Electronic Transactions on Numerical Analysis.
Volume 33, pp. 189-206, 2009.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

ERROR ANALYSIS AND COMPUTATIONAL ASPECTS OF SR FACTORIZATION VIA OPTIMAL SYMPLECTIC HOUSEHOLDER TRANSFORMATIONS*

A. SALAM† AND E. AL-AIDAROUS‡

Dedicated to Gérard Meurant on the occasion of his 60th birthday

Abstract. Symplectic QR like methods for solving some structured eigenvalue problems involves SR factorization as a key step. The optimal symplectic Householder SR factorization (SROSH algorithm) is a suitable choice for performing such a factorization. In this paper, we carry out a detailed error analysis of the SROSH algorithm. In particular, backward and forward error results are derived. Also, the computational aspects of the algorithm (such as storage, complexity, implementation, factored form, block representation) are described. Some numerical experiments are presented.

Key words. Skew-symmetric inner product, optimal symplectic Householder transformations, SR factorization, error analysis, backward and forward errors, implementation, factored form, WY factorization, complexity.

AMS subject classifications. 65F15, 65F50

1. Introduction. Let A be a $2n$-by-$2n$ real matrix. The SR factorization consists in writing A as the product SR, where S is symplectic and R is J-upper triangular, i.e.,

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$$

is such that R_{11}, R_{12}, R_{22} are upper triangular and R_{21} is strictly upper triangular [3, 4]. This decomposition is a key step in constructing symplectic QR-like methods in order to solve the eigenvalue problem of a class of structured matrices; see [3, 9, 14]. It can be viewed as the equivalent of the classical QR factorization, when instead of an Euclidean space, one considers a linear space equipped with a specified skew-symmetric inner product; see [8, 12] and the references therein. The space is then called symplectic, and its orthogonal group with respect to this inner product is called the symplectic group. In contrast with the Euclidean case, the symplectic group is not compact. A numerical determination of the canonical form for symplectic matrices has been derived by Godunov et al. [5]. The set of real $2n$-by-$2n$ matrices for which a SR factorization exists is dense in $\mathbb{R}^{2n\times 2n}$ [3].

Computing the QR factorization is currently handled by the Gram-Schmidt orthogonalization process [1, 2] or via Householder transformations [6, 10, 15]. In the symplectic case, the SR factorization can be performed via symplectic Gram-Schmidt (SGS) algorithms (see [12] and the references therein) or by using the SRDECO algorithm [3]. Recently, a new algorithm (SROSH) for computing the SR factorization was proposed in [13]. It is a method based on optimal symplectic Householder transformations. We present here a detailed error analysis of the method. In particular, backward and forward error results are derived. Questions about how close is the computed symplectic factor to being symplectic and how large is the error in the SR factorization are answered. We also describe its most important computational aspects: storage, complexity, implementation, factored form, block representation.

*Received January 31, 2008. Accepted May 26, 2009. Published online on December 11, 2009. Recommended by Lothar Reichel.

†Université Lille Nord de France, ULCO, Laboratoire de Mathématiques Pures et Appliqués, 50 rue F. Buisson, B.P. 699, 62228 Calais Cedex, France. (Ahmed.Salam@lmpa.univ-littoral.fr).

‡King Abdul Aziz University (KAU), Department of Mathematics, Girls Section, P.O. Box 80203, Jeddah 21589, Kingdom of Saudi Arabia. (ealaidarous@kau.edu.sa). This author was partially supported by KACST, Saudi Arabia.

The remainder of this paper is organized as follows. In Section 2, the symplectic Householder SR factorization is reviewed. Section 3 is devoted to the computational aspects of the optimal symplectic Householder SR factorization. Section 4 treats in detail the error analysis of the SROSH algorithm. Some illustrative numerical experiments are presented.

2. SR decomposition. We review briefly the SR factorization, via symplectic Householder transformations. A detailed study is provided in [13].

2.1. Symplectic Householder transformations (SHT). Let J_{2n} (or simply J) be the $2n$-by-$2n$ real matrix

$$J_{2n} = \begin{bmatrix} 0_n & I_n \\ -I_n & 0_n \end{bmatrix}, \tag{2.1}$$

where 0_n and I_n stand for the n-by-n null and identity matrices, resectively. The linear space $\mathbb{R}^{2n}$ with the indefinite skew-symmetric inner product

$$(x, y)_J = x^T J y \tag{2.2}$$

is called symplectic. We denote the orthogonality with respect to $(\cdot,\cdot)_J$ by $\perp'$, i.e., for $x, y \in \mathbb{R}^{2n}$, $x \perp' y$ stands for $(x, y)_J = 0$. The symplectic adjoint x^J of a vector x is defined by

$$x^J = x^T J. \tag{2.3}$$

The symplectic adjoint of $M \in \mathbb{R}^{2n\times 2k}$ is defined by

$$M^J = J_{2k}^T M^T J_{2n}. \tag{2.4}$$

DEFINITION 2.1. *A matrix $S \in \mathbb{R}^{2n\times 2k}$ is called symplectic if*

$$S^J S = I_{2k}. \tag{2.5}$$

The symplectic group (multiplicative group of square symplectic matrices) is denoted by $\mathbb{S}$. A transformation T given by

$$T = I + cvv^J \text{ where } c \in \mathbb{R},\quad v \in \mathbb{R}^{\nu} \quad (\text{with } \nu \text{ even}) \tag{2.6}$$

is called a symplectic Householder transformation [13]. It satisfies

$$T^J = I - cvv^J. \tag{2.7}$$

The vector v is called the direction of T.

2.2. The mapping problem. For $x, y \in \mathbb{R}^n$, there exists a symplectic Householder transformation T such that $Tx = y$ if $x = y$ or $x^J y \neq 0$. T is given by

$$T = I - \frac{1}{x^J y}(y - x)(y - x)^J.$$

Moreover, each non-null vector x can be mapped onto any non-null vector y by a product of at most two symplectic Householder transformations. Symplectic Householder transformations are rotations, i.e., $\det(T) = 1$; and the symplectic group $\mathbb{S}$ is generated by symplectic Householder transformations.

2.3. SR factorization via symplectic Householder transformations. The symplectic Householder SR factorization can be viewed as the analogue of Householder QR factorization in the symplectic case; see [13] for more details. Let $\{e_1, \ldots, e_{2n}\}$ be the canonical basis of $\mathbb{R}^{2n\times 2}$, $[a, b] \in \mathbb{R}^{2n\times 2}$, and ρ, μ, and ν be arbitrary scalars. We seek symplectic Householder transformations T_1 and T_2 such that

$$T_1(a) = \rho e_1, \qquad T_2(e_1) = e_1, \qquad T_2(T_1(b)) = \mu e_1 + \nu e_{n+1}. \tag{2.8}$$

The fact that T_2T_1 is a symplectic isometry yields the necessary condition

$$a^J b = (T_2T_1(a))^J (T_2T_1(b)) = \rho\nu. \tag{2.9}$$

THEOREM 2.2. *Let ρ, ν be such that (2.9) is satisfied, and let*

$$c_1 = -\frac{1}{\rho a^J e_1}, \qquad v_1 = \rho e_1 - a,$$

$$c_2 = -\frac{1}{(T_1(b))^J(\mu e_1 + \nu e_{n+1})}, \qquad v_2 = \mu e_1 + \nu e_{n+1} - T_1(b).$$

Then

$$T_1 = I + c_1 v_1 v_1^J, \qquad T_2 = I + c_2 v_2 v_2^J, \tag{2.10}$$

satisfy (2.8).

2.4. The SRSH algorithm. We outline here the steps of the algorithm. Let $A = [a_1, \ldots, a_p, a_{p+1}, \ldots, a_{2p}] \in \mathbb{R}^{2n\times 2p}$. Let r_{11}, $r_{1,p+1}$, $r_{p+1,p+1}$ be arbitrary scalars satisfying $r_{11}r_{p+1,p+1} = a_1^J a_{p+1}$. The first step of the SRSH algorithm is to find T_1 (i.e., c_1 and v_1 by Theorem 2.2) such that $T_1(a_1) = r_{11}e_1$ and T_2 (i.e., c_2 and v_2 by Theorem 2.2) such that $T_2(e_1) = e_1$ and $T_2T_1(a_{p+1}) = r_{1,p+1}e_1 + r_{p+1,p+1}e_{n+1}$. This step involves two free parameters. The action of T_2T_1 on A is

$$T_2T_1A = \begin{bmatrix} r_{11} & r(1, 2:p) & r_{1,p+1} & r(1, p+2:2p) \\ 0 & A_{11}^{(2)} & 0 & A_{12}^{(2)} \\ 0 & r(p+1, 2:p) & r_{p+1,p+1} & r(p+1, p+2:2p) \\ 0 & A_{21}^{(2)} & 0 & A_{22}^{(2)} \end{bmatrix}.$$

Denote

$$\tilde{A}^{(2)} = \begin{bmatrix} A_{11}^{(2)} & A_{12}^{(2)} \\ A_{21}^{(2)} & A_{22}^{(2)} \end{bmatrix},$$

and let r_{22}, $r_{2,p+2}$, $r_{p+2,p+2}$ be arbitrary scalars with $\tilde{A}^{(2)}(1,:)^J \tilde{A}^{(2)}(p,:) = r_{22}\ r_{p+2,p+2}$. The next step is to apply the previous step to $\tilde{A}^{(2)}$, i.e., find (by Theorem 2.2)

$$\tilde{T}_3 = I_{2n-2} + c_3\tilde{v}_3\tilde{v}_3^J, \text{ where } \tilde{v}_3 = \begin{bmatrix} u_3 \\ \hline w_3 \end{bmatrix} \in \mathbb{R}^{2n-2}, \text{ and}$$

$$\tilde{T}_4 = I_{2n-2} + c_4\tilde{v}_4\tilde{v}_4^J, \text{ where } \tilde{v}_4 = \begin{bmatrix} u_4 \\ \hline w_4 \end{bmatrix} \in \mathbb{R}^{2n-2},$$

such that

$$\tilde{T}_4\tilde{T}_3\tilde{A}^{(2)} = \begin{bmatrix} r_{22} & r(2, 3:p) & r_{2,p+2} & r(2, p+3:2p) \\ 0 & A_{11}^{(3)} & 0 & A_{12}^{(3)} \\ 0 & r(p+2, 3:p) & r_{p+2,p+2} & r(p+2, p+3:2p) \\ 0 & A_{21}^{(3)} & 0 & A_{22}^{(3)} \end{bmatrix}.$$

Set $T_3 = I_{2n} + c_3 v_3 {v_3}^J$, $T_4 = I_{2n} + c_4 v_4 {v_4}^J$, where

$$\begin{bmatrix} v_3 & v_4 \end{bmatrix} = \left[\begin{array}{cc} 0 & 0 \\ u_3 & u_4 \\ \hline 0 & 0 \\ w_3 & w_4 \end{array}\right] \in \mathbb{R}^{2n\times 2}.$$

Then T_3, T_4 are symplectic Householder transformations [13], and their action is given by

$$T_4T_3T_2T_1A = \begin{bmatrix} r_{11} & r_{12} & r(1,3:p) & r_{1,p+1} & r_{1,p+2} & r(1,p+3:2p) \\ 0 & r_{22} & r(2,3:p) & 0 & r_{2,p+2} & r(2,p+3:2p) \\ 0 & 0 & A_{11}^{(3)} & 0 & 0 & A_{12}^{(3)} \\ 0 & r_{p+1,2} & r(p+1,3:p) & r_{p+1,p+1} & r_{p+1,p+2} & r(p+1,p+3:2p) \\ 0 & 0 & r(p+2,3:p) & 0 & r_{p+2,p+2} & r(p+2,p+3:2p) \\ 0 & 0 & A_{21}^{(3)} & 0 & 0 & A_{22}^{(3)} \end{bmatrix}.$$

The jth step is now clear. At the last step (pth step), we obtain

$$T_{2p}T_{2p-1}\dots T_4T_3T_2T_1A = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = R \in \mathbb{R}^{2n\times 2p},$$

where R_{11}, R_{12}, R_{22} are upper triangular and R_{21} is strictly upper triangular. R is called J-upper triangular. We get $A = SR$ with $S = T_1^J T_2^J \dots T_{2p-1}T_{2p}^J$.

The algorithm SRSH involves free parameters. At each iteration j of SRSH, two of the three parameters r_{jj}, $r_{j,p+j}$, $r_{j+p,j+p}$ can be chosen freely.

3. SR factorization via optimal symplectic Householder transformations. The SROSH algorithm corresponds to an efficient way of choosing the free parameters in the SRSH algorithm. It is based on the following result; see [13].

THEOREM 3.1. *Let* $[a,b] \in \mathbb{R}^{2n\times 2}$*, and let*

$$\rho = sign(a_1)\|a\|_2,\ c_1 = -\frac{1}{\rho a^J e_1},\ v_1 = a - \rho e_1,\ T_1 = I + c_1 v_1 v_1^J.$$

Then T_1 *has minimal 2-norm condition number and satisfies*

$$T_1(a) = \rho e_1. \tag{3.1}$$

Let u *be the vector* $u = T_1(b)$ *and* u_j *its* j*th component, and let*

$$\begin{aligned} &\nu = u_{n+1}, && \xi = \|u - u_1e_1 - u_{n+1}e_{n+1}\|_2, && \mu = u_1 \pm \xi, \\ &c_2 = -\frac{1}{\pm\xi u_{n+1}}, && v_2 = u - \mu e_1 - u_{n+1}e_{n+1}, && T_2 = I + c_2v_2v_2^J. \end{aligned}$$

Then T_2 *has minimal 2-norm condition number and satisfies*

$$T_2(e_1) = e_1 \text{ and } T_2(u) = \mu e_1 + \nu e_{n+1}. \tag{3.2}$$

We refer to T_1, T_2 as optimal symplectic Householder transformations.

REMARK 3.2. *The vector* v_2 *differs from* u *only by the first and* $(n+1)$*th components and satisfies* $v_2(n+1) = 0$*. This will be taken in consideration when storing* v_2*.*

3.1. The SROSH algorithm. Given an input vector a, the procedure $osh1$ below returns the coefficient c_1 and the vector v_1 for the optimal symplectic Householder transformation T_1 of Theorem 3.1. Similarly, the procedure $osh2$ computes the optimal symplectic Householder transformation T_2.

We normalize the optimal symplectic Householder vector of the procedure $osh1$ so that $v_1(1) = 1$. Thus $v_1(2:2n)$ can be stored where the zeros have been introduced in a, i.e., $a(2:2n)$. As in the classical case, we refer to $v_1(2:2n)$ as the essential part of the optimal symplectic Householder vector of T_1. This gives the following procedure:

ALGORITHM 3.3. *(First optimal symplectic Householder transformation)*
Given $a \in \mathbb{R}^{2n}$, this function computes $v \in \mathbb{R}^{2n}$ with $v(1) = 1$ and $c \in \mathbb{R}$ such that $T = I + cvv^J$ is the optimal symplectic Householder transformation satisfying $Ta = \rho e_1$ of Theorem 3.1.

function $[c, v]$ = *osh1(a)*
  $den = length(a);\;\; n = den/2;$
  $J = [zeros(n), eye(n); -eye(n), zeros(n)];$
  $\rho = sign(a_1)\,\|a\|_2\,;\;\; aux = a(1) - \rho;$
  if $aux = 0$
    $c = 0;\;\; v = 0;\;\; \%T = I;$
  else if $a_{n+1} = 0$
    display('division by zero');
    return
  else
    $v = \dfrac{a}{aux};\; c = \dfrac{aux^2}{\rho a_{n+1}};\;\; v(1) = 1;$
    $\%\, T = (eye(den) + c * v * v' * J);$
  end

In a similar way, it is possible to normalize the optimal symplectic Householder vector v_2 so that $v_2(1) = 1$. Moreover, since the $v_2(n+1)$ is zero. $v_2(2:2n)$ can be stored where the zeros have been introduced in u. This is clarified by the following figure. Such storage is not possible if the symplectic Householder transformation used is not optimal (since in this case, $v_2(n+1)$ is not necessarily zero).

$$\begin{array}{ccccc} u & & T_2 u & & v_2 \\ \left[\begin{array}{c} u_1 \\ u_2 \\ \vdots \\ u_n \\ \hline u_{n+1} \\ u_{n+2} \\ \vdots \\ u_{2n} \end{array}\right] & \xrightarrow{T_2} & \left[\begin{array}{c} \times \\ 0 \\ \vdots \\ 0 \\ \hline \times \\ 0 \\ \vdots \\ 0 \end{array}\right] & \longleftarrow & \left[\begin{array}{c} 1 \\ v_2(2) \\ \vdots \\ v_2(n) \\ \hline 0 \\ v_2(n+2) \\ \vdots \\ v_2(2n) \end{array}\right] \end{array}$$

We refer to $v_2(2:2n)$ as the essential part of the optimal symplectic Householder vector of T_2.

ALGORITHM 3.4. *(Second optimal symplectic Householder transformation)*
Given $u \in \mathbb{R}^{2n}$, this function computes $v \in \mathbb{R}^{2n}$ with $v(1) = 1$ and $c \in \mathbb{R}$ such that $T = I + cvv^J$ is the optimal symplectic Householder transformation satisfying $Te_1 = e_1$ and $Tu = \mu e_1 + \nu e_{n+1}$ of Theorem 3.1.

function $[c, v]$ = *osh2*(u)
    $den = length(u); \;\; n = den/2;$
    $J = [zeros(n), eye(n); -eye(n), zeros(n)];$
    if $\; n = 1$
        $v = zeros(den, 1); \;\; c = 0; \;\; \%T = I$
    else
        $I = [2 : n, n + 2 : den]; \;\; \xi = norm(u(I));$
        if $\; \xi = 0$
            $v = zeros(den, 1); \;\; c = 0; \;\; \%T = I;$
        else
            $\nu = u(n + 1); \;\; \%\mu = u_1 + \xi;$ *no need to compute* μ;
            if $\; u(n + 1) = 0$
                display('division by zero')
                return
            else
                $v = -\dfrac{u}{\xi}; \;\; v(1) = 1; \;\; v(n + 1) = 0; \;\; c = \dfrac{\xi}{u(n + 1)};$
            end
        end
    end

REMARK 3.5. *The quantity* ξ *also can be taken to be* $-norm(u(I))$ *in Algorithm 3.4.*

Note that the product of a symplectic Householder matrix $T = I + cvv^J$ with a given vector a of dimension $2n$ can be computed easily without explicitly forming T itself, since

$$Ta = a + cv(v^J a).$$

This formulation requires $8n$ flops. If $A \in \mathbb{R}^{2n\times 2p}$ is a matrix, then the TA can be written as $TA = A + cv(v^J A) = A + cv(v^T JA) = A + cvw^T$, where $w = -A^T Jv$. Likewise, if $T = I + cvv^T J \in \mathbb{R}^{2p\times 2p}$, then

$$AT = A(I + cvv^T J) = A + cwv^T J,$$

where $w = cAv$. Hence, a $2n$-by-$2p$ symplectic Householder transformation is a rank-one matrix update and involves a matrix-vector multiplication and an outer product update. It requires $16np$ flops. If T is treated as a general matrix and the structure is not exploited, the amount of work increases by an order of magnitude. Thus, as in the classical case, symplectic Householder updates never entail the explicit formation of the symplectic Householder matrix. Note that both of the above Householder updates can be implemented in a way that exploits the fact that $v_1(1) = 1$ for T_1 and $v_2(1) = 1$, $v_2(n + 1) = 0$ for T_2.

3.2. Factored form representation. Many symplectic Householder-based factorization algorithms compute products of symplectic Householder matrices

$$S = T_1 T_{n+1} T_2 T_{n+2} \dots T_k T_{n+k}, \quad T_j = I + c_j v^{(j)} {v^{(j)}}^T J, \tag{3.3}$$

where $k \le n$ and the vectors $v^{(j)}$, $v^{(n+j)}$, $j = 1, \dots, k$ have the form

$$v^{(j)} = \begin{bmatrix} 0_{j-1}^T & 1 & v_{j+1}^{(j)} & \dots & v_n^{(j)} & | & 0_{j-1}^T & v_{n+j}^{(j)} & \dots & v_{2n}^{(j)} \end{bmatrix}^T,$$
$$v^{(n+j)} = \begin{bmatrix} 0_{j-1}^T & 1 & v_{j+1}^{(n+j)} & \dots & v_n^{(n+j)} & | & 0_j^T & v_{n+j+1}^{(n+j)} & \dots & v_{2n}^{(n+j)} \end{bmatrix}^T.$$

It is not necessary to compute S explicitly, even if S is required in subsequent calculations. Thus, if a product $S^J B$ is needed, where $B \in \mathbb{R}^{2n\times 2k}$, then the following loop can be executed:

ALGORITHM 3.6.
for $j = 1 : k$
$B = T_j B;$
$B = T_{n+j} B;$
end.

The storage of the symplectic Householder vectors $v^{(1)}, \ldots, v^{(k)},\ v^{(n+1)}, \ldots, v^{(n+k)}$ and the corresponding c_j amounts to a factored representation of S. For $b \in \mathbb{R}^{2n}$, we examine carefully the implementation of the product $T_j^J b$. Set $H_j = c_j v^{(j)} {v^{(j)}}^T$; then $T_j^J = I - H_j J_{2n}$. Due to the structure of v_j, the submatrix $H_j([1 : j-1, n+1 : n+j-1], :)$, obtained from H_j by deleting all rows except rows $1, \ldots, j-1$ and $n+1, \ldots, n+j-1$, is null. This implies that components $1, \ldots, j-1$ and $n+1, \ldots, n+j-1$ of $H_j J_{2n} b$ vanish. Hence, the corresponding components of $T_j^J b$ remain unchanged. Similarly, the submatrix $H_j(:, [1 : j-1, n+j : 2n])$ (obtained from H_j by deleting all columns except columns $1, \ldots, j-1$ and $n+j, \ldots, 2n$) is null. Setting $rows = [j : n, n+j : 2n]$, one gets $[H_j J_{2n} b](rows) = [H_j](rows, rows)[J_{2n} b](rows)$. It follows from the particular structure of J_{2n} that $[J_{2n} b](rows) = J_{2(n-j+1)} b(rows)$. Thus, the product $T_j^J b$ is reduced to compute $b(rows) - [H_j](rows, rows) J_{2(n-j+1)} b(rows)$.

Suppose now that the essential part $v^{(j)}([j+1 : n, n+j : 2n])$ of the vector $v^{(j)}$ is stored in $A([j+1 : n, n+j : 2n], j)$, and similarly that $v^{(j+n)}([j+1 : n, n+j+2 : 2n])$ is stored in$A([j+1 : n, n+j+2 : 2n], n+j)$. The overwriting of B with $S^J B$ can then be implemented as follows.

ALGORITHM 3.7. *Product in factored form*
for $j = 1 : k$*;*

$$rows = [j : n,\ j+n : 2n]; \qquad v(rows) = \begin{bmatrix} 1 \\ A(j+1 : n, j) \\ A(n+j : 2n, j) \end{bmatrix};$$

$$B(rows, :) = B(rows, :) - c_j v(rows) v(rows)^T J_{2(n-j+1)} B(rows, :);$$

$$v(rows) = \begin{bmatrix} 1 \\ A(j+1 : n, j+n) \\ 0 \\ A(n+j+1 : 2n, j+n) \end{bmatrix};$$

$$B(rows, :) = B(rows, :) - c_j v(rows) v(rows)^T J_{2(n-j+1)} B(rows, :);$$

end

Algorithm 3.7 illustrates the economies of the factored form representation. It requires only $16mk(2n-k)$ flops. If S is explicitly represented as a $2n$-by-$2n$ matrix, $S^J B$ would involves $16n^2 m$ flops.

REMARK 3.8. *Note that in lines 3 and 5 of Algorithm 3.7, it should be* $J_{2n}(rows, rows)$ *instead of* $J_{2(n-j+1)}$*. Due to the particular structure of* J*, we have* $J_{2n}(rows, rows) = J_{2(n-j+1)}$*. The interest of this fact is that the product* $J_{2(n-j+1)} B(rows, :)$ *does not induce additional flops cost (the situation would be different if instead of* J*, one has a general matrix).*

When it is necessary to compute S explicitly, one can use a forward accumulation algorithm:

ALGORITHM 3.9.
$S = I_{2n};$
for $j = 1 : k$

$\quad\quad S = ST_j;$
$\quad\quad S = ST_{n+j};$
$\quad$ *end*

Another option is to use a backward accumulation algorithm:

ALGORITHM 3.10.
$S = I_{2n}$
for $j = k : -1 : 1$
$\quad S = T_j S;$
$\quad S = T_{n+j} S;$
end

Furthermore, it is important to note that backward accumulation requires fewer flops than forward accumulation. This is due to the particular structure of T_j. In fact, the block $(j-1)$-by-$2n$ (resp. $(n+j)$-by-$2n$) of T_j (resp. of T_{n+j}) is equal to the block $(j-1)$-by-$2n$ (resp. $(n+j)$-by-$2n$) of the identity. Hence, S is "mostly the identity" at the beginning of backward accumulation and becomes gradually full as the iteration progresses. In contrast, S is full in forward accumulation immediately after the first iteration. For this reason, backward accumulation is cheaper. It can be implemented with $32(n^2k - nk^2 + k^3/3)$ flops as follows.

ALGORITHM 3.11. *Backward accumulation*
$S = I_{2n};$
for $j = k : -1 : 1$
$\quad rows = [j : n \;\; j+n : 2n];$

$$v(rows) = \begin{bmatrix} 1 \\ A(j+1:n, j+n) \\ 0 \\ A(n+j+1:2n, j+n) \end{bmatrix};$$

$$S(rows, rows) = S(rows, rows) - c_j v(rows) v(rows)^T J_{2(n-j+1)} S(rows, rows);$$

$$v(rows) = \begin{bmatrix} 1 \\ A(j+1:n, j) \\ A(n+j:2n, j) \end{bmatrix};$$

$$S(rows, rows) = S(rows, rows) - c_j v(rows) v(rows)^T J_{2(n-j+1)} S(rows, rows);$$

end

3.3. A block representation. Suppose that $S = T_1 T_{n+1} T_2 T_{n+2} \dots T_k T_{n+k}$ is a product of $2n$−by−$2n$ optimal symplectic Householder matrices as in (3.3). Since each T_j is a rank-one modification of the identity, it follows from the structure of the symplectic Householder vectors that S is a rank-$2k$ modification of the identity and can be written in the form

$$S = I + WY^T J, \tag{3.4}$$

where W and Y are $2n$-by-$2k$ matrices. We refer to this block representation as a $J - WY$ representation. Its computation is based on the following lemma.

LEMMA 3.12. *Suppose that $S = I + WY^T J$ is an $2n$-by-$2n$ symplectic matrix with $W, Y \in \mathbb{R}^{2n\times j}$. If $T = I + cvv^T J$ with $v \in \mathbb{R}^{2n}$ and $z = cSv$, then*

$$ST = I + W_+ Y_+^T J,$$

where $W_+ = [W \;\; z]$ and $Y_+ = [Y \;\; v]$ are $2n$-by-$(j+1)$ matrices.

Proof.

$$\begin{aligned} ST &= (I + WY^T J)(I + cvv^T J) = I + WY^T J + cSvv^T J \\ &= I + WY^T J + zv^T J = I + [W \;\; z][Y \;\; v]^T J. \end{aligned} \qquad \square$$

The block representation of S can be generated from the factored form by repeatedly applying the Lemma 3.12 as follows.

ALGORITHM 3.13. *Block representation*
Suppose that $S = T_1T_{n+1}T_2T_{n+2}\dots T_kT_{n+k}$ *is a product of* $2n\text{−by−}2n$ *optimal symplectic Householder matrices as in (3.3). This algorithm computes matrices* $W,\ Y \in \mathbb{R}^{2n\times 2k}$ *such that* $S = I + WY^TJ$.

$Y = v^{(1)};\ \ W = c_1v^{(1)};$
$z = c_{n+1}(I + WY^TJ)v^{(n+1)};$
$W = [W\ z];\ \ Y = [Y\ v^{[n+1]}];$
for $j = 2 : k$
  $z = c_j(I + WY^TJ)v^{(j)};$
  $W = [W\ z];\ \ Y = [Y\ v^{(j)}];$
  $z = c_{n+j}(I + WY^TJ)v^{(n+j)};$
  $W = [W\ z];\ \ Y = [Y\ v^{(n+j)}];$
end.

This algorithm requires about $16(k^2n - k^3/3)$ flops if the zeros in $v^{(j)}$ and $v^{(n+j)}$ are exploited. Note that Y is obviously the matrix of symplectic Householder vectors up a permutation of columns. Clearly, the central task in the generation of the $J-WY$ representation (3.4) is the computation of the W matrix.

The block representation of products of symplectic Householder matrices is attractive in situations where S must be applied to a matrix. Suppose that $C \in \mathbb{R}^{2n\times 2k}$. It follows that the operation $C \longleftarrow S^JC = (I+WY^TJ)^JC = C-Y(W^TJC)$ is rich in level-3 operations. If S is in factored form, S^JC is just rich in the level-2 operations of matrix-vector multiplication and outer products updates.

We mention that the $J - WY$ representation is not a generalized symplectic Householder transformation from the geometric point of view. True symplectic block reflectors are discussed in [11].

3.4. The SROSH algorithm. The steps of the SROSH algorithm are the same as those of the SRSH algorithm. The only change is that the free parameters in SRSH are replaced by optimal ones in SROSH. Let $A \in \mathbb{R}^{2n\times 2p}$ and assume that $p \le n$. Given $j \le m \le k \le 2n$ and $j' \le m' \le k' \le 2p$, let $A([j : m, k : 2n], [j' : m', k' : 2p])$ denote the submatrix obtained from A by deleting all rows except rows $j, \dots, m$ and $k, \dots, 2n$ and all columns except columns $j', \dots, m'$ and $k', \dots, 2p$.

ALGORITHM 3.14. *SROSH algorithm, factored form*
Given $A \in \mathbb{R}^{2n\times 2p}$ *with* $n \ge p$*, the following algorithm finds (implicitly) optimal symplectic Householder matrices* $T_1, \dots, T_{2p}$ *such that if* $S^J = T_{2p}^JT_p^J\dots T_{p+1}^JT_1^J$*, then* $S^JA = R$ *is* J*-upper triangular. The* J*-upper triangular part of* A *is overwritten by the* J*-upper triangular part of* R*, and the essential part of the optimal symplectic Householder vectors are stored in the zeroed portion of* A*.*

for $j = 1 : p$
  $ro = [j : n, n + j : 2n];\ co = [j : p, p + j : 2p];$
  $[c, v_1] = osh1(A(ro, [j]));$
  $A(ro, co) = A(ro, co) + cv_1v_1^TJA(ro, co);$
  $[c, v_2] = osh2(A(ro, [p + j]));$
  $A(ro, co) = A(ro, co) + cv_2v_2^TJA(ro, co);$
  if $j < n$

$A(j+1:n,j) = v_1(2:n-j+1)$;
$A(j+n:2n,j) = v_1(n-j+2:2(n-j+1))$;
$A(j+1:n,j+p) = v_2(2:n-j+1)$;
$A(j+1+n:2n,j+p) = v_2(n-j+3:2(n-j+1))$;
else
$A(j+n:2n,p) = v_1(2)$;
end
end

This algorithm requires $16m^2(n-p/3)$ flops. To clarify how A is overwritten, we give the following example:

$$A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} & r_{15} & r_{16} \\ v_1(2) & r_{22} & r_{23} & v_4(2) & r_{25} & r_{26} \\ v_1(3) & v_2(3) & r_{33} & v_4(3) & v_5(3) & r_{36} \\ v_1(4) & r_{42} & r_{43} & r_{44} & r_{45} & r_{46} \\ v_1(5) & v_2(5) & r_{53} & v_4(5) & r_{55} & r_{36} \\ v_1(6) & v_2(6) & v_3(6) & v_4(6) & v_5(6) & r_{36} \end{bmatrix}.$$

If the matrix S is required, it can be computed in $32(n^2p - np^2 + p^3/3)$ flops using backward accumulation. Note that due to (2.7), computing the product $S^J = T_{2p}^J T_p^J \ldots T_{p+1}^J T_1^J$ does not cost any more flops due of the presence of J-transpositions.

4. Error analysis of SROSH.

4.1. Background for error analysis. To carry out a rounding error analysis of the SROSH algorithm, we adopt the standard model for floating operations

$$fl(x \text{ op } y) = (x \text{ op } y)(1+\delta), \quad |\delta| \leq \mathbf{u}, \quad \text{op} = +,\ -,\ *,\ /. \tag{4.1}$$

The quantity $\mathbf{u}$ stands for the unit roundoff. In the following analysis and throughout the paper, a hat denotes a computed quantity. We now recall some important notations, conventions, and results of rounding error analysis developed in [7]. We will use these results in what follows to derive a rounding error analysis for the SROSH algorithm.

LEMMA 4.1. *If* $|\delta_i| \leq \mathbf{u}$, *and* $\rho_i = \pm 1$ *for* $i = 1:n$ *and* $n\mathbf{u} < 1$, *then*

$$\prod_{i=1}^{n} (1+\delta_i)^{\rho_i} = 1 + \theta_n,$$

where

$$|\theta_n| \leq \frac{n\mathbf{u}}{1 - n\mathbf{u}} =: \gamma_n.$$

Proof. See [7, pp. 63]. □

The θ_n and γ_n notation is used throughout this section. It is implicitly assumed that $n\mathbf{u} < 1$ (which is usually satisfied), whenever we write γ_n. Consider the inner product $s_n = x^T y$, where $x, y \in \mathbb{R}^n$. Using Lemma 4.1, the computed $\hat{s}_n$ is then given by

$$\hat{s}_n = x_1y_1(1+\theta_n) + x_2y_2(1+\theta_n') + x_3y_3(1+\theta_{n-1}) + \ldots + x_ny_n(1+\theta_2), \tag{4.2}$$

with $|\theta_i| \leq \gamma_i \leq \gamma_n$, $i = 1:n$, $\theta_n' \leq \gamma_n$, and thus each relative perturbation is certainly bounded by γ_n. Hence, we obtain

$$fl(x^Ty) = \hat{s}_n = (x+\Delta x)^T y = x^T(y+\Delta y), \quad |\Delta x| \leq \gamma_n |x|, \quad |\Delta y| \leq \gamma_n |y|, \tag{4.3}$$

where $|x|$ denotes the vector such that $|x|_i = |x_i|$ and inequalities between vectors (or matrices) hold componentwise. The result (4.3) means that computation of an inner product is a backward stable process.

For an outer product $A = xy^T$, we have $\hat{a}_{ij} = x_i y_j (1 + \delta_{ij})$, $|\delta_{ij}| \leq \mathbf{u}$, so

$$\hat{A} = xy^T + \Delta, \quad |\Delta| \leq \mathbf{u} \left| xy^T \right|. \tag{4.4}$$

The computation of an outer product is not backward stable. Nevertheless, (4.4) is a satisfactory result.

The following lemma provides some necessary rules for manipulating the $1 + \theta_k$ and γ_k terms involved in Lemma 4.1.

LEMMA 4.2. *For any positive integer k, let θ_k denote a quantity bounded according to*

$$|\theta_k| \leq \gamma_k = \frac{k\mathbf{u}}{1 - k\mathbf{u}}.$$

The following relations hold:

$$(1 + \theta_k)(1 + \theta_j) = (1 + \theta_{k+j}), \qquad \frac{1 + \theta_k}{1 + \theta_j} = \begin{cases} 1 + \theta_{k+j}, & j \leq k, \\ 1 + \theta_{k+2j}, & j > k, \end{cases}$$
$$\gamma_k \gamma_j \leq \gamma_{\min(k,j)} \text{ for } \max(j,k)\mathbf{u} \leq 1/2,$$
$$i\gamma_k \leq \gamma_{ik}, \qquad \gamma_k + \mathbf{u} \leq \gamma_{k+1}, \qquad \gamma_k + \gamma_j + \gamma_k\gamma_j \leq \gamma_{k+j}.$$

Proof. See [7, pp. 67]. □

It is straightforward to analyze matrix-vector and matrix-matrix products, once error analysis for inner products has been given. Let $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, and $y = Ax$. We have the backward error result

$$\hat{y} = (A + \Delta A)x, \qquad |\Delta A| \leq \gamma_n |A|. \tag{4.5}$$

The following result is needed.

LEMMA 4.3. *If $X_j + \Delta X_j \in \mathbb{R}^{m\times m}$ satisfies $\|\Delta X_j\|_F \leq \delta_j \|X_j\|_2$ for all j, then*

$$\left\| \prod_{j=0}^{p} (X_j + \Delta X_j) - \prod_{j=0}^{p} X_j \right\|_F \leq \left(\prod_{j=0}^{p} (1 + \delta_j) - 1 \right) \prod_{j=0}^{p} \|X_j\|_2.$$

Proof. See [7, pp. 73]. □

4.2. Error analysis for optimal symplectic Householder transformations. In the following analysis, it is not worthwhile to keep the precise value of the integer constants in the γ_k terms. We make use of the notation $\tilde{\gamma}_k = ck\mathbf{u}/(1 - ck\mathbf{u})$, where c denotes a small integer constant; see [7]. For example, we write $4\gamma_n \leq \gamma_{4n} = \tilde{\gamma}_n$.

LEMMA 4.4. *Consider the optimal symplectic Householder transformation $T_1 = I + c_1 v_1 v_1^J$ given by Algorithm 3.3. The computed $\hat{c}_1$ and $\hat{v}_1$ satisfy $\hat{v}_1(2 : 2n) = v_1(2 : 2n)$ and*

$$\hat{c}_1 = c_1(1 + \tilde{\theta}_{2n}), \qquad \hat{v}_1(1) = v_1(1)(1 + \tilde{\theta}_{2n}), \tag{4.6}$$

where $\left|\tilde{\theta}_{2n}\right| \leq \tilde{\gamma}_{2n}$.

Proof. We mention that each occurrence of δ denotes a different number bounded by $\delta \leq \mathbf{u}$. We have $\rho = sign(a_1)\, \|a\|_2$ and $v_1(1) = a_1 - \rho$, and we use the formula

$$v_1(1) = \frac{a_1^2 - \rho^2}{a_1 + \rho} = -\frac{a_2^2 + \ldots + a_{2n}^2}{a_1 + \rho}$$

to avoid cancellation errors in computing $v_1(1)$. Setting $d = a_2^2 + \ldots + a_{2n}^2$ and $e = d + a_1^2$, we get $\hat{d} = d(1+\theta_{2n-1})$, and then $\hat{e} = (\hat{d} + a_1^2(1+\delta))(1+\delta) = (d+a_1^2)(1+\theta_{2n})$ (because there is cancellation in the sum). Setting $f = \sqrt{\hat{e}}$, we obtain

$$\hat{f} = \sqrt{\hat{e}}(1+\delta) = (d + a_1^2)^{1/2}(1+\theta_{2n})^{1/2}(1+\delta) = (d + a_1^2)^{1/2}(1+\theta_{2n+1}).$$

Since $\rho = sign(a_1)(d + a_1^2)^{1/2}$, it follows that

$$\hat{\rho} = sign(a_1)\hat{f} = sign(a_1)(d + a_1^2)^{1/2}(1+\theta_{2n+1}),$$

and thus

$$\hat{\rho} = \rho(1+\theta_{2n+1}). \tag{4.7}$$

From $v_1(1) = -d/(a_1+\rho)$, we deduce

$$\hat{v}_1(1) = -\frac{\hat{d}(1+\delta)}{(a_1+\hat{\rho})(1+\delta)}.$$

Since there is no cancellation in the sum $a_1 + \hat{\rho}$ and using (4.7), we obtain

$$\hat{v}_1(1) = -\frac{\hat{d}(1+\delta)}{(a_1+\rho)(1+\theta_{2n+2})} = -\frac{d(1+\theta_{2n})}{(a_1+\rho)(1+\theta_{2n+2})}.$$

Furthermore, by applying Lemma 4.1, we get

$$\hat{v}_1(1) = -\frac{d}{a_1+\rho}(1+\theta_{6n+4}) = v_1(1)(1+\theta_{6n+4}).$$

From

$$\left|\tilde{\theta}_{2n}\right| = |\theta_{6n+4}| \leq \gamma_{6n+4} \leq \gamma_{8n} = \tilde{\gamma}_{2n},$$

we obtain

$$\hat{v}_1(1) = v_1(1)(1+\tilde{\theta}_{2n}) \text{ and } |\Delta v_1(1)| \leq |v_1(1)|\, \tilde{\gamma}_{2n}. \qquad \square$$

LEMMA 4.5. *Consider the optimal symplectic Householder transformation $T_2 = I + c_2 v_2 v_2^J$ given by Algorithm 3.4. The computed $\hat{c}_2$ and $\hat{v}_2$ satisfy $\hat{v}_2(2:2n) = v_2(2:2n)$, and*

$$\hat{c}_2 = c_2(1+\tilde{\theta}_{2n}), \quad \hat{v}_2(1) = v_2(1)(1+\tilde{\theta}_{2n}), \tag{4.8}$$

where $\left|\tilde{\theta}_{2n}\right| \leq \tilde{\gamma}_{2n}$.

Proof. We have $v_2(1) = -\xi = -(a_2^2 + \ldots + a_n^2 + a_{n+2}^2 + \ldots + a_{2n})^{1/2}$. Thus, we get $\hat{\xi} = \xi(1+\theta_{2n-1})$ and hence $\hat{v}_2(1) = -\hat{\xi} = v_2(1)(1+\tilde{\theta}_{2n})$. Similar to the results of the analysis of $c_2 = 1/(\xi u_{n+1})$, we find

$$\hat{c}_2 = \frac{(1+\delta)^2}{\hat{\xi} u_{n+1}} = \frac{(1+\delta)^2}{\xi(1+\theta_{2n-1})u_{n+1}}.$$

From Lemma 4.1, we then obtain

$$\hat{c}_2 = \frac{1}{\xi u_{n+1}}(1+\theta_{4n}) = c_2(1+\tilde{\theta}_{2n}). \qquad \square$$

The next result describes the application of a symplectic Householder transformation to a vector, and is the basis of the subsequent analysis. For convenience we here write symplectic Householder matrices in the form $T_1 = I + w_1 v_1^J$ and $T_2 = I + w_2 v_2^J$, which requires $w_1 = c_1 v_1$ and $w_2 = c_2 v_2$. Using then Lemmas 4.4 and 4.5, we obtain

$$\begin{aligned} \hat{w}_1 &= w_1 + \Delta w_1, \quad |\Delta w_1| \le \tilde{\gamma}_{2n} |w_1|, \text{ and} \\ \hat{w}_2 &= w_2 + \Delta w_2, \quad |\Delta w_2| \le \tilde{\gamma}_{2n} |w_2|. \end{aligned} \tag{4.9}$$

LEMMA 4.6. *Let $b \in \mathbb{R}^{2n}$, and consider the computation of $y = \hat{T}_1 b = (I + \hat{w}_1 \hat{v}_1^J)b$ where $\hat{v}_1$ and $\hat{w}_1$ satisfy (4.6) and (4.9), respectively. The computed $\hat{y}$ satisfies*

$$\hat{y} = (T_1 + \Delta T_1)b, \qquad \|\Delta T_1\|_F \le 3\tilde{\gamma}_{2n} \|T_1\|_2. \tag{4.10}$$

Moreover,

$$\|\Delta T_1\|_F \le 5\tilde{\gamma}_{2n} \frac{\|a\|_2}{|a_{n+1}|}. \tag{4.11}$$

Proof. We have

$$\hat{z} = fl(\hat{w}_1(\hat{v}_1^J b)) = (\hat{w}_1 + \Delta\hat{w}_1)(\hat{v}_1^J(b + \Delta b)),$$

where $|\Delta\hat{w}_1| \le \mathbf{u} |\hat{w}_1|$ and $|\Delta b| \le \gamma_{2n} |b|$. Hence

$$\hat{z} = (w_1 + \Delta w_1 + \Delta\hat{w}_1)(v_1 + \Delta v_1)^J (b + \Delta b) = w_1(v_1^J b) + \Delta z,$$

where $|\Delta z| \le \tilde{\gamma}_{2n} |w_1| \left|v_1^J\right| |b|$. It follows that

$$\hat{y} = fl(b + \hat{z}) = b + w_1(v_1^J b) + \Delta z + \Delta y_1, \qquad |\Delta y_1| \le \mathbf{u} |b + \hat{z}|.$$

We get

$$|\Delta z + \Delta y_1| \le \mathbf{u} |b| + \tilde{\gamma}_{2n} |w_1| \left|v_1^J\right| |b|. \tag{4.12}$$

Setting $\Delta y = \Delta z + \Delta y_1$ and $\Delta T_1 = (\Delta y b^T)/(b^T b)$, we obtain

$$\hat{y} = T_1 b + \Delta y = (T_1 + \Delta T_1)b, \qquad \|\Delta T_1\|_F = \frac{\|\Delta y\|_2}{\|b\|_2}.$$

From (4.12), we have

$$\|\Delta y\|_2 \le \mathbf{u} \|b\|_2 + \tilde{\gamma}_{2n} \left\| |w_1| \left|v_1^J\right| \right\|_2 \|b\|_2 \le \tilde{\gamma}_{2n}(1 + \left\| |w_1| \left|v_1^J\right| \right\|_2) \|b\|_2.$$

Since

$$\left\| |w_1| \left|v_1^J\right| \right\|_2 = \| |w_1| \|_2 \left\| |v_1^J| \right\|_2 = \|w_1\|_2 \left\|v_1^J\right\|_2 = \left\|w_1 v_1^J\right\|_2 = \|I - T_1\|_2 \le 1 + \|T_1\|_2,$$

we then get

$$\|\Delta T_1\|_F = \frac{\|\Delta y\|_2}{\|b\|_2} \le \tilde{\gamma}_{2n}(2 + \|T_1\|_2) \le 3\tilde{\gamma}_{2n} \|T_1\|_2.$$

Substituting $w_1 = c_1 v_1$ in (4.12), $\|\Delta y\|_2$ can be bounded by

$$\|\Delta y\|_2 \leq \mathbf{u} \|b\|_2 + \tilde{\gamma}_{2n} |c_1| \|v_1\|_2^2 \|b\|_2 \,,$$

and thus

$$\|\Delta T_1\|_F = \frac{\|\Delta y\|_2}{\|b\|_2} \leq \mathbf{u} + \tilde{\gamma}_{2n} |c_1| \|v_1\|_2^2 \,.$$

Since

$$|c_1| = \frac{(|a_1| - \|a\|_2)^2}{\|a\|_2 \, a_{n+1}} \text{ and } \|v_1\|_2^2 = 1 + \frac{\|a\|_2^2 - a_1^2}{(|a_1| - \|a\|_2)^2} = 2 \frac{\|a\|_2}{\|a\|_2 - |a_1|},$$

it follows that

$$|c_1| \|v_1\|_2^2 = \frac{2(\|a\|_2 - |a_1|)}{|a_{n+1}|}.$$

Hence,

$$\|\Delta T_1\|_F \leq \mathbf{u} + \tilde{\gamma}_{2n} \frac{4 \|a\|_2}{|a_{n+1}|} \leq 5\tilde{\gamma}_{2n} \frac{\|a\|_2}{|a_{n+1}|}. \qquad \square$$

REMARK 4.7. *A very interesting exploitation of (4.11) is to use a strategy of pivoting in the process of optimal symplectic Householder SR factorization, with symplectic permutations so that the ratio $\|a\|_2 / |a_{n+1}|$ is minimal.*

Next, we consider a sequence of optimal symplectic Householder transformations applied to a matrix.

LEMMA 4.8. *Let (A_k) be the sequence of matrices given by*

$$A_{k+1} = T_k A_k, \quad k = 1 : r,$$

where $A_1 = A \in \mathbb{R}^{2n \times 2m}$ and $T_k = I + w_k v_k^J \in \mathbb{R}^{2n \times 2n}$ is a (optimal) symplectic Householder matrix. Assume that

$$r\tilde{\gamma}_{2n} < \frac{1}{2}. \tag{4.13}$$

Then the computed matrix $\hat{A}_{r+1}$ satisfies

$$\hat{A}_{r+1} = S(A + \Delta A), \tag{4.14}$$

where $S = T_r T_{r-1} \dots T_1$ and

$$\|S\Delta A\|_2 \leq \frac{3r\tilde{\gamma}_{2n}}{1 - 3r\tilde{\gamma}_{2n}} \prod_{i=1}^{r} \|T_i\|_2 \|A\|_2 \,, \tag{4.15}$$

$$\|\Delta A\|_2 \leq \frac{3r\tilde{\gamma}_{2n}}{1 - 3r\tilde{\gamma}_{2n}} \prod_{i=1}^{r} \kappa_2(T_i) \|A\|_2 \,. \tag{4.16}$$

Proof. The matrix A_{r+1} is given by $A_{r+1} = T_r T_{r-1} \dots T_1 A$. By Lemma 4.6, we have

$$\hat{A}_{r+1} = (T_r + \Delta T_r) \dots (T_1 + \Delta T_1) A, \tag{4.17}$$

where each ΔT_k satisfies $\|\Delta T_k\|_F \le 3\tilde{\gamma}_{2n} \|T_k\|_2$. Using Lemma 4.3 and assumption (4.13), we obtain $\hat{A}_{r+1} = S(A + \Delta A)$, where

$$\|S\Delta A\|_2 \le ((1+3\tilde{\gamma}_{2n})^r - 1) \prod_{i=1}^{r} \|T_i\|_2 \|A\|_2 \le \frac{3r\tilde{\gamma}_{2n}}{1-3r\tilde{\gamma}_{2n}} \prod_{i=1}^{r} \|T_i\|_2 \|A\|_2 . \tag{4.18}$$

Then $\|\Delta A\|_2$ can be bounded by

$$\|\Delta A\|_2 = \left\|S^J S\Delta A\right\|_2 \le \|S\Delta A\|_2 \left\|S^J\right\|_2 \le \|S\Delta A\|_2 \prod_{i=1}^{r} \left\|T_i^{-1}\right\|_2 .$$

Using (4.18), we get

$$\begin{aligned} \|\Delta A\|_2 &\le \frac{3r\tilde{\gamma}_{2n}}{1-3r\tilde{\gamma}_{2n}} \left(\prod_{i=1}^{r} \|T_i\|_2\right) \left(\prod_{i=1}^{r} \left\|T_i^{-1}\right\|_2\right) \|A\|_2 \\ &\le \frac{3r\tilde{\gamma}_{2n}}{1-3r\tilde{\gamma}_{2n}} \prod_{i=1}^{r} \kappa_2(T_i) \|A\|_2 . \qquad \square \end{aligned}$$

Lemma 4.8 yields the standard backward and forward error results for symplectic Householder SR factorization.

THEOREM 4.9. *Let $\hat{R} \in \mathbb{R}^{2n\times 2m}$ be the computed J-upper trapezoidal SR factor of $A \in \mathbb{R}^{2n\times 2m}$ $(n \ge m)$ obtained via the SR optimal symplectic Householder algorithm SROSH. Then there exists a symplectic $S \in \mathbb{R}^{2n\times 2n}$ such that*

$$A + \Delta A = S\hat{R},$$

where

$$\|\Delta A\|_2 \le \frac{6m\tilde{\gamma}_{2n}}{1-6m\tilde{\gamma}_{2n}} \|A\|_2 \prod_{i=1}^{2m} \kappa_2(T_i). \tag{4.19}$$

The matrix S is given explicitly by $S = (\hat{T}_{2m}\hat{T}_{2m-1}\ldots\hat{T}_1)^J$, where $\hat{T}_i$ is the symplectic Householder matrix that corresponds to the exact application of the ith step of the algorithm to $\hat{A}_k$.

Proof. This is a direct application of Lemma 4.8, with T_k (resp. T_{k+m}) defined as the optimal symplectic Householder transformation that produces zeros in the desired entries in the kth (resp. the $(k+m)$th) column of the computed matrix $\hat{A}_k$. Note that in this algorithm, we do not compute the null elements of $\hat{R}$ explicitly, but rather set them to zero explicitly. Nevertheless, the conclusions of Lemmas 4.6 and 4.8 are still valid. The reason is that the elements of $\Delta T_1 b$ (respectively $\Delta T_2 b$) in Lemma 4.6 that correspond to elements that are zeroed by the optimal symplectic Householder transformation T_1 (respectively T_2) are forced to be zero, and hence we can set the corresponding rows of ΔT_1 (respectively ΔT_2) to zero, too, without compromising the bound for $\|\Delta T_1\|_F$ (respectively $\|\Delta T_2\|_F$). $\square$

Note that the matrix S in Theorem 4.9 is not computed by the optimal symplectic Householder SR factorization algorithm and is of purely theoretical interest. The fact that S is exactly symplectic makes the result so useful. When S is explicitly formed, two questions arise:

1. How close is the computed $\hat{S}$ to being symplectic?
2. How large is $A - \hat{S}\hat{R}$?

Using the analysis above, these questions are answered as follows.

THEOREM 4.10. *Let $\hat{R} \in \mathbb{R}^{2n\times 2m}$ be the computed J-upper trapezoidal SR factor of $A \in \mathbb{R}^{2n\times 2m}$ ($n \geq m$) obtained via the SR optimal symplectic Householder algorithm SROSH and $\hat{S}$ the computed factor of the product $S = \hat{T}_1^{\,J}\hat{T}_2^{\,J}\ldots\hat{T}_{2m}^{J}$ evaluated in the more efficient right-to-left order. Then*

$$\left\|\hat{S} - S\right\|_2 \leq \frac{6m\tilde{\gamma}_{2n}}{1-6m\tilde{\gamma}_{2n}} \prod_{i=1}^{2m} \|T_i\|_2 . \tag{4.20}$$

Moreover,

$$\left\|(A - \hat{S}\hat{R})\right\|_2 \leq \frac{6m\tilde{\gamma}_{2n}}{1-6m\tilde{\gamma}_{2n}}(1 + \frac{1}{1-6m\tilde{\gamma}_{2n}})\,\|A\|_2 \prod_{i=1}^{2m} \kappa_2(T_i). \tag{4.21}$$

Proof. Lemma 4.8 gives (with $A_1 = I_{2n}$)

$$\hat{S} = S(I_{2n} + \Delta I), \qquad \left\|S - \hat{S}\right\|_2 = \|S\Delta I\|_2 \leq \frac{6m\tilde{\gamma}_{2n}}{1-6m\tilde{\gamma}_{2n}} \prod_{i=1}^{2m} \|T_i\|_2 .$$

Using Theorem 4.9, we have

$$\begin{aligned}
\left\|(A - \hat{S}\hat{R})\right\|_2 &= \left\|(A - S\hat{R}) + ((S-\hat{S})\hat{R})\right\|_2 \\
&\leq \frac{6m\tilde{\gamma}_{2n}}{1-6m\tilde{\gamma}_{2n}} \|A\|_2 \prod_{i=1}^{2m} \kappa_2(T_i) + \left\|S - \hat{S}\right\|_2 \left\|\hat{R}\right\|_2 \\
&\leq \frac{6m\tilde{\gamma}_{2n}}{1-6m\tilde{\gamma}_{2n}} \|A\|_2 \prod_{i=1}^{2m} \kappa_2(T_i) + \frac{6m\tilde{\gamma}_{2n}}{1-6m\tilde{\gamma}_{2n}} \prod_{i=1}^{2m} \|T_i\|_2 \left\|\hat{R}\right\|_2 .
\end{aligned}$$

From Lemma 4.8, we have

$$\left\|\hat{R}\right\|_2 \leq \frac{1}{1-6m\tilde{\gamma}_{2n}} \|A\|_2 \prod_{i=1}^{2m} \|T_i\|_2 .$$

Since $\kappa_2(T_i) = \|T_i\|_2 \left\|T_i^{-1}\right\|_2 = \|T_i\|_2 \left\|T_i^{J}\right\|_2 = \|T_i\|_2^2$, the relation (4.21) is then straightforward. ☐

REMARK 4.11.

1. *The relation (4.20) gives a bound for the loss of J-orthogonality of the computed factor $\hat{S}$, and shows that $\hat{S}$ may be very close to a symplectic matrix if the optimal symplectic Householder transformations used in the process are well-conditioned.*
2. *The relation (4.21) gives a bound for the backward error in the factorization. This backward error may be small if the optimal symplectic Householder transformations involved in the process are well-conditioned.*
3. *The condition number of an optimal symplectic Householder transformation is minimal (see [13]), and this error analysis shows that their use in the SROSH algorithm constitutes the best choice.*
4. *Lemma 4.6 suggests a pivoting strategy for reinforcing the numerical accuracy of the algorithm.*

TABLE 4.1
Residual errors in SR factors computed by SROSH for Example 4.12.

n	$\|S^J S - I\|_2^{\text{SROSH}}$	$\|A - SR\|_2^{\text{SROSH}}$
8	$1.464898e-015$	$1.194492e-014$
9	$1.464898e-015$	$1.749372e-014$
10	$1.464898e-015$	$3.158085e-014$
11	$1.464898e-015$	$2.842371e-014$
12	$1.464898e-015$	$6.759145e-014$
11	$1.464898e-015$	$1.149066e-013$
12	$1.464898e-015$	$2.722963e-013$
13	$1.464898e-015$	$2.692070e-013$
14	$1.464898e-015$	$4.746886e-012$
15	$2.031758e-015$	$4.711156e-012$

4.3. A numerical example. To illustrate numerically the J-orthogonality and backward error, we consider the following example.

EXAMPLE 4.12.

$$A = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

with

$$M_{11} = eye(n);$$
$$M_{22} = diag(e^{1/2}, e, \dots, e^{n/2});$$
$$M_{12} = \begin{bmatrix} 1 & & & \\ e^{-1} & 1 & & \\ & \ddots & \ddots & \\ & & e^{-1} & 1 \end{bmatrix}; \qquad M_{21} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{bmatrix},$$

where eye is the MATLAB identity function. Errors for this example are shown in Table 4.1

The second example is $A = rand(2n, 2n)$, where $rand$ is the MATLAB random function. Errors in this example are summarized in Table 4.2.

5. Conclusion. An error analysis of SROSH algorithm is presented. Moreover, it is showed that the loss of J-orthogonality of the computed symplectic factor $\hat{S}$ and the backward error of the factorization are bounded in terms of the condition number of optimal symplectic Householder transformations involved in the process. From this point of view, the free parameters as taken in optimal symplectic Householder transformations constitute the best choice. The study led us also to a pivoting strategy for increasing the accuracy of the algorithm. This will be investigated in a forthcoming paper. Computational aspects of the SROSH algorithm are studied. Storage, complexity, different implementations, factored form, block representation are discussed.

Acknowledgement. The authors are grateful to an anonymous referee for his useful comments and suggestions, which greatly improved the presentation.

TABLE 4.2
Residual errors in SR factors computed by SROSH for random $2n$-by-$2n$ matrices.

n	$\|S^J S - I\|_2^{\mathrm{SROSH}}$	$\|A - SR\|_2^{\mathrm{SROSH}}$
10	$1.422043e-015$	$4.488151e-016$
11	$3.920943e-015$	$2.942877e-015$
12	$1.213806e-012$	$5.514552e-014$
13	$2.642990e-011$	$1.264768e-012$
14	$4.465041e-013$	$2.210450e-013$
15	$1.878234e-010$	$1.174566e-011$
16	$1.859176e-090$	$2.665177e-011$
17	$1.018534e-090$	$2.708870e-013$
18	$5.868227e-010$	$1.877163e-011$
19	$4.827328e-010$	$1.019172e-011$
20	$4.805909e-011$	$1.769842e-013$

REFERENCES

[1] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[2] Å. BJÖRCK, *Numerics of Gram-Schmidt orthogonalization*, Linear Algebra Appl., 197/198 (1994), pp. 297–316.
[3] A. BUNSE-GERSTNER AND V. MEHRMANN, *A symplectic QR-like algorithm for the solution of the real algebraic Riccati equation*, IEEE Trans. Automat. Control, AC-31 (1986), pp. 1104–1113.
[4] J. DELLA-DORA, *Numerical linear algorithms and group theory*, Linear Algebra Appl., 10 (1975), pp. 267–283.
[5] S.K. GODUNOV AND M. SADKANE, *Numerical determination of a canonical form of a symplectic matrix*, Siberian Math. J., 42 (2001), pp. 629–647.
[6] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, third ed., The Johns Hopkins University Press, Baltimore, 1996.
[7] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, second ed., SIAM, Philadelphia, 2002.
[8] D. S. MACKEY, N. MACKEY AND F. TISSEUR, *$\mathbb{G}$-Reflectors: Analogues of Householder transformations in scalar product spaces*, Linear Algebra Appl., 385 (2004), pp. 187–213.
[9] C. PAIGE AND C. VAN LOAN, *A Schur decomposition for Hamiltonian matrices*, Linear Algebra Appl., 41 (1981), pp. 11–32.
[10] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Co., Boston, 1996.
[11] M. SADKANE AND A. SALAM, *A note on symplectic block reflectors*, Electron. Trans. Numer. Anal., 33 (2009), pp. 45–52.
http://etna.math.kent.edu/vol.33.2008-2009/pp45-52.dir/.
[12] A. SALAM, *On theoretical and numerical aspects of symplectic Gram-Schmidt-like algorithms*, Numer. Algorithms, 39 (2005), pp. 237–242.
[13] A. SALAM AND E. AL-AIDAROUS AND A. ELFAROUK, *Optimal symplectic Householder transformations for SR-decomposition*, Linear Algebra Appl., 429 (2008), pp. 1334–1353.
[14] C. VAN LOAN, *A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix*, Linear Algebra Appl., 61 (1984), pp. 233–251.
[15] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.