ROCZNIKI POLSKIEGO TOWARZYSTWA MATEMATYCZNEGO
Seria III: MATEMATYKA STOSOWANA XXXVIII (1995)

PAWEŁ ZIELIŃSKI

Opole

KRYSTYNA ZIĘTAK

Wrocław

The polar decomposition—properties, applications and algorithms

(*Received 10.05.1994*)

Abstract. In the paper we review the numerical methods for computing the polar decomposition of a matrix. Numerical tests comparing these methods are included. Moreover, the applications of the polar decomposition and the most important its properties are mentioned.

1. Introduction. In recent years interests in the *polar decomposition* have increased. Many interesting papers have appeared on properties, applications and numerical methods for this decomposition. In the paper we review the most important results concerning this very useful tool. Also we present numerical experiments comparing several algorithms for computing it.

The polar decomposition was introduced by Autonne [1] in 1902. A thorough discussion of the history of it is given in Horn and Johnson [29, Sect. 3.0]. Let A be an arbitrary complex matrix, $A \in \mathcal{C}^{m\times n}$. A polar decomposition of A is a factorization

$$A = UH, \tag{1}$$

AMS Subject Classification: 65F30, 65F35, 65F25.

Keywords and phrases: polar decomposition, singular value decomposition, complete orthogonal decomposition, approximation of matrices, Procrustes problem, matrix square root, matrix sign, Newton's method, optimal scaling, acceleration, Schultz iteration.

where $H \in \mathcal{C}^{n\times n}$ is Hermitian positive semi-definite matrix, $H = H^H$ and $x^H H x \geq 0$ for every $x \in \mathcal{C}^n$, and U, $U \in \mathcal{C}^{m\times n}$, satisfies one of the following conditions

$$U^H U = I_n \quad \text{for } m \geq n, \tag{2}$$

$$U U^H = I_m \quad \text{for } m \leq n, \tag{3}$$

where I_n denotes the identity matrix of order n. The condition (2) means that the matrix U has orthonormal columns. If $m = n$ then U is unitary, so $U^{-1} = U^H$. For $m < n$ the matrix U has orthonormal rows (see (3)). The matrix H is called a *Hermitian factor*, U is an *unitary factor* from the polar decomposition of A. If A is real then U and H are real. Thus H is a *symmetric factor* and U is an *orthogonal factor*. Instead of the notion *a Hermitian positive semi-definite matrix* we will use a short version *a positive semi-definite matrix*.

The polar decomposition (1) is very well-known for the case $m \geq n$ (see for example Gantmacher [12, Sect. 9], Horn and Johnson [28, Sect. 7], [29, Sect. 3]). For $m < n$ besides (1) one considers also the following definition (see Horn and Johnson [29, p. 152])

$$A = HU,$$

where H is positive semi-definite and U has orthonormal rows. This definition has inspired Higham and Schreiber [27] to introduce the form (1) also for $m < n$ (see (3)). The consistency of the definition (1) can be seen in the result that for any m and n the unitary factor U from (1) has some common approximation properties (see for example Higham and Schreiber [27]). In this paper we will deal only with the decomposition (1).

The polar decomposition is a generalization, to complex matrices, of the trygonometric representation of a complex number $z = re^{i\varphi}$, $r \geq 0$, with $e^{i\varphi}$ becaming the unitary factor U and $r = |z|$ corresponding to the Hermitian factor H.

Let A have the following *singular value decomposition*, called the SVD (see Golub and Van Loan [14, p. 71], Horn and Johnson [29, p. 144], Kiełbasiński and Schwetlick [32, p. 41])

$$A = P\Sigma Q^H, \tag{4}$$

where P and Q are unitary matrices, $P \in \mathcal{C}^{m\times m}$, $Q \in \mathcal{C}^{n\times n}$, Σ is a rectangular diagonal matrix with nonnegative diagonal elements ordered decreasingly

$$\Sigma = \operatorname{diag}(\sigma_j), \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq 0.$$

Elements $\sigma_j = \sigma_j(A)$ are called *singular values* of A. Let $m \geq n$. Then Σ has the form

$$\Sigma = \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix}, \quad \Sigma_1 \in \mathcal{R}^{n\times n}.$$

We divide P into two blocks $P = [P_1, P_2]$, $P_1 \in \mathcal{C}^{m\times n}$. Then we can derive the polar decomposition of A from the following formula

$$A = P_1\Sigma_1 Q^H = (P_1 Q^H)(Q\Sigma_1 Q^H), \tag{5}$$

so $U = P_1 Q^H$ and $H = Q\Sigma_1 Q^H$. If $m < n$ then Σ has the form $\Sigma = [\Sigma_1, 0]$, $\Sigma_1 \in \mathcal{R}^{m\times m}$. Let the matrix Q be divided into two blocks $Q = [Q_1, Q_2]$, $Q_1 \in \mathcal{C}^{n\times m}$. Then the polar decomposition of A can be obtained from the following formula

$$A = (PQ_1^H)(Q_1\Sigma_1 Q_1^H). \tag{6}$$

Therefore $U = PQ_1^H$ and $H = Q_1\Sigma_1 Q_1^H$.

From (5) and (6) it follows that the polar decomposition always exists and it can be computed by means of the SVD. However such an approach can be unnecessarily expensive, especially when A is close to an unitary matrix. In Sections 3 and 4 we present other methods for computing H and U. These methods depend on a construction of a sequence of matrices convergent to U. Then H can be obtained immediately from (1).

The polar decomposition has interesting properties and applications. We recall them in Section 2. In Section 5 we present numerical experiments. The computations were done by the first author in his master thesis written under direction of the second author (see Zieliński [43]).

In the present paper we do not deal with the perturbations bounds for the polar decomposition. This problem is considered in the papers of Barrlund [3], Higham [17], Kenney and Laub [30], and Mathias [34].

In the remaining part of the paper we assume $m \geq n$, although some properties and algorithms are valid also for the case $m < n$.

2. Properties and applications of the polar decomposition. The polar decomposition is an important tool in various applications. We will mention some of them.

Let $A \in \mathcal{C}^{m\times n}$, $m \geq n$, have the polar decomposition (1). The Hermitian factor H and the unitary factor U have the following properties (see for example Higham [17]). The matrix H is the unique positive semi-definite square root of $A^H A$ (the definition and properties of the *square roots of matrices* are given for example in Higham [18], [19], Horn and Johnson [29, p. 419])

$$H = (A^H A)^{1/2}. \tag{7}$$

This implies that the Hermitian factor H is always unique. On the other hand the unitary factor U is unique if and only if the matrix A has full rank, rank $A = n$ (see Higham and Schreiber [27]).

Using (7) Higham [17] has proposed the following method for computing the positive definite square root of a positive definite matrix B

(i) compute the Cholesky decomposition of B, $B = LL^H$, where L is lower triangular,

(ii) compute the Hermitian factor H_L from the polar decomposition of L^H.

Then $B^{1/2} = H_L$.

The polar decomposition of A can be applied to computation of the SVD of A (see Higham and Papadimitriou [25]). Let U and H be determined as in (1). We compute the spectral decomposition of H, $H = VDV^H$, where V – unitary, $D = \mathrm{diag}(\lambda_j)$ and the eigenvalues λ_j are ordered decreasingly. The eigenvalues λ_j are nonnegative because H is positive semi-definite. Therefore A has the reduced SVD (5) with $P_1 = UV$, $Q = V$ and $\Sigma_1 = D$. This way of finding (5) can be cheaper than traditional approach when we use a special parallel method for the polar decomposition. Such a method will be described in the next section.

Construction of *block reflectors* is another example of the applications of the polar decomposition (see Schreiber and Parlett [39]). The block reflectors are a generalization of Householder transformations.

In our opinion the most important applications of the polar decomposition are connected with its approximation properties. We now describe them shortly.

The unitary factor has the following property (see Higham [17]). Let $A, B \in \mathcal{C}^{m\times n}$ and let $C = B^H A$ have the polar decomposition $C = U_C H_C$. Then for arbitrary unitary matrix Z the following inequalities hold

$$\|A - BU_C\|_F \leq \|A - BZ\|_F \leq \|A + BU_C\|_F, \tag{8}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\|A\|_F = (\sum_{i,j} |\, a_{ij}\,|^2)^{1/2}$. For $m = n$ and $B = I$ we obtain the following corollary. Let $A \in \mathcal{C}^{n\times n}$ have the polar decomposition (1). Then

$$\|A - U\|_F = \min_{Z-unitary} \|A - Z\|_F. \tag{9}$$

The unitary factor U is the best unitary approximation to A with respect to the Frobenius norm. It is also true for arbitrary unitarily invariant norm, i.e. U is the best approximation to A with respect to all unitarily invariant norms (see Fan and Hoffman [9], Higham [17], [22]). The norm $\|\cdot\|$ is unitarily invariant if $\|A\| = \|VAW\|$ for arbitrary unitary matrices V and W (see Horn and Johnson [28, Sect. 7], [29, p. 203]). The Frobenius norm $\|\cdot\|_F$ and the spectral norm $\|\cdot\|_2$ are unitarily invariant. We recall that $\|A\|_2 = \sigma_1(A), \|A\|_F^2 = \sum_{i=1}^n \sigma_i^2(A)$.

The left inequality in (8) means that if $\|\cdot\|$ is the Frobenius norm then the minimum

$$\min_{Z-unitary} \|A - BZ\| \tag{10}$$

is reached for the unitary factor from the polar decomposition of $B^H A$. Unfortunately, it is not true for an arbitrary unitarily invariant norm (see Mathias [34], Rao [37]). The solution of the problem (10) for arbitrary unitarily invariant norm is more complicated than for the Frobenius norm. Watson [41] has proposed a method, based on quadratic programming, for computing the solutions of the problem (10) for A, B – real and the Schatten c_p-norms. We recall that the c_p-norm of A is equal to the l_p-norm of the vector of the singular values of A.

If A has full rank, rank $A = n$, then the problem (9) has unique solution. The same holds for the problem (10) with respect to the Frobenius norm, if rank $B^H A = n$ (see Higham [22]).

The Hermitian factor also has some important approximation properties. They are related to an approximation of a matrix A by positive semi-definite matrices

$$\delta(A) = \min_{X - \text{ psd}} \|A - X\| \tag{11}$$

where the minimum is taken over all positive semi-definite matrices X. Investigations connected with such an approximation were inspired by Halmos [15]. It is a difficult problem and it has an easy solution only for some special cases (see Bhatia and Kittaneh [4], Higham [17], [21], [22]). Let A be normal, $AA^H = A^H A$, and let $A = B + iC$ be its *Cartesian represantation*, i.e. $B^H = B$ and $C^H = C$. In this case the *positive part* of B

$$X_F = \frac{1}{2}[B + (B^H B)^{1/2}] \tag{12}$$

is the solution of the problem (11) for all unitarily invariant norms (see Bhatia and Kittaneh [4]). For the spectral norm it was shown by Halmos [15]. We recall that $(B^H B)^{1/2}$ is equal to the Hermitian factor of B. If A is Hermitian then $C = 0$, hence $X_F = (1/2)(A + H)$, where H is the Hermitian factor of A.

The matrix X_F is not any solution of (11) in the general case for arbitrary A and arbitrary unitarily invariant norm, in particular for the spectral norm. However, for the Frobenius norm the matrix (12) is always the unique solution of (11) for arbitrary matrix A (see Higham [21]).

For the spectral norm the solution of (11) can be not unique, even for A normal. Higham [21] has proposed a method for solving the problem (11) for an arbitrary real matrix A and the spectral norm. His method is based on a characterization of the approximant which is often called the *Halmos positive approximant*. The Halmos positive approximant is generally different from X_F. Let $\delta_2(A)$ denote the minimum (11) for the spectral norm. Then we have for arbitrary A (see Higham [21])

$$\delta_2(A) \leq \|A - X_F\|_2 \leq 2\delta_2(A).$$

Problems (9), (10) and (11) have practical applications. The approximation considered in (10) is known in factor analysis as the *unitary Procrustes problem* (see Rao [37]). The problem (9) appears in aerospace computations (see Bar-Itzhack and Fegley [2], Higham [17]). Namely, an orthogonal matrix $D \in \mathcal{R}^{3\times 3}$ (*direction cosine matrix*, DCM), which transforms vectors from one coordinate system to another, can lose its orthogonality because of errors incurred in computation of D. Then the computed $\hat{D}$ is replaced by the nearest orthogonal matrix, that is, by the orthogonal factor of $\hat{D}$.

The approximation by means of positive semi-definite matrices plays important role in the optimization methods. Let $F : R^n \to R$. Newton's method for the minimization of $F(x)$ requires at each stage computation of a search direction $p^{(k)}$ (see Gill, Murray and Wright [13]). For this purpose one solves the linear system (see Higham [17])

$$G_k p^{(k)} = -g^{(k)},$$

where $g^{(k)} = \nabla F(x^{(k)})$ is the gradient vector and

$$G_k = \left[\frac{\partial^2 F(x^{(k)})}{\partial x_i \partial x_j}\right]$$

is the symmetric Hessian matrix. Difficulties occur when G_k is not positive definite since $p^{(k)}$, if defined, need not be a descent direction. It is suggested that in such a situation G_k should be replaced by its Hermitian factor. If G_k is nonsingular then its Hermitian factor is positive definite.

Examples of other applications of the polar decomposition are mentioned in Higham and Papadimitriou [26].

Some methods for computing the polar decomposition are closely related to the methods for the matrix sign A. Therefore we now recall the definition of the matrix sign A introduced by Roberts in 1971 as a tool for solving the *algebraic Riccati equation* (see Byers [6], Roberts [38])

$$G + A^T X + XA - XFX = 0,$$

where A, G and F are known real square matrices. Let a complex matrix $A \in \mathcal{C}^{n\times n}$ have the eigenvalues λ_j and let

$$d_j = \begin{cases} +1 & \text{if } Re\ \lambda_j > 0, \\ -1 & \text{if } Re\ \lambda_j < 0. \end{cases} \tag{13}$$

We assume that the matrix A has no pure imaginary eigenvalues. Then the matrix sign A is defined as follows

$$S = \operatorname{sign} A = T^{-1}\operatorname{diag}(d_j)T,$$

where T is a matrix which transforms A into Jordan form $J = TAT^{-1}$. We choose the ordering of the eigenvalues in (13) consistently with their ordering in the matrix J. Every square matrix can be written in the form

$A = SN$, where $S = \operatorname{sign} A$ and N is an appropriate matrix (see Higham [24]). There are some anologies and relationships between this factorization and the polar decomposition.

3. Algorithms for the polar decomposition. We now review some methods for computing the polar decomposition of a matrix $A \in \mathcal{C}^{m\times n}$, $m \geq n$. In fact we need only methods for finding the unitary factor U because the Hermitian factor H can be obtained from the formula (compare (1))

$$H = \frac{1}{2}(U^H A + A^H U).$$

For this purpose one does not use the formula $H = U^H A$ because on account of rounding errors the computed matrix H could be not Hermitian.

Let $A \in \mathcal{C}^{m\times n}$, $m > n$, rank $A = n$, have the following *QR factorization* $A = QR$, where R is upper triangular, Q has orthonormal columns. Then A has the polar decomposition

$$A = (QU_R)H_R, \tag{14}$$

where

$$R = U_R H_R \tag{15}$$

is the polar decomposition of R. The matrix R is nonsingular because rank $A = n$. If A is not full rank then at first we compute its complete orthogonal decomposition, shortly called the COD, (see Golub and Van Loan [14, p. 236])

$$A = P\begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} Q^H, \tag{16}$$

where $P \in \mathcal{C}^{m\times m}$ and $Q \in \mathcal{C}^{n\times n}$ are unitary, $R \in \mathcal{C}^{r\times r}$ is nonsingular upper triangular, $r = \operatorname{rank} A$. Let R have the polar decomposition (15). Then A has the following unitary factor U (see Higham and Schreiber [27])

$$U = \left[P_1 U_R, P_2 \begin{bmatrix} I_{n-r} \\ 0 \end{bmatrix}\right] Q^H$$

and the Hermitian factor $H = Q_1 H_R Q_1^H$, where $P = [P_1, P_2]$, $Q = [Q_1, Q_2]$, $P_1 \in \mathcal{C}^{m\times r}$ and $Q_1 \in \mathcal{C}^{n\times r}$. Therefore in both cases the problem of computing the polar decomposition of A can be reduced to the problem with a square triangular nonsingular matrix R of order $r = \operatorname{rank} A$. Therefore we will pay attention mostly to the methods for square nonsingular matrices.

We now present iterative methods for finding U. In these methods one constructs a sequence X_k convergent to U. Let ϵ be the machine precision, so $1+\epsilon$ is the smallest number in a computer such that $1+\epsilon > 1$. Let δ be a con-

vergence tolerance, $\delta \geq \epsilon$. The iterative process is continued until we obtain

$$\|X_{k+1} - X_k\|_1 \leq \delta \|X_{k+1}\|_1 \tag{17}$$

where $\|\cdot\|_1$ is the matrix norm induced by the vector l_1-norm

$$\|A\|_1 = \max_j \sum_i |a_{ij}|.$$

In 1971 Björck and Bowie [5] proposed a family of iterative methods for computing the orthogonal factor U of a complex matrix A, $A \in \mathcal{C}^{m\times n}$, $m \geq n$. These methods are based on the identity

$$e^{i\varphi} = z(1-q)^{-1/2}, \tag{18}$$

where $z = |z|e^{i\varphi}$ is a complex number, $q = 1 - |z|^2$. The Taylor series of the right side of (18) has the form

$$e^{i\varphi} = z\left(1 + \frac{1}{2}q + \frac{3}{8}q^2 + \cdots + (-1)^p \binom{-\frac{1}{2}}{p} q^p + \cdots\right). \tag{19}$$

Let the unitary factor U of A be expressed in the form (compare (18) and (19))

$$U = A(I - (I - A^H A))^{-1/2} = A(I + \frac{1}{2}T + \frac{3}{8}T^2 + \cdots), \tag{20}$$

where $T = I - A^H A$. The expressions (20) lead to the following family of iterative methods of order $p+1$ (see Björck and Bowie [5])

$$X_{k+1} = X_k\left(I + \frac{1}{2}T_k + \frac{3}{8}T_k^2 + \cdots + (-1)^p \binom{-\frac{1}{2}}{p} T_k^p\right) \equiv X_k g_p(T_k), \tag{21}$$

where $X_0 = A, T_k = I - X_k^H X_k$. For $p = 1$ we obtain

$$X_{k+1} = X_k(I + \frac{1}{2}(I - X_k^H X_k)), \quad k = 0, 1, \ldots \tag{22}$$

This method is quadratically convergent.

The method (21) is unitarily invariant in a certain sense (see Björck and Bowie [5]). Namely, let $X_0 = A$ have the SVD (4). Then $X_k = PD_kQ^H$, where the D_k are real diagonal matrices, $D_0 = \Sigma$, and $D_{k+1} = D_k g_p(I - D_k^2)$. Thus

$$\lim_{k\to\infty} X_k = PQ^H = U \quad \text{if and only if} \quad \lim_{k\to\infty} D_k = I.$$

Therefore a convergence of (21) depends on a convergence of the corresponding scalar iteration

$$z_{k+1} = z_k g_p(1 - |z_k|^2).$$

For $\{X_k\}$, generated by the method (21), to converge to U it is necessary and sufficient that z_k converges to 1 for all starting values $z_0 = \sigma_j(A), j =$

$1, \ldots, n$ (see Björck and Bowie [5]). If rank $A = n$ then $\|I - A^H A\| < 2$ is sufficient for the convergence of (22).

Higham [17] has considered another method for computing the unitary factor U of a complex nonsingular matrix A. His method is inspired by Newton's method for computing a square root of a positive number c

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{c}{x_k}\right) \quad k = 0, 1, \ldots \tag{23}$$

Let A be a nonsingular complex matrix and let $X_0 = A$. In *Higham's method* one constructs the sequence

$$X_{k+1} = \frac{1}{2}\left(X_k + X_k^{-H}\right) \quad k = 0, 1, \ldots, \tag{24}$$

where X_k^{-H} denotes $(X_k^{-1})^H$. The sequence $\{X_k\}$ is quadratically convergent to U. If A is nearly unitary only a few iterations are needed. The relation (24) can be written in the form

$$X_{k+1} = \frac{1}{2} X_k \left(I + (X_k^H X_k)^{-1}\right) \quad k = 0, 1, \ldots \tag{25}$$

This simple modification, proposed by Gander [11], extends Higham's method to the case of rectangular matrices A of full rank, $m > n$. Other modifications of the method (24) are described in the next section. The method (25) works well if A is not too ill-conditioned. The presence of $X_k^H X_k$ in (25) is a disadvantage because the condition-squaring effect of forming and inverting this matrix reduces the numerical accuracy of the process.

Instead of Newton's method we can apply Halley's method to compute the square root of a positive number c (see Gander [10])

$$x_{k+1} = x_k \frac{x_k^2 + 3c}{3x_k^2 + c}. \tag{26}$$

Halley's method has the order of convergence 3. Using (26) we obtain the following *Halley method* for computing U (see Gander [11])

$$X_{k+1} = X_k (X_k^H X_k + 3I)(3X_k^H X_k + I)^{-1} \quad k = 0, 1, \ldots, \tag{27}$$

where $X_0 = A$ is nonsingular. This method is applicable also to rectangular matrices of full rank. Though Halley's method converges cubically, global convergence can be poor because at initial iterations the error is roughly reduced by a factor 3 which can be too slow for practical purposes (see Gander [11]).

Methods (22), (25) and (27) have some common distinctive feature. Namely, the matrices X_{k+1}, constructed by these methods, can be expressed in the form

$$X_{k+1} = X_k \, h(X_k^H X_k), \tag{28}$$

where $h(t)$ is an appropriate real rational function and $X_0 = A$. Therefore Gander [11] considers the following family of the method of the type (28) with the following function h

$$h(x^2) = \frac{2f - 3 + x^2}{f - 2 + fx^2}, \tag{29}$$

where f is a parameter. For $f = 0$ we obtain (22), for $f = 2$ we have (25). The method (27) follows from (28) for $f = 3$. If $A \in \mathcal{C}^{m\times n}$ has rank n and

$$X_0 = \alpha A \tag{30}$$

for appropriate scaling parameter α then the sequence (28) generated by the function (29)

$$X_{k+1} = X_k[(2f - 3)I + X_k^H X_k][(f - 2)I + fX_k^H X_k]^{-1} \tag{31}$$

is at least quadratically convergent to the orthogonal factor U for every $f \neq 1$ (see Gander [11]). The algorithm (31) is unitarily invariant as the method (21).

As we have mentioned the method (24) is based on Newton's method for computing a positive square root of $c = 1$ (see (23)). However, the method (31) is related to Newton's method applied to the scalar equation

$$v(x) \equiv x^\nu(x^2 - 1) = 0.$$

It is easy to verify that (see (29))

$$xh(x^2) = x - v(x)/v'(x) \quad \text{for} \quad \nu = (2 - f)/(f - 1). \tag{32}$$

Therefore it is not surprising that the method (31) is at least of order 2. The method of Björck and Bowie (22) corresponds to $\nu = -2$, Higham's method (25) to $\nu = 0$, Halley's method (27) to $\nu = -0.5$.

The method (31) can be applied also to matrices with deficient rank, rank $A < n$, without having to determine in advance the numerical rank of A. It is possible by appropriate choice of f. We call this method *Gander's method*. The parameter f should be choosen in the following way (see Gander [11])

	$\epsilon \le \delta \le 10\epsilon$ $step = \left\lceil \frac{\ln g}{\ln 0.1} \right\rceil$	$10\epsilon < \delta \le 10^4\epsilon$ $step = \left\lceil \frac{\ln g}{\ln(\epsilon/\delta)} \right\rceil$	$10^4\epsilon < \delta \le 0.01$ $step = \left\lceil \frac{\ln g}{\ln 10^{-5}} \right\rceil$
$0 \le k < step$	$f = 2.1$	$f = 2 + \epsilon/\delta$	$f = 2.0001$
$k = step$	$f = 3$	$f = 1/\sqrt{8\epsilon/\delta}$	$f = 100$
$k > step$	$f = 3$	$f = 3$	$f = 3$

where ϵ is the machine precision, δ is a convergence tolerance (see (17)), $\epsilon \le \delta \le 0.01$, the parameter g depends on the rank and the inverse condition of A

$$g = \max\{rcon, \epsilon\},$$

where *rcon* is equal to 0 if A is not full rank and *rcon* has to be an estimation of the inverse condition number σ_n/σ_1 otherwise. Hence in Gander's method one performs (31) with appropriate f in the initial iterations and Halley's method ($f = 3$) in the final iterations. The initial matrix is scaled (see (30)). Gander has proposed the following choice of α

$$\alpha = \sqrt{n}/\|A\|_\infty, \tag{33}$$

where $\|\cdot\|_\infty$ is the matrix norm induced by the vector l_∞-norm

$$\|A\|_\infty = \max_i \sum_j |a_{ij}|.$$

It is known that α^{-1} is a lower bound for the spectral norm $\|A\|_2$ (see Golub and Van Loan [14, p. 57]).

Here is the scheme of Gander's method written in the pseudo–Pascal notation for real matrices (we now do not assume that $m \geq n$):

Gander's method

```
procedure GANDER (A,U,H; m,n; δ; rcon; k)
{ rcon - estimate for inverse condition number σn/σ1,
  zero for rank deficient matrix A }
var
  g, f, ma, : real;
  step : integer;
  Y, H1, Xk, Xk+1 : array[1..Nmax,1..Nmax];
begin
  k := 0;
  if δ < ε then δ := ε;
  if δ > 0.01 then δ := 0.01;
  g := max(rcon, ε);
  if δ ≤ 10ε then
     begin
        f := 2.1; ma := 3;
        step := trunc(ln g/ ln 0.1) + 1
     end
  else
    if δ > 10000ε then
       begin
          f := 2.00001; ma := 100;
          step := trunc(ln g/ ln 0.00001) + 1
       end
    else
      begin
         f := 2 + ε/δ; ma := 1/√(8ε/δ);
         step := trunc(ln g/ ln(ε/δ)) + 1
      end;
  if m ≥ n then
     begin
       Xk := 0; Xk+1 := (√n/||A||∞)A;
```

```
      while ||X_{k+1} - X_k||_1 > δ||X_{k+1}||_1 do
        begin
          X_k := X_{k+1}; Y := X_k^T X_k;
          k := k + 1;
          X_{k+1} := X_k((2f - 3)I + Y)((f - 2)I + fY)^{-1};
          if k = step then f := ma
            else
              if k > step then f := 3
        end
    end
  else { m < n }
    begin
      X_k := 0; X_{k+1} := (√n/||A||_∞)A;
      while ||X_{k+1} - X_k||_1 > δ||X_{k+1}||_1 do
        begin
          X_k := X_{k+1}; Y := X_k X_k^T;
          k := k + 1;
          X_{k+1} := ((2f - 3)I + Y)((f - 2)I + fY)^{-1} X_k;
          if k = step then f := ma
            else
              if k > step then f := 3
        end
    end;
  U := X_{k+1};
  H_1 := U^T A;
  H := (H_1 + H_1^T)/2
end; {GANDER}
```

We now present a method which is very attractive for parallel computation (see Higham and Papadimitriou [26]). In this method the following sequence is constructed

$$X_{k+1} = \frac{1}{p} X_k \sum_{i=1}^{p} \frac{1}{\xi_i} (X_k^H X_k + \alpha_i^2 I)^{-1}, \tag{34}$$

where $X_0 = A \in \mathcal{R}^{m \times n}$, rank $A = n$, p is a fixed natural number,

$$\xi_i = \frac{1}{2}\left(1 + \cos\frac{(2i-1)\pi}{2p}\right), \quad \alpha_i^2 = \frac{1}{\xi_i} - 1, \quad 1 \le i \le p.$$

For p large the parameters α_i and ξ_i should be calculated from stable formulae

$$\xi_i = \cos^2\frac{(2i-1)\pi}{4p}, \quad \alpha_i^2 = \tan^2\frac{(2i-1)\pi}{4p}.$$

The sequence $\{X_{k+1}\}$ is convergent to U with the order of convergence $2p$. If $p = 1$ then from (34) we obtain

$$X_{k+1} = 2X_k(X_k^H X_k + I)^{-1}.$$

This sequence is related to the sequence $\{\hat{X}_k\}$ generated by Higham's method (25) in the following way

$$X_k^+ = \hat{X}_k^H \quad \text{for } k = 1, \ldots,$$

where X_k^+ is the Moore-Penrose generalized inverse of X_k. The convergence of the sequence (34) can be faster by introducing some acceleration parameters (see Higham and Papadimitriou [25], [26]).

The method (34) is a simple modification of the method, based on the Padé approximation of the function $1/\sqrt{1-z}$, for finding the matrix sign A (see Pandey, Kenney and Laub [36]).

In the next section we consider an acceleration only for Higham's method.

4. Modifications of Higham's method. Let $A \in \mathcal{C}^{n\times n}$ be nonsingular. Although Higham's method (24) is quadratically convergent, the convergence can be slow initially. Therefore Higham [17] introduces scaling which accelerates the convegence. The scaling is perfomed at the start of each iteration. Namely, instead of (24) one constructs the sequence

$$X_{k+1} \equiv X_{k+1}(\gamma_k) = \frac{1}{2}\left(\gamma_k X_k + \frac{1}{\gamma_k} X_k^{-H}\right), \tag{35}$$

where γ_k are acceleration parameters. Kenney and Laub [30] give a thorough analysis of scaling Higham's method. The choice of optimal acceleration parameters

$$\gamma_k^{(opt)} = (\|X_k^{-1}\|_2 / \|X_k\|_2)^{1/2}, \tag{36}$$

effects very rapid convergence, even when A is ill conditioned with respect to inversion. However in practice the optimal acceleration parameters must be approximated. Therefore one considers the following parameters (see Gander [11], Higham [17], Kenney and Laub [30])

$$\gamma_k^{(1,\infty)} = ((\|X_k^{-1}\|_1 \|X_k^{-1}\|_\infty)/(\|X_k\|_1 \|X_k\|_\infty))^{1/4}, \tag{37}$$

$$\gamma_k^{(F)} = (\|X_k^{-1}\|_F / \|X_k\|_F)^{1/2}, \tag{38}$$

$$\gamma_k^{(det)} = 1/|\det X_k|^{1/n}. \tag{39}$$

The parameters (37) proposed by Higham are mostly used in practice.

Kenney and Laub [30] have proved that if $\gamma_k = \gamma_k^{(opt)}$ in each iteration (35) then $X_s = U$ where s is the number of distinct singular values of A; that is, exact convergence is obtained in $s \leq n$ iterations. Practical experience shows that even using $\gamma_k^{(1,\infty)}$, convergence is almost always obtained in ten iterations or less. Kenney and Laub [30] have shown that method (35) is faster (nonslower) than (24)

$$\|X_{k+1}(\gamma_k) - U\|_2 \leq \|X_{k+1}(1) - U\|_2$$

if and only if $(\gamma_k^{(opt)})^2 \leq \gamma_k \leq 1, \quad (k > 0)$. The parameter (38) satisfies these inequalities

$$(\gamma_k^{(opt)})^2 \leq \gamma_k^{(F)} \leq 1.$$

Kenney and Laub [30] conjecture that it is also true for $\gamma_k = \gamma_k^{(1,\infty)}$. This conjecture is confirmed by numerical experiments. Higham [17] has shown that

$$n^{-1/4}\gamma_k^{(opt)} \leq \gamma_k^{(1,\infty)} \leq n^{1/4}\gamma_k^{(opt)}.$$

The same inequalities are satisfied by $\gamma_k^{(F)}$. It can be expected that the acceleration parameters (39) have the worst properties because they are not asymptotically optimal (see Kenney and Laub [30]).

Acceleration is also applied to the method (25) for A of full rank (see Gander [11]). We compute

$$X_{k+1} = \frac{1}{2}X_k\left(\gamma_k I + \frac{1}{\gamma_k}(X_k^H X_k)^{-1}\right) \tag{40}$$

with γ_k chosen analogously as in (36)–(39)

$$\gamma_k^{(opt)} = (\|Y_k^{-1}\|_2/\|Y_k\|_2)^{1/4}, \tag{41}$$

$$\gamma_k^{(1,\infty)} = (\|Y_k^{-1}\|_1/\|Y_k\|_1)^{1/4}, \tag{42}$$

$$\gamma_k^{(F)} = (\|Y_k^{-1}\|_F/\|Y_k\|_F)^{1/4}, \tag{43}$$

$$\gamma_k^{(det)} = 1/|\det Y_k|^{1/(2n)}, \tag{44}$$

where $Y_k = X_k^H X_k$. The parameters (41) and (42) were introduced in Gander [11]. Motivated by the formulae (38) and (39) we also consider (43) and (44). The parameters (41) and (42) satisfy the following inequalities

$$n^{-1/8}\gamma_k^{(opt)} \leq \gamma_k^{(1,\infty)} \leq n^{1/8}\gamma_k^{(opt)}.$$

We now describe another modification of Higham's method. Higham and Schreiber [27] have formulated a *hybrid algorithm* which adaptively switches from the matrix inversion in (35) to a matrix multiplication based on Schulz iterations. It depends on whether the condition

$$\|X_k^H X_k - I\|_1 \leq \theta < 1, \tag{45}$$

where θ is a given accuracy, is satisfied. An idea of using the inner Schultz iteration to compute X_k^{-1} belongs to Schreiber and Parlett [39]. Namely, they have proposed to use the inner Schultz iteration

$$Z_{j+1} = Z_j + (I - Z_j X_k)Z_j, \quad Z_0 = X_{k-1}^{-1},$$

to compute X_k^{-1} on all iterations after the first. The sequence X_k constructed in (35) is converging quadratically, and X_{k-1}^{-1} is an increasingly good approximation to X_k^{-1}. Therefore it is sufficient to perform only one

Schulz iteration with $Z_0 = X_k^H$ beacuse X_k is convergent to unitary matrix. This leads to the following method (see Higham and Schreiber [39])

$$X_{k+1} = \frac{1}{2}(X_k + (Z_0^H + Z_0^H(I - Z_0X_k)^H)) = X_k(I + \frac{1}{2}(I - X_k^H X_k)).$$

It is precisely the quadratically convergent method of Björck and Bowie [5] for computing the unitary factor U (see (22)). The hybrid method of Higham and Schreiber needs computing the norm (see (45))

$$\|X_k^H X_k - I\|_1. \tag{46}$$

For this purpose we can use the matrix norm estimator CONEST which computes a lower bound for $\|C\|_1$ by sampling several matrix-vector products Cx and $C^H x$ (see Higham [20]). Therefore we can compute (46) without forming $X^H X$. Let μ be an estimation of (46). A suitable way to use estimate μ in (45) is to test whether it is less than $\beta\theta$

$$\mu \leq \beta\theta < 1, \tag{47}$$

where $1/3 \leq \beta < 1$. If so, $X^H X - I$ is formed and its norm is taken, otherwise we take μ as the left side of the inequality (45). In the hybrid method if (45) is satisfied then (22) is performed, otherwise we compute (35). The iterative process is stoped when the norm (46) is smaller or equal to δ. This criterion is different from the condition (17) applied in other methods. The hybrid algorithm is as follows (A – real nonsingular matrix of order n).

Hybrid algorithm

```
procedure HYBRID (A,U,H; n; δ,β,θ; k)
var
  μ : real;
  H1, R, Xk, Xk+1 : array[1..Nmax,1..Nmax];
  switched:boolean;
begin
  k := −1;
  μ := 1;
  switched:=false;
  X0 := A;
  repeat
    k := k + 1;
    if switched then
       begin
          R := I − XkT Xk; μ := ||R||1;
          evaluate (22)
       end
    else
      begin
         μ := CONEST(I − XkT Xk);
         if μ > βθ then evaluate (35)
         else
```

```
            begin
               R := I − X_k^T X_k; μ := ||R||_1;
               if μ > θ then evaluate (35)
                  else
                    begin
                       switched:=true;
                       evaluate (22)
                    end
            end
      end;
  until μ ≤ δ;
  U := X_{k+1};
  H_1 := U^T A;
  H := (H_1 + H_1^T)/2
end;{HYBRID}
```

Remark. We recall that the procedure CONEST does not use directly the matrix $I - X_k^T X_k$ because one computes only vectors of the form $(I - X_k^T X_k)y$. Therefore the procedure CONEST needs only the matrix X_k.

The hybrid algorithm will be more efficient than Higham's method if matrix multiplication can be done at twice the rate of matrix inversion (see Higham and Schreiber [27]). The experiments done by Higham and Schreiber have shown that the hybrid algorithm is fairly insensitive to the choices of θ and β.

5. Numerical experiments. We now present numerical tests comparing the algorithms described in the two previous sections, except the method (34). Computations were done on the computer IBM PC 486 DX/ 33MHz. All programs were written in TURBO–PASCAL 7.0.

Every test was done twice

– without numerical processor (directive N–, type *real*, machine precision $\epsilon = 9.09E - 13$),

– with numerical processor (directive N+, type *extended*, machine precision $\epsilon = 1.08E - 19$).

The algorithms were tested only for real square matrices. A matrix A of an order n was computed from the formula $A = P\Sigma Q^T$, where the diagonal elements of $\Sigma = \mathrm{diag}(\sigma_j)$ are given, P and Q are random orthogonal matrices obtained by means of the procedure ORTHOG_RAND from the unit EIGENSYM (see Hardt-Olejniczak and Ziętak [16]). The procedure ORTHOG_RAND generates random orthogonal matrices using Householder transformations (see Demmel and McKenney [7], Stewart [40]). Computations were done also for Hilbert and others matrices for which numerical experiments were described in Gander [11], Higham [17], Higham and Schreiber [27]. Tested matrices were constructed in the same precision as computations were done.

Inverses and determinants of matrices were computed by Gaussian elimination with partial pivoting. However for the methods (27), (31) and (40) the Cholesky decomposition was used instead of Gaussian elimination. Moreover, the Cholesky method was applied to compute the parameters (42) - (44).

The matrices X_{k+1} generated by the methods of Halley and Gander were not computed directly from (27) and (31), respectively. Namely, the matrix (31) can be written in the following form

$$X_{k+1} = (1/f)X_k + 2[(f-1)/f]^2 X_k\{[(f-2)/f]I + X_k^H X_k\}^{-1}.$$

Therefore we compute X_{k+1} in the following way. Let

$$\alpha := (f-2)/f; \quad W := X_k^T X_k + \alpha I,$$

$$W := \hat{L}\hat{L}^T \quad \text{(Cholesky decomposition)}, \tag{48}$$

$$L := \{f/[(f-1)\sqrt{2}]\}\hat{L}.$$

Then

$$X_{k+1} = (1/f)X_k + G,$$

where $G = (X_k L^{-T})L^{-1}$ is computed by solving appropriate linear systems with the triangular matrices L and L^T using the relation $L(L^T G^T) = X_k^T$. The Cholesky algorithm was implemented as in LINPACK's [8] subprogram SCHDC but without pivoting (for an error analysis of the Cholesky algorithm see Higham [23], Kielbasiński [31]). This way of computing X_{k+1} is less expensive and better stable.

The complete orthogonal decomposition was constructed by means of Householder method with column pivoting (see Golub and Van Loan [14, p. 236], Lawson and Hanson [33, p. 13]). The numerical rank of a matrix A, which is needed in the COD decomposition, was determined in the following way (see Lawson and Hanson [33, p. 79], compare Golub and Van Loan [14, p. 245]). After the *kth* stage of the Householder method with column pivoting let A be transformed to the form

$$B = P^T AQ = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}, \tag{49}$$

where R_{11} is an upper triangular matrix of order k with every diagonal element greater than $\epsilon\|A\|_\infty$. If the norm of R_{22} is small enough then we regard the order of R_{11} as the numerical rank of A. For this purpose we use the following criterion. If the next Householder transformation with pivoting gives the $(k+1)th$ diagonal element $\hat{r}_{k+1,k+1}$ satisfying

$$|\hat{r}_{k+1,k+1}| \leq \epsilon\|A\|_\infty$$

then the numerical rank of A is equal to k and we take $R_{22} = 0$. Otherwise we continue Householder transformations. The COD decomposition (16) of

A is obtained from (49) by additional Householder transformations applied to B^T. For comparing the numerical rank of a matrix was determined also by means of the SVD and Gaussian elimination with partial pivoting. Namely, the numerical rank of A is equal to the largest index k such that

$$\sigma_k(A) \geq n\epsilon\sigma_1(A)$$

or to the largest index of pivots in Gaussian elimination which absolute value is larger than $\epsilon\|A\|_F$.

In the methods of Gander, Halley and Higham the iterative process was continued until the condition (17), where $\delta = 10\epsilon$, was satisfied. In the hybrid algorithm we have used the criterion $\|X_k^T X_k - I\|_1 \leq \delta$. The estimation μ of (46) was obtained by means of a procedure based on the FORTRAN subprogram CONEST published by Higham [20]. We have taken $\theta = 0.6$ and $\beta = 0.75$ in the hybrid method (see (45), (47)). In every test performed for Halley's and Gander's methods an initial scaling of A was done. For this purpose we have applied the parameter (33). Tests were repeated also without initial scaling in Halley's method. If $\|A\| >> 1$ then the initial scaling improves the condition of inverted matrices. This can have an influence on the accuracy of results (see Example 6 below). However, it may increase the number of iterations (see Gander [11]).

In Gander's method the parameter *rcon* was computed by means of the singular values obtained from the SVD. We did not use any estimation of the singular values to compute *rcon*. We have applied a procedure based on the ALGOL procedure SVD published in the book by Wilkinson and Reinsch [42] to compute the SVD of a matrix.

The acceleration parameters were calculated immediately from (36)–(39) in Higham's method (35). For this purpose we have applied the SVD and Gaussian elimination. Higham [17] has suggested that if

$$\|X_k - X_{k-1}\|_1 \leq 0.01 \tag{50}$$

then the unaccelerated iteration should be performed in the next steps. Therefore we have done computation with such a version of Higham's method. We have performed also computation using the acceleration in every iteration but time of computations was bigger than with the criterion (50). Therefore we do not present these experiments here. However, in the hybrid method the acceleration parameters were used in every Higham's iteration (35). The criterion (50) was also applied to the method (40) with the parameters (41)–(44).

Our experiments confirm the conclusion of Higham (see Higham [17]) that the acceleration is very effective. The number of iterations, performed in Higham's method with acceleration parameters, was smaller than in other iterative methods. Unfortunately, computing the accelerетion parameters

costs. Therefore they should be used only in the initial iterations as it was suggested by Higham. We recall that the orders of Halley's and Higham's methods are 3 and 2, respectively. In spite of it Halley's method is not faster than Higham's method with acceleration.

We have used Higham's methods (35) and (40) also to singular or nearly singular matrices. Computations were done completely in almost every test because pivots in Gaussian elimination were different from zero. However, the accuracy was not satisfactory, especially when we have used the acceleration parameters (39) and (44). The similar remarks concern also the hybrid algorithm and the methods of Björck-Bowie and Halley (see Gander [11]). We recall that theoretically these methods work only for matrices of full rank. Due to rounding errors the singular values of X_k do not remain zero during the iterations. Therefore the methods are performed completely and the sequences X_k are convergent. For very ill conditioned matrices W (see (48)) the Cholesky algorithm was sometimes stopped because of a square root of a negative real number. In such a case we have added $\epsilon\|W\|_1$ to an argument of a square root. This modification was necessary in Higham's method (40) used to Hilbert matrix of order 20 and to the gallery(5) from MATLAB [35] (see Examples 5 and 6 given below).

In the examples given below we use the following notation:

- *COD* – the method using the COD and the Higham method (35) applied to the block R (see (14), (15), (16)),
- *SVD* – the method applying the SVD,
- *GAND* – the method of Gander,
- *HALL*1 – the method (27) with initial scaling (see (30), (33)),
- *HALL*2 – the method (27) without initial scaling,
- *HIGH*1 – the method (35),
- *HIGH*2 – the method (40),
- *HYBR* – the hybrid algorithm with the acceleration parameters (37).

In Higham's methods (35) and (40) the unaccelerated iterations were performed when the condition (50) was satisfied.

In the tables below the values of

$$\|A - \tilde{U}\tilde{H}\|_F/\|A\|_F,$$

where $\tilde{U}$ and $\tilde{H}$ denote the computed orthogonal and symmetric factors from the polar decomposition of A, are given for $n = 5$ and $n = 20$. The parameter *rcon* used in Gander's method, the norm $\|A\|_F$ and the numerical rank of A, obtained by means of the COD, are also presented. Moreover, time of computations done for receiving $\tilde{U}$ and $\tilde{H}$ altogether and the numbers of the iterations performed during computing $\tilde{U}$ are given. Time of performing Higham's methods is calculated together with time of computing the acceleration parameters. Time of computing the estimation of μ was included

into the time of the hybrid method. We do not take into consideration time of computing the parameter *rcon* in Gander's method. The time is given in ticks of a computer clock. One tick is equal to 1/18 second.

At almost every test the computed orthogonal factors $\tilde{U}$ were orthogonal within machine accuracy – the error $\|\tilde{U}^T\tilde{U} - I\|_F$ was small. Therefore we do not give these errors in the tables below. The error $\|\tilde{U}^T\tilde{U} - I\|_F$, computed for all matrices A considered in the examples below, was smaller than $6.09E-18$ for computations with the numerical processor and smaller than $2.82E-10$ without the numerical processor. The worst cases were for the method applying the SVD.

The positive semi-definitness of the computed Hermitian factors was tested by attempting to compute a Cholesky decomposition of $\tilde{H}$. In this test all computed symmetric factors $\tilde{H}$ in Examples 1, 2, 3 and 4 were positive semi-definite, i.e. Cholesky's tests were performed completely. However, in Example 5 for Hilbert matrix of order 20 Cholesky's tests were broken off for all methods. The similar situation was in Example 6 for computations without the numerical processor—Cholesky decomposition of $\tilde{H}$ was obtained only for the method *HIGH*2 without accerelation. However, the following methods

COD, *GAND*, *HIGH*1 with $\gamma_k^{(opt)}, \gamma_k^{(F)}$ and without acceleration

have given positive semi-definite $\tilde{H}$ in Example 6 when computations were preformed with the numerical processor.

The advantage of the iterative methods was well seen for matrices A close to orthogonal matrices (see Example 2 below). In this case only a few iterations were needed.

In the tables below the results obtained without and with the numerical processor are given in the left and right columns, respectively.

EXAMPLE 1. A – a random matrix of order n with given singular values $\sigma_i = i$ for $i = 1, \ldots, n$.

Case 1: $n = 5$,

$\|A\|_F = 7.41E+00$, $rcon = 2.0E-01$, numerical rank of A is 5.

Case 2: $n = 20$,

$\|A\|_F = 5.35E+01$, $rcon = 5.0E-02$, numerical rank of A is 20.

	n=5						n=20					
	time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$		time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$	
COD	2	1	—	—	2.61E-12	2.07E-19	78	48	—	—	5.72E-12	2.37E-19
SVD	2	7	—	—	9.07E-12	6.09E-19	57	62	—	—	1.14E-10	1.79E-18
GAND	2	1	5	5	4.21E-12	1.75E-19	50	27	5	6	4.91E-12	3.22E-19
HALL1	2	1	7	7	1.90E-12	2.81E-19	83	39	9	9	4.89E-12	3.13E-19
HALL2	2	1	5	6	3.24E-12	2.41E-19	66	31	7	7	4.20E-12	3.10E-19

	n=5						n=20					
	time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$		time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$	
HIGH1												
$\gamma_k^{(opt)}$	6	37	6	6	2.74E-12	9.85E-20	110	171	6	7	4.98E-12	2.92E-19
$\gamma_k^{(1,\infty)}$	2	1	6	6	2.70E-12	9.99E-20	60	41	7	8	4.68E-12	2.57E-19
$\gamma_k^{(F)}$	2	1	6	7	1.84E-12	1.40E-19	61	41	7	8	5.37E-12	2.95E-19
$\gamma_k^{(det)}$	1	1	6	7	3.17E-12	1.68E-19	81	51	8	8	5.37E-12	3.24E-19
no accel.	1	1	7	8	2.93E-12	8.57E-20	72	50	9	10	5.26E-12	2.76E-19
HIGH2												
$\gamma_k^{(opt)}$	6	36	6	6	1.99E-12	1.36E-19	157	175	6	7	4.41E-12	2.67E-19
$\gamma_k^{(1,\infty)}$	2	1	6	6	2.72E-12	1.69E-19	98	40	6	7	4.37E-12	2.94E-19
$\gamma_k^{(F)}$	2	1	6	6	2.49E-12	1.17E-19	116	46	7	8	4.76E-12	3.15E-19
$\gamma_k^{(det)}$	2	1	6	7	2.15E-12	1.55E-19	143	55	8	8	6.75E-12	4.68E-19
no accel.	3	1	7	8	2.40E-12	1.22E-19	144	55	9	10	4.09E-12	2.43E-19
HYBR	1	1	6	6	1.82E-12	7.46E-20	70	36	7	7	3.63E-12	2.22E-19

EXAMPLE 2. A – a random matrix of order n with the singular values σ_i $(i = 1, \cdots, n)$ being the increasing arithmetic sequence such that $\sigma_1 = 1$, $\sigma_n = \text{cond}$ where cond $= 1.0001$ is the given condition number of A.

Case 1: $n = 5$,

$\|A\|_F = 2.23E + 00$, $rcon = 9.99E - 01$, numerical rank of A is 5.

Case 2: $n = 20$,

$\|A\|_F = 4.47E + 00$, $rcon = 9.99E - 01$, numerical rank of A is 20.

	n=5						n=20					
	time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$		time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$	
COD	1	1	—	—	2.87E-12	1.63E-19	45	34	—	—	4.52E-12	2.61E-19
SVD	2	7	—	—	3.26E-11	6.14E-19	54	52	—	—	7.97E-11	1.57E-18
GAND	1	0	4	4	3.55E-12	2.67E-19	41	20	4	4	4.46E-12	3.25E-19
HALL1	1	1	5	6	4.49E-12	2.98E-19	59	32	6	7	4.40E-12	3.44E-19
HALL2	1	1	2	3	5.50E-12	2.82E-19	27	15	2	3	3.83E-12	2.76E-19
HIGH1												
$\gamma_k^{(opt)}$	2	12	3	3	2.11E-12	1.13E-19	46	52	3	3	4.70E-12	2.62E-19
$\gamma_k^{(1,\infty)}$	0	0	3	3	1.47E-12	1.29E-19	30	18	3	3	4.44E-12	2.84E-19
$\gamma_k^{(F)}$	1	1	3	3	2.00E-12	1.16E-19	30	18	3	3	4.75E-12	2.64E-19
$\gamma_k^{(det)}$	1	1	3	3	2.11E-12	1.20E-19	33	19	3	3	4.60E-12	2.70E-19
no accel.	1	0	3	4	1.55E-12	1.10E-19	29	23	3	4	4.44E-12	2.67E-19
HIGH2												
$\gamma_k^{(opt)}$	2	9	3	3	1.88E-12	1.41E-19	69	53	3	3	2.80E-12	1.70E-19
$\gamma_k^{(1,\infty)}$	1	1	3	3	2.65E-12	1.35E-19	53	20	3	3	3.02E-12	1.94E-19
$\gamma_k^{(F)}$	1	0	3	3	1.78E-12	1.54E-19	54	19	3	3	2.82E-12	2.02E-19
$\gamma_k^{(det)}$	1	1	3	3	1.66E-12	8.95E-20	55	21	3	3	2.82E-12	2.00E-19
no accel.	1	1	3	4	2.46E-12	1.62E-19	53	24	3	4	3.17E-12	1.93E-19
HYBR	1	0	2	3	1.59E-12	1.26E-19	36	17	3	3	2.72E-12	1.96E-19

EXAMPLE 3. A – a random matrix with given singular values $\sigma_i = 2^i$, where $i = 1, \ldots, n$.

Case 1: $n = 5$,

$\|A\|_F = 3.69E + 01$, $rcon = 6.25E - 02$, numerical rank of A is 5.

Case 2: $n = 20$,

$\|A\|_F = 1.21E+06$, $rcon = 1.90E-06$, numerical rank of A is 20.

	n=5						n=20					
	time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$		time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$	
COD	1	2	—	—	2.89E-12	1.21E-19	85	61	—	—	6.08E-12	2.85E-19
SVD	1	7	—	—	2.39E-11	9.18E-19	44	45	—	—	2.78E-11	9.27E-19
GAND	1	0	5	6	4.16E-12	3.63E-19	82	44	9	10	1.15E-11	8.81E-19
HALL1	3	1	8	8	5.03E-12	2.27E-19	157	75	18	18	5.54E-12	3.99E-19
HALL2	2	1	7	7	3.49E-12	3.08E-19	139	71	16	17	6.45E-09	5.65E-16
HIGH1												
$\gamma_k^{(opt)}$	6	36	5	5	1.22E-12	6.46E-20	144	248	8	9	4.66E-12	3.05E-19
$\gamma_k^{(1,\infty)}$	2	1	6	7	1.40E-12	1.65E-19	67	46	8	9	5.82E-12	2.67E-19
$\gamma_k^{(F)}$	1	1	6	7	1.37E-12	1.05E-19	68	47	8	9	5.76E-12	3.18E-19
$\gamma_k^{(det)}$	2	2	6	7	1.87E-12	8.35E-20	81	55	8	9	4.81E-12	2.90E-19
no accel.	3	1	10	11	2.15E-12	1.36E-19	186	124	25	26	4.77E-12	2.12E-19
HIGH2												
$\gamma_k^{(opt)}$	6	39	5	5	3.00E-12	1.46E-19	209	252	8	9	1.59E-07	7.54E-15
$\gamma_k^{(1,\infty)}$	2	1	6	7	2.60E-12	1.60E-19	128	50	8	9	1.71E-07	8.17E-15
$\gamma_k^{(F)}$	2	1	6	7	2.96E-12	2.06E-19	132	52	8	9	1.61E-07	7.87E-15
$\gamma_k^{(det)}$	2	2	6	7	3.17E-12	1.94E-19	143	60	8	9	1.56E-07	8.06E-15
no accel.	3	1	10	11	1.74E-12	1.81E-19	384	136	25	26	2.99E-09	2.60E-16
HYBR	1	1	6	7	1.07E-12	7.74E-20	85	41	9	8	4.62E-12	2.03E-19

EXAMPLE 4. A – a random matrix with given singular values $\sigma_i = i^4$, where $i = 1, \ldots, n$.

Case 1: $n = 5$,

$\|A\|_F = 6.80E+02$, $rcon = 1.60E-03$, numerical rank of A is 5.

Case 2: $n = 20$,

$\|A\|_F = 2.65E+05$, $rcon = 6.25E-06$, numerical rank of A is 20.

	n=5						n=20					
	time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$		time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$	
COD	2	1	—	—	2.22E-12	1.20E-19	85	61	—	—	7.00E-12	2.42E-19
SVD	2	8	—	—	1.88E-11	4.77E-19	42	45	—	—	7.77E-11	1.06E-18
GAND	2	1	7	7	6.67E-12	4.32E-19	83	44	9	10	9.29E-12	6.75E-19
HALL1	3	2	11	11	2.61E-12	2.60E-19	150	72	17	17	5.34E-12	3.84E-19
HALL2	3	1	10	10	5.27E-12	3.09E-19	132	63	15	15	1.47E-09	3.30E-17
HIGH1												
$\gamma_k^{(opt)}$	7	45	6	6	2.28E-12	1.12E-19	146	243	8	8	5.21E-12	2.71E-19
$\gamma_k^{(1,\infty)}$	1	1	7	8	2.36E-12	1.83E-19	68	46	8	9	5.65E-12	2.71E-19
$\gamma_k^{(F)}$	2	1	7	8	3.39E-12	1.45E-19	69	46	8	9	5.80E-12	3.37E-19
$\gamma_k^{(det)}$	3	3	8	8	4.01E-12	1.40E-19	120	80	12	13	8.86E-11	9.85E-18
no accel.	3	3	14	15	2.15E-12	1.13E-19	165	109	22	23	4.27E-12	2.15E-19
HIGH2												
$\gamma_k^{(opt)}$	10	45	6	6	3.38E-11	1.14E-17	210	248	8	8	2.52E-08	1.98E-15
$\gamma_k^{(1,\infty)}$	3	2	7	8	4.40E-11	1.19E-17	128	45	8	8	2.55E-08	1.90E-15
$\gamma_k^{(F)}$	2	1	7	8	4.44E-11	9.35E-18	131	46	8	8	2.87E-08	2.06E-15
$\gamma_k^{(det)}$	3	1	8	8	2.30E-10	4.18E-17	213	87	12	13	2.08E-06	1.37E-13
no accel.	5	2	14	15	2.25E-12	1.85E-19	339	120	22	23	1.93E-10	4.99E-17
HYBR	2	1	7	8	1.71E-12	1.55E-19	84	41	9	8	3.93E-12	2.45E-19

EXAMPLE 5. A – Hilbert matrix.

Case 1: $n = 5$,

$\|A\|_F = 1.58E + 00$, $rcon = 2.09E - 06$, numerical rank of A is 5.

Case 2: $n = 20$,

$\|A\|_F = 1.96E + 00$, $rcon = 0$, numerical rank of A is 15.

	n=5						n=20					
	time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$		time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$	
COD	2	1	—	—	3.49E-12	1.29E-19	35	34	—	—	9.63E-12	3.51E-19
SVD	1	7	—	—	7.05E-12	3.50E-19	27	42	—	—	5.17E-12	1.15E-18
GAND	3	1	9	10	6.20E-12	4.46E-19	150	100	17	24	1.45E-11	1.26E-18
HALL1	5	2	17	17	3.18E-12	2.32E-19	298	202	35	50	6.63E-12	4.45E-19
HALL2	4	2	15	16	3.10E-12	2.39E-19	281	194	33	48	5.41E-12	6.52E-19
HIGH1												
$\gamma_k^{(opt)}$	8	47	6	6	1.22E-12	5.70E-20	**	320	**	10	*****	5.63E-16
$\gamma_k^{(1,\infty)}$	2	1	8	8	9.09E-13	8.23E-20	83	52	10	10	6.99E-10	1.87E-16
$\gamma_k^{(F)}$	2	1	8	9	7.47E-13	7.82E-20	86	52	10	10	6.28E-10	3.01E-16
$\gamma_k^{(det)}$	2	2	8	9	1.25E-12	1.70E-19	111	81	11	13	9.58E-12	9.33E-15
no accel.	4	3	23	24	1.01E-09	1.45E-16	388	353	53	76	4.48E-01	1.34E-01
HIGH2												
$\gamma_k^{(opt)}$	8	44	7	6	8.39E-08	4.23E-15	**	1370	**	37	*****	3.89E-09
$\gamma_k^{(1,\infty)}$	3	1	8	8	9.51E-08	3.75E-15	143	56	9	10	4.70E-05	4.63E-07
$\gamma_k^{(F)}$	3	1	8	8	5.81E-08	3.09E-15	148	62	9	11	3.83E-05	3.17E-07
$\gamma_k^{(det)}$	3	1	8	9	3.27E-07	9.57E-15	285	94	16	14	1.04E-06	7.12E-09
no accel.	7	3	23	24	1.68E-03	1.23E-11	356	188	23	36	1.69E-01	5.23E-02
HYBR	2	1	7	8	9.09E-13	1.14E-19	95	51	10	10	6.99E-10	1.87E-16

***** - computations were broken off because it was impossible to compute $\gamma_k^{(opt)}$ since the computed smallest singular value was equal to zero.

Hilbert matrix A is Hermitian and positive definite. Therefore its Hermitian factor H is equal to A and the orthogonal factor U is equal to I. Unfortunately the numerical rank of A is only 15. Therefore numerically computed orthogonal factor can be different from I because of the nonuniquennes. Although the computed $\tilde{U}$ is orthogonal within the machine accuracy, $\tilde{U}$ is different from I. Here are additional results showing it.

	n=5				n=20			
	$\|H-A\|_F$		$\|U-I\|_F$		$\|H-A\|_F$		$\|U-I\|_F$	
COD	3.20E-12	1.85E-19	1.55E-11	2.29E-17	1.18E-11	5.31E-19	4.45E+00	2.99E+00
SVD	6.67E-12	3.91E-19	4.01E-10	1.31E-18	9.05E-12	1.57E-18	3.95E+00	2.85E+00
GAND	9.76E-12	6.46E-19	2.92E-10	1.21E-17	2.73E-11	2.48E-18	3.68E+00	2.58E+00
HALL1	3.36E-12	2.76E-19	1.14E-09	2.05E-17	1.00E-11	7.96E-19	3.86E+00	2.90E+00
HALL2	3.58E-12	2.35E-19	2.62E-10	5.20E-18	7.44E-12	1.28E-18	3.64E+00	2.82E+00
HIGH1								
$\gamma_k^{(opt)}$	1.22E-12	8.43E-20	2.55E-10	8.70E-18	*****	1.10E-15	*****	3.41E+00
$\gamma_k^{(1,\infty)}$	1.20E-12	1.21E-19	2.54E-10	8.79E-18	1.37E-09	3.68E-16	4.49E+00	3.41E+00
$\gamma_k^{(F)}$	1.22E-12	1.32E-19	2.55E-10	8.95E-18	1.23E-09	5.93E-16	4.49E+00	3.41E+00
$\gamma_k^{(det)}$	1.96E-12	2.93E-19	2.57E-10	8.89E-18	1.75E-11	1.83E-14	4.49E+00	3.41E+00
no accel.	1.22E-09	1.77E-16	1.82E-09	2.69E-16	9.98E-01	1.99E-01	4.74E+00	3.43E+00

	n=5				n=20			
	$\|H-A\|_F$		$\|U-I\|_F$		$\|H-A\|_F$		$\|U-I\|_F$	
HIGH2								
$\gamma_k^{(opt)}$	1.19E-07	6.49E-15	2.19E-04	8.66E-12	*****	7.67E-09	*****	2.87E+00
$\gamma_k^{(1,\infty)}$	1.36E-07	5.50E-15	2.19E-04	8.66E-12	9.28E-05	9.13E-07	4.37E+00	2.87E+00
$\gamma_k^{(F)}$	8.99E-08	4.85E-15	2.19E-04	8.66E-12	7.56E-05	6.26E-07	4.36E+00	2.86E+00
$\gamma_k^{(det)}$	4.56E-07	1.47E-14	2.19E-04	8.66E-12	2.08E-06	1.40E-08	4.21E+00	2.87E+00
no accel.	2.03E-03	1.51E-11	3.00E-03	2.41E-11	3.34E-01	1.02E-01	4.44E+00	2.91E+00
HYBR	1.20E-12	9.11E-20	2.54E-10	8.79E-18	1.37E-09	3.68E-16	4.49E+00	3.41E+00

***** - computations were broken off because it was impossible to compute $\gamma_k^{(opt)}$ since the computed smallest singular value was equal to zero.

EXAMPLE 6. A – the gallery(5) from MATLAB (see Moler, Little and Bangert [35])

$$A = \begin{bmatrix} -9 & 11 & -21 & 63 & -252 \\ 70 & -69 & 141 & -421 & 1684 \\ 575 & 575 & -1149 & 3451 & -13801 \\ 3891 & -3891 & 7782 & -23345 & 93365 \\ 1024 & -1024 & 2048 & -6144 & 24572 \end{bmatrix},$$

$\|A\|_F = 1.01E+05$, $rcon = 0$, numerical rank of A is 4.

	time		iter		$\frac{\|A-UH\|_F}{\|A\|_F}$	
COD	2	1	—	—	3.98E-12	2.34E-19
SVD	1	8	—	—	6.46E-12	6.08E-19
GAND	4	3	17	23	1.46E-11	3.00E-19
HALL1	9	8	34	48	4.36E-12	7.21E-19
HALL2	8	4	22	37	5.62E-10	3.47E-17
HIGH1						
$\gamma_k^{(opt)}$	8	63	7	7	1.46E-11	3.19E-19
$\gamma_k^{(1,\infty)}$	2	1	9	10	3.70E-12	1.27E-19
$\gamma_k^{(F)}$	2	1	9	10	4.58E-12	1.25E-19
$\gamma_k^{(det)}$	2	2	10	12	3.38E-09	1.98E-10
no accel.	6	8	33	58	5.54E-09	2.79E-09
HIGH2						
$\gamma_k^{(opt)}$	23	241	17	28	3.41E-07	8.42E-10
$\gamma_k^{(1,\infty)}$	3	2	9	10	1.47E-06	5.05E-08
$\gamma_k^{(F)}$	3	1	9	10	1.04E-06	5.04E-08
$\gamma_k^{(det)}$	4	2	9	11	7.48E-06	8.37E-08
no accel.	7	3	22	23	3.61E-08	3.68E-08
HYBR	2	1	9	9	3.65E-12	1.90E-19

6. Comments. The authors do not know a complete error analysis of the algorithms for computing the polar decomposition. Some results concerning this important problem are given in Higham [17]. Higham has proposed a test which provides an inexpensive means of monitoring the stability of a method for computing the polar decomposition.

In connection with the relation (32) it is interesting to test if the criterion (17) is appropriate for all methods. May be we should apply another criterion using an order of a method. It could decrease the number of iterations.

Acknowledgement. The authors thank very much Professor A. Kiełbasiński (Warsaw) for drawing their attention to the relation (32), for some advises concerning additional numerical tests and for remarks which have improved the presentation.

References

[1] L. Autonne, Sur les groupes linéaires, réels et orthogonaux, *Bull. Soc. Math. France* 30 (1902), 121-134.

[2] I. Y. Bar-Itzhack and K. A. Fegley, Orthogonalization techniques of a direction cosine matrix, *IEEE Trans. Aeorospace Electron. Systems* 5 (1969), 798-804.

[3] A. Barrlund, Perturbation bounds on the polar decomposition, *BIT* 30 (1989), 101-113.

[4] R. Bhatia and F. Kittaneh, Approximation by positive operators, *Linear Algebra Appl.* 161 (1992), 1-9.

[5] Å. Björck and C. Bowie, An iterative algorithm for computing the best estimate of an orthogonal matrix, *SIAM J. Numer. Anal.* 8 (1971), 358 - 364.

[6] R. Byers, Solving the algebraic Riccati equation with the matrix sign function, *Linear Algebra Appl.* 85 (1987), 267-279.

[7] J. Demmel and A. McKenney, A test matrix generation suite, *LAPACK Working Note* 9, Courant Institute, New York 1989.

[8] J. J. Dongarra, J. R. Bunch, C. B. Moler and G. W. Stewart, *LINPACK Users's Guide*, SIAM, Philadelphia 1979.

[9] K. Fan and A. J. Hoffman, Some matrix inequalities in the space of matrices, *Proc. Amer. Math. Soc.* 6 (1955), 111-116.

[10] W. Gander, On Halley's iteration method, *Amer. Math. Monthly* 92 (1985), 131-134.

[11] W. Gander, Algorithms for polar decomposition, *SIAM J. Sci. Statist. Comput.* 11 (1990), 1102-1115.

[12] F. R. Gantmacher, *Theory of Matrices*, vol. 1, Chelsea, New York 1959.

[13] P. E. Gill, W. Murray and M. H. Wright, *Practical Optimization*, Academic Press, London 1981.

[14] G. H. Golub and Ch. F. Van Loan, *Matrix Computation*, John Hopkins University Press, Maryland 1989.

[15] P. R. Halmos, Positive approximants of operators, *Indiana Univ. Math. J.* 21 (1972), 951-960.

[16] G. Hardt-Olejniczak and K. Ziętak, Moduł AL-Algebra liniowa, in: *Elementy informatyki. Pakiet oprogramowania edukacyjnego*, M. M. Sysło, Red., Inst. Inf. Uniwersytetu Wrocławskiego i OFEK-Poznań 1992.

[17] N. J. Higham, Computing the polar decomposition – with applications, *SIAM J. Sci. Stat. Comput.* 7 (1986), 1160-1173.

[18] N. J. Higham, Newton's method for the matrix square root, *Math. Comput.* 46 (1986), 537-549.

[19] N. J. Higham, Computing real square roots of a real matrix, *Linear Algebra Appl.* 88/89 (1987), 405-430.

[20] N. J. Higham, Fortran codes for estimating the one-norm of real or complex matrix, with applications to condition estimation, *ACM Trans. Math. Software* 14 (1988), 381-396.

[21] N. J. Higham, Computing a nearest symmetric positive semidefinite matrix, *Linear Algebra Appl.* 103 (1988), 103-118.

[22] N. J. Higham, Matrix nearness problem and applications, in: *Applications of Matrix Theory*, M. J. C. Gover and S. Barnett, Eds, Oxford Univ. Press, Oxford 1989, pp. 1-27.

[23] N. J. Higham, Analysis of the Cholesky decomposition of a semi-definite matrix, in: *Reliable Numerical Computation*, M. G. Cox and S. J. Hammarling, Eds, Oxford University Press, Oxford 1990, pp. 161-185.

[24] N. J. Higham, The matrix sign decomposition and its relation to the polar decomposition, *Numer. Anal. Report* 225, Dept. of Math. Univ. of Manchester, Manchester 1993, to appear in *Linear Algebra Appl.* 212/213.

[25] N. J. Higham and P. Papadimitriou, A new parallel algorithm for computing the singular value decomposition, in: *Proc. The Fifth SIAM Conf. on Appl. Lin. Algebra*, J. G. Lewis, Ed., SIAM, Philadelphia 1994, pp. 80-84.

[26] N. J. Higham and P. Papadimitriou, A parallel algorithm for computing the polar decomposition, *Parallel Computing* 20 (1994), 1161-1173.

[27] N. J. Higham, and R. S. Schreiber, Fast polar decomposition of an arbitrary matrix *SIAM J. Sci. Stat. Comput.* 11 (1990), 648-655.

[28] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge 1985.

[29] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge 1991.

[30] Ch. Kenney and A. J. Laub, On scaling Newton's method for polar decomposition and the matrix sign function, *SIAM J. Matrix Anal. Appl.* 13 (1992), 688-706.

[31] A. Kiełbasiński, A note on rouding-error analysis of Cholesky factorization, *Linear Algebra Appl.* 88/89 (1987), 487-494.

[32] A. Kiełbasiński and H. Schwetlick, *Numerische lineare Algebra*, VEB Deutscher Verlag der Wissenschaften, Berlin 1988.

[33] Ch. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice Hall, New York 1974.

[34] R. Mathias, Perturbation bounds for the polar decomposition, *SIAM J. Matrix Anal. Appl.* 14 (1993), 588-597.

[35] C. Moler, J. Little and S. Bangert, *PRO-MATLAB User's Guide*, The MathWorks Inc., Sherborn 1987.

[36] P. Pandey, Ch. Kenney and A. J. Laub, A parallel algorithm for the matrix sign function, *Int. J. High Speed Computing* 2 (1990), 181-191.

[37] C. R. Rao, Matrix approximation and reduction of dimensionality in multivariate statistical analysis, in: *Multivariate Analysis* V, *Proc. of the Fifth Intern. Symp. on Multiv. Anal.*, P. R. Krishnaiah, Ed., North - Holland, Amsterdam 1980.

[38] J. Roberts, Linear model reduction and solution of the Riccati equation by use the sign function, *Internat. J. Control* 32 (1980), 677-687.

[39] R. S. Schreiber and B. N. Parlett, Block reflectors: Theory and computation, *SIAM J. Numer. Anal.* 25 (1988), 189-205.

[40] G. W. Stewart, On efficient generation of random orthogonal matrices with an application to condition estimators, *SIAM J. Numer. Anal.* 17 (1980), 403-409.

[41] G. A. Watson, Solving generalizations of orthogonal Procrustes problem, in: *World Scientific Series in Applicable Analysis* 2, *Contributions in Numerical Mathematics*, R. P. Agarwal, Ed., World Scientific, Singapore 1993, pp. 413-426.

[42] J. H. Wilkinson and C. Reinsch, *Handbook for Automatic Computation. Linear Algebra*, Springer, Berlin 1971.

[43] P. Zieliński, Rozkład polarny macierzy, Inst. Inf. Uniwersytetu Wrocławskiego, Wrocław 1993 (*Master thesis*).

PAWEŁ ZIELIŃSKI
DEPARTEMENT OF MATHEMATICS
HIGHER ENGINEERING SCHOOL
UL. LUBOSZYCKA 3
45-036 OPOLE, POLAND

KRYSTYNA ZIĘTAK
INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF WROCŁAW
UL. PRZESMYCKIEGO 20
51-151 WROCŁAW, POLAND