2}x^T Ax - b^T x + (B^T x - c)^T y},$$

where y represents the vector of Lagrangian multipliers and it satisfies

$$\boldsymbol{L(x_*, y) \leq L(x_*, y_*) \leq L(x, y_*)},$$

for any $x \in R^m$ and $y \in R^n$.

Such problems occur in many scientific and technical applications such as in constrained and weighted least squares problems (Golub and Loan, 1996) (Björck, 1996) computational fluid dynamics (CFD) (Glowinski, 2008) (Wesseling, 2009), in constrained optimization (Gill, *et al.,* 1981) (Gill, *et al.,* 1992) etc. For detail survey of saddle point problems, we refer the readers to (Benzi, *et al.,* 2005). Due to its wide scale of applications, a large amount of work has been devoted quite recently for obtaining the numerical solution of saddle point problems using direct and indirect methods studied in (Smoktunowicz and Okulicka-Dłużewska, 2013) (Najafi and Edalatpanah, 2015) (Liang and Zhang 2016) etc.

The authors in (Smoktunowicz and Okulicka-Dłużewska, 2013) studied some direct methods for the numerical solution of the symmetric saddle point problems. These methods were based on orthogonal decomposition of matrices such as Householder QR decomposition and singular value decomposition (SVD). Numerous aspects of updating matrix factorization for different kinds of problems are studied in (Murray, *et al.,* 1974) (Kaufman, *et al.,* 1976) (Reichel and Gragg, 1990) (Hammarling and Lucas, 2008) (Yousaf, 2010) (Andrew and Dingle, 2014). Here, we exploit the large structure of the problem (1) and reduce it to a small subproblem using partition process by removing block of rows and columns, calculating its QR factorization and then updating its R factor by appending back the removed block of columns and rows respectively and obtained its solution. We used the QR Householder factorization in our updating procedure due to its stability property (Wilkinson, 1965) (Golub and Loan, 1996) (Björck, 1996). Our procedure is mainly based on the updating techniques used in (Yousaf, 2010) for the solution of least squares problems. It provides a useful alternative in case of unknown iterations in iterative methods. Moreover, the proposed technique is suitable for large scale and ill-conditioned type of problems (1).

++Corresponding Author: Email: zebsalman@gmail.com, myousafuom@gmail.com

We organized the paper as follows. The basic concepts are studied in section 2 and the proposed QR Householder updating procedure for the solution of (1) is given in section 3. We provided numerical experiments in section 4 and illustrated the results. Conclusion is given in section 5.

2. MATERIALS AND METHODS

In this section, we discuss basic concepts which will be used in our main results. For details, we refer to (Golub and Loan, 1996) (Björck, 1996).

2.1 QR FACTORIZATION

Let

$$F = QR,\ Q \in R^{m\times m}, R \in R^{m\times n} \quad (2)$$

be the QR factorization of the matrix $F \in R^{m\times n}$. This factorization can be found using various methods given in (Golub and Loan, 1996) (Björck, 1996). Here, we discuss the Householder method. For a non-zero vector $v \in R^n$, we define the Householder matrix $H \in R^{n\times n}$ such that

$$H = I_n - \tau v v^T, \text{ where } \tau = \frac{2}{v^T v}. \quad (3)$$

The matrix H is symmetric and orthogonal. The Householder vector can be obtained using several ways. We define it in the simplest form as

$$v = x \pm \|x\|_2 e_1,$$

such that

$$Hx = x - \tau v v^T x = \mp \alpha e_1, \quad (4)$$

where $\alpha = \|x\|_2$ and e_1 is a unit vector $(1,0,\ldots,0)$. We will use the following form of the Householder vector v based on the parlett's formula (Parlett, 1971) as

$$v_1 = x_1 - \|x\|_2 = \frac{{x_1}^2 - {\|x\|_2}^2}{x_1 + \|x\|_2} \quad (5)$$

where $x_1 > 0$ be the first element of a vector $x \in R^n$. We state the following algorithm due to (Golub and Loan, 1996) for computation of Householder vector v and Householder matrix H used in QR Householder factorization Algorithm.

Algorithm 2.2: Computation of parameters τ, α, and Householder vector v.

Function $[v, \tau] = householder(x)$

$s = {\|x\|_2}^2$

$v = x;\ v(1) = 1$

if $s = 0$

$\tau = 0$

else

$u = \sqrt{{x_1}^2 + s}$

% choose sign of v

if $x(1) \le 0$

$v(1) = x_1 - u$

else

$v(1) = \dfrac{-s}{x_1 + u}$

end

$\tau = \dfrac{2v(1)^2}{s + {x_1}^2}$

$v = \dfrac{v}{v(1)}$

$\alpha = u$

end

Definition 2.3: (Smoktunowicz and Okulicka-Dluzewska, 2013) An algorithm for computing $z = \binom{x}{y}$ for the problem (1) will be forward stable if the computed solution $z_p = \binom{x_p}{y_p}$ in floating point arithmetic satisfies

$$\frac{\|z_p - z\|}{\|z\|} \le L_1 \varepsilon_M \kappa(S) + O({\varepsilon^2}_M), \quad (6)$$

and forward stable for computing x and y if it satisfies

$$\frac{\|x_p - x\|}{\|x\|} \le L_2 \varepsilon_M \kappa_x(A, B; z) + O({\varepsilon^2}_M), \quad (7)$$

and

$$\frac{\|y_p - y\|}{\|y\|} \le L_3 \varepsilon_M \kappa_y(A, B; z) + O({\varepsilon^2}_M), \quad (8)$$

respectively, where L_1, L_2, and L_3 are modestly growing functions of m and n, ε_M is the machine rounding off unit and κ represents the condition number of a matrix.

3. RESULTS AND DISCUSSION

We consider the problem (1) and re-write the matrix S and its corresponding right hand side (RHS) f as

$$S = \begin{pmatrix} S(1{:}s-1, 1{:}m+n) \\ S(s{:}s+r, 1{:}m+n) \\ S(s+r+1{:}m+n, 1{:}m+n) \end{pmatrix}, f = \begin{pmatrix} f(1{:}s-1) \\ f(s{:}s+r) \\ f(s+r+1{:}m+n) \end{pmatrix}. \quad (9)$$

We remove the block of rows $S_r = S(s{:}s+r, 1{:}n)$ and $f_r = f(s{:}s+r)$ from s^{th} position of S and f in (9) using partition process and obtain the following:

$$S_1 = \begin{pmatrix} S(1{:}s-1, 1{:}m+n) \\ S(s+r+1{:}m+n, 1{:}m+n) \end{pmatrix}, f_1 = \begin{pmatrix} f(1{:}s-1) \\ f(s+r+1{:}m+n) \end{pmatrix}, \quad (10)$$

where $S_1 \in R^{p_1\times q_1}$, $f_1 \in R^{p_1}$, $p_1 = m+n-r$, and $q_1 = m+n$. In a similar fashion, we remove a block of columns $S_c = S(:, s{:}s+c)$ from the s^{th} position of the matrix S_1 in (10) which provided the following incomplete problem from (1)

$$S_2 z_2 = f_2,\ S_2 \in R^{p_2\times q_2}, f_2 \in R^{p_2}, z_2 \in R^{q_2}, \quad (11)$$

where $S_2 = [\,S_1(:, 1{:}s-1), S_1(:, s+1{:}q_2)]$ and $q_2 = q_1 - c$, $p_2 = p_1$, $f_2 = f_1$.

The QR factorization of the reduced problem (11) is calculated with the help of Algorithm 2.2. Applying Householder matrices $H_{q_2} \cdots H_1$, we obtain

$$R_2 = H_{q_2} \cdots H_1 S_2,\ g_2 = H_{q_2} \cdots H_1 f_2, \quad (12)$$

and $T_c = H_{q_2} \cdots H_1 S_c$, where H represents the Householder matrix, g_2 the corresponding RHS of R_2 and T_c is the updated block of columns. Now, we append the removed block of columns T_c to the triangular factor R_2 at the s^{th} position as follow:

$$R_{2c} = \big(R_2(:, 1{:}s-1)\ T_c(:, s{:}s+c)\ R_2(:, s+c+1{:}m+n)\big). \quad (13)$$

If the factor R_2 is in upper triangular or trapezoidal form, then we consider $R_2 = R_1$. Otherwise, we form the Householder matrices $H_{q_2+c} \cdots H_s$ using Algorithm 2.2 and update the R_2 factor to obtain the following

$$\boldsymbol{R_1 = H_{q_2+c} \cdots H_s R_{2c}}, \textbf{ and } \boldsymbol{g_1 = H_{q_2+c} \cdots H_s g_2}. \quad \textbf{(14)}$$

Similarly, by appending the removed block of rows to the R_1 factor of (14) and to its corresponding RHS, we get

$$\boldsymbol{R_{1r} = \begin{pmatrix} R_1(1{:}s-1,:) \\ S_r(1{:}r,:) \\ R_1(s+1{:}p_1,:) \end{pmatrix}, \quad g_{1r} = \begin{pmatrix} g_1(1{:}s-1) \\ f_r(1{:}r) \\ g_1(s+1{:}p_1) \end{pmatrix}}. \quad \textbf{(15)}$$

We apply the permutation matrix P on R_{1r} in (15) to shift the appended block of rows to the bottom as

$$\boldsymbol{PR_{1r} = \begin{pmatrix} R_1(1{:}p_1,:) \\ S_r(1{:}r) \end{pmatrix}}, \textbf{ and } \boldsymbol{Pg_{1r} = \begin{pmatrix} g_1(1{:}p_1) \\ f_r(1{:}r) \end{pmatrix}}.$$

Furthermore, by applying Householder matrices $H_n \cdots H_1$ using Algorithm 2.2 on above equation, we obtain

$$\boldsymbol{R = H_{m+n} \cdots H_1(PR_{1r}), g = H_{m+n} \cdots H_1(Pg_{1r})}, \quad \textbf{(16)}$$

where R is the upper triangular factor and g is its corresponding RHS. Applying back substitution using MATLAB built-in command *backsub* or by using backslash operator \ , we get the solution of the problem (1) as

$$\boldsymbol{z = \binom{x}{y} \in R^{m+n}}, \textbf{ where } \boldsymbol{x \in R^m}, \textbf{ and } \boldsymbol{y \in R^n}.$$

The procedure described in this section for solution of problem (1) is given in algorithmic form as follow.

Algorithm 3.1: Algorithm for Solution of Saddle Point Problems

Input: $\boldsymbol{A \in R^{m\times m},\ B \in R^{m\times n},\ b \in R^m, c \in R^n}$

Output: $\boldsymbol{z = \binom{x}{y} \in R^{m+n},\ x \in R^m, y \in R^n}$.

1: Partition Process: $[\boldsymbol{S_r, f_r, S_1, f_1, S_c, S_2, f_2}] = \boldsymbol{Partition(S, f)}$

2: Calculate QR factorization of S_2 using Algorithm 2.2: $\boldsymbol{R_2 = H_{q_2} \cdots H_1 S_2}$, $\boldsymbol{g_2 = H_{q_2} \cdots H_1 f_2}$ and

$\boldsymbol{T_c = H_{q_2} \cdots H_1 S_c}$

3: Appending block of columns and updating the factor R_2 using Algorithm 2.2 if required:

$\boldsymbol{R_2(1{:}p_2, q_2+1{:}q_2+c) = T_c(1{:}p_2, 1{:}c)}$,
If $\boldsymbol{p_2 \le q_2}$, **then**
$\boldsymbol{R_1 = triu(R_2),\ g_1 = g_2}$
else
for $\boldsymbol{j = 1\ to \min(p_2, q_2+c)}$ **do**
$\boldsymbol{[v, \tau, R_2(j,j)] = householder(R_2(j,j), R_2(j+1{:}p_2, j))}$
$\boldsymbol{W = R_2(j, j+1{:}q_2+c) + v^T R_2(j+1{:}p_2, j+1{:}q_2+c)}$
$\boldsymbol{R_1(j, j+1{:}q_2+c) = R_2(j, j+1{:}q_2+c) - \tau W}$
if $\boldsymbol{j < q_2+c}$ **then**
$\boldsymbol{R_1(j+1{:}p_2, j+1{:}q_2+c) = R_2(j+1{:}p_2, j+1{:}q_2+c) - \tau v W}$
end if
$\boldsymbol{g_1(j{:}p_2) = g_2(j{:}p_2) - \tau \binom{1}{v}(1\ v^T) g_2(j{:}p_2)}$
end for
$\boldsymbol{R_1 = triu(R_2)}$
end if

4: Appending block of rows S_r to R_1 and f_r to g_1 and updating the factors using Algorithm 2.2:

for $\boldsymbol{j = 1\ to \min\ (p_1, q_1)}$ **do**
$\boldsymbol{[v, \tau, R_1(j,j)] = householder(R_1(j,j), S_r(1{:}r, j))}$
$\boldsymbol{W = R_1(j, j+1{:}q_1) + v^T S_r(1{:}r, j+1{:}q_1)}$
$\boldsymbol{R(j, j+1{:}q_1) = R_1(j, j+1{:}q_1) - \tau W}$
if $\boldsymbol{j < q_1}$ **then**
$\boldsymbol{S_r(1{:}r, j+1{:}q_1) = S_r(1{:}r, j+1{:}q_1) - \tau v W}$
end if
$\boldsymbol{g_j = g(j)}$
$\boldsymbol{g(j) = (1-\tau) g(j) - \tau v^T f_r(1{:}r)}$
$\boldsymbol{f_r(1{:}r) = f_r(1{:}r) - \tau v g_j - \tau v v^T f_r(1{:}r)}$
end for
if $\boldsymbol{p_1 < q_1}$ **then**
$\boldsymbol{[Q_r, R_r] = qr(S_r(:, p_1+1{:}q_1))}$
$\boldsymbol{R_r(p_1+1{:}p_1+r, p_1+1{:}q_1) = R}$
$\boldsymbol{f_r = Q^T{}_r f_r}$
end if
$\boldsymbol{R = \binom{R_r}{0},\ g = \binom{g_1}{f_r}}$

5: $\boldsymbol{= backsub(R, g)}$, $\boldsymbol{x = z(1{:}m,:)}$, $\boldsymbol{y = z(m+1{:}m+n,:)}$.

4. NUMERICAL EXPERIMENTS

In this section, we consider different test problems for illustration of our results. The results obtained are also compared with QR Householder Algorithm studied in (Smoktunowicz and Okulicka-Dłużewska, 2013) which is the generalization of the Golub's method for the augmented system formulation (ASF) (Björck and Paige, 1994) (Björck, 1996) and uses the Householder QR decomposition given in (Golub and Van Loan, 1996). We performed our numerical experiments in MATLAB with machine precision $\varepsilon_m = 2.22e-16$. In the following examples, let z_p, x_p and y_p be the computed solutions corresponding to true values z, x and y where $x = randn(m,1)$, $y = randn(n,1)$, and z_{QR}, x_{QR} and y_{QR} be the solutions obtained through QR Householder method, and $b = A*x + B*y$, $c = B'*x$. Moreover, the condition number of any matrix F is represented by $\kappa(F) = \|F^{-1}\|\|F\|$ where $\|\cdot\|$ denotes the 2-norm.

Example 4.1: We consider $A = H = hilb(m)$ of size $m \times m$ which is a Hilbert matrix (symmetric positive definite and ill-conditioned) with elements of the form $h_{ij} = \frac{1}{i+j-1}$, $i, j = 1, \cdots, m$, and $B = randn(m,n)$ which is generated with MATLAB command $\boldsymbol{randn('state', 0)}$ where $\boldsymbol{randn}$ is used to generate random numbers with uniformly normal distribution and $(\boldsymbol{'state', 0})$ is used to reset the random number generator to its initial state.

From **(Table 1)**, we observe that the matrix A in problem (i), (ii), and (iii) is very ill-conditioned due to its large condition number in comparison with matrix S. Moreover, the proposed Algorithm 3.1 gives forward stable solution for saddle point problem (1).

Table 1: Description and Results for Example 4.1 with partition to subproblem matrix S_2 of size 3 × 3

Problem	$Size(A)$	$\kappa(A)$	$Size(B)$	$\kappa(S)$	$\frac{\|z_p - z\|}{\|z\|}$	$\frac{\|z_{QR} - z\|}{\|z\|}$	$\frac{\|x_p - x\|}{\|x\|}$	$\frac{\|x_{QR} - x\|}{\|x\|}$	$\frac{\|y_p - y\|}{\|y\|}$	$\frac{\|y_{QR} - y\|}{\|y\|}$
i	10×10	1.60e+13	10×5	6.57e+05	4.92e-11	3.88e-11	6.55e-11	5.08e-11	2.44e-15	1.27e-15
ii	15×15	4.43e+17	15×10	1.83e+07	8.47e-10	8.85e-11	1.40e-09	1.49e-10	1.95e-14	2.13e-15
iii	20×20	2.09e+18	20×15	2.94e+05	9.41e-12	1.66e-12	1.13e-11	2.00e-12	2.84e-15	2.52e-15

Example 4.2: Consider a symmetric positive matrix $A = \frac{(A_1 + A'_1)}{2}$ where $A_1 = PDP$, P is $m \times m$ random orthogonal and D is a diagonal matrix. We obtained the results which are stated in the following table.

Table 2: Results for Example 4.2 with partition to subproblem matrices S_2 of size 5×5, 10×5 and 50×50 respectively

Problem	$Size(A)$	$\kappa(A)$	$Size(B)$	$\kappa(S)$	$\frac{\|z_p - z\|}{\|z\|}$	$\frac{\|z_{QR} - z\|}{\|z\|}$	$\frac{\|x_p - x\|}{\|x\|}$	$\frac{\|x_{QR} - x\|}{\|x\|}$	$\frac{\|y_p - y\|}{\|y\|}$	$\frac{\|y_{QR} - y\|}{\|y\|}$
iv	500×500	1.00e+07	500×300	3.95e+08	1.69e-08	8.06e-09	2.15e-08	1.02e-08	3.60e-14	2.06e-14
v	1000×1000	1.00e+05	1000×500	5.46e+06	4.97e-10	3.07e-10	6.04e-10	3.74e-10	1.09e-14	8.52e-15
vi	2000×2000	1.00e+06	2000×1000	7.64e+07	7.68e-09	6.12e-09	9.40e-09	7.55e-09	2.55e-14	1.84e-14

From (**Table 2**), we conclude that in problem (iv), (v), and (vi) the matrix S is ill-conditioned in comparison with A and the obtained results are forward stable.

5. CONCLUSION

We consider symmetric saddle point problem and obtained its solution using updating QR Householder factorization procedure. The proposed algorithm is applicable for large scale and ill-conditioned type matrices. In future work, the problem can further be studied for sparse problems using Givens QR factorization.

REFERENCES:

Andrew, R., N. Dingle, (2014). Implementing QR factorization updating Algorithms on GPUs. Parallel Computing. 40(7): 161-172.

Bojorok A., C. C. Paige, (1994). Solution of augmented linear systems using orthogonal factorizations. BIT. 34:1–24.

Benzi, M., G. H. Golub, J. Liesen, (2005). Numerical solution of saddle point problems. Acta Numerica. 14(1): 1-37.

Gill, P. E., W. Murray, D. B. Poncelon, M. A. Saunders, (1992). Preconditioners for indefinite systems arising in optimization. SIAM Journal of Matrix Analysis and Applications. 13(1): 292-311.

Gill, P. E., W. Murray, M. H. Wright, (1981). Practical Optimization, Academic Press Inc. London.

Glowinski, R., (2008). Lectures on Numerical Methods for Non-Linear Variational Problems, Springer Science and Business Media, Springer-Verlag Berlin Heidelberg.

Golub, G. H., C. F. Van Loan, (1996). Matrix Computation, 3rd edition, Johns Hopkins University Press, Baltimore, MD, USA.

Hammarling, S., C. Lucas, (2008). Updating the QR factorization and the least squares problems. MIMS EPrint. University of Manchester. UK. 111.

Kaufman, L., J. Daniel, W. B. Gragg, G. W. Stewart, (1976). Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. Mathematics of Computation. 30:772-795.

Liang, Z-Zheng, and G-Feng. Zhang, (2016) Augmented block splitting preconditioner for singular saddle point problems. Applied Mathematics Letters 56:34-41.

Murray, W., P. E. Gill, G. H. Golub, M. Saunders, (1974). Methods for Modifying matrix factorizations. SIAM Mathematics of Computation. 28:505-535.

Najafi, H. S., S. A. Edalatpanah, (2015) On the modified symmetric successive over-relaxation method for augmented systems. Comp. Appl. Math. 34:607-617.

Parlett, B. N., (1971). Analysis of algorithms for reflections in bisectors. SIAM Review. 13:197-208.

Reichel, L., W. B. Gragg, (1990). Fortran subroutines for updating the QR decomposition. ACM Trans. Math. Software. 16:369-377.

Smoktunowicz, A., F. Okulicka-Dłużewska, (2013). Numerical solutions of symmetric saddle point problem by direct methods. Demonstratio Mathematica. 46(3):437-451.

Wesseling, P., (2009). Principles of computational fluid dynamics, Vol. 29, Springer Science and Business Media, Springer-Verlag Berlin Heidelberg.

Wilkinson, J. H., (1965) Error analysis of transformations based on the use of matrices of the form $I - 2ww^T$. Error in Digital Computation. Wiley New York. 2:77-101.

Yousaf, M., (2010). Repeated Updating as a Solution Tool for Linear Least Squares Problems, PhD thesis, Department of Mathematical Sciences, University of Essex, UK.