Earthline Journal of Mathematical Sciences
E-ISSN: 2581-8147
Volume 14, Number 1, 2024, Pages 63-74
https://doi.org/10.34198/ejms.14124.063074

Rotation Matrix and Angles of Rotation in the Polar Decomposition

Stephen Ehidiamhen Uwamusi

Department of Mathematics, Faculty of Physical Sciences, University of Benin,
Benin City, Edo State, Nigeria
ORCID: https://orcid.org/0000-0002-0712-3295
e-mail: stephen.uwamusi@uniben.edu

Abstract

This paper aims at computing the rotation matrix and angles of rotations using Newton and Halley's methods in the generalized polar decomposition. The method extends the techniques of Newton's and Halley's methods for iteratively finding the zeros of polynomial equation of single variable to matrix rotation valued problems. It calculates and estimates the eigenvalues using Chevbyshev's iterative method while computing the rotation matrix. The sample problems were tested on a randomly generated matrix of order n from the family of matrix market. Numerical examples are given to demonstrate the validity of this work.

1. Introduction

The idea of computing zeros of a nonlinear equation of degree n with $f(x) = 0$, where f is continuously differentiable in the interval $[a, b]$ dates back to history. There, methods of Newton, Halley, Chevbyshev iterations and their variants were very popular in use among various researchers. Later on, there developed sporadic interests in the inventions of other higher iterative method and a combination of them to attain higher orders of convergences. Such later day development of these methods usually entailed higher computing time in space and complexities. Therefore as time went on the ideas arose of extending hitherto iterative methods for finding zeros of single variable polynomials of degree n to matrix valued problems. This birthed the idea of computing polar decomposition by these methods which gained favourable appeals in the minds of numerical analysts. For example, the matrix sign function written as sign(A) is usually

Received: September 13, 2023; Revised & Accepted: October 6, 2023; Published: October 24, 2023
2020 Mathematics Subject Classification: 65F60, 65F55.
Keywords and phrases: polar decomposition, rotation matrix, angles of rotation matrix, Halley's iteration.

calculated by the Newton, Halley and Chebyshev's iterative formulas. The matrix sign function $f(A)$ belongs to a class of matrix square root problem which provides information on chromodynamic and density matrix in a molecular system with Fermi level [4] as a few examples.

In [13] a great enumeration of viable areas in which matrix sign function has gained prominence documented to include the following among others: solving algebraic Riccati equation, Lyapunov matrix equation, generalized Bernoulli equations, separation of matrix eigenvalue problems, computing the pth roots of a matrix and a host of others in the cited references.

In [7] "Functions of Matrices: Theory and Computation", Higham gave a well-documented evidence of applicability of the matrix sign function. Bjorck in [3] also had a very excellent documentation on the matrix sign functions and polar decomposition of a matrix. In a nutshell, matrix sign function is a well-known class of polar decomposition. We work along this line in this paper to provide a new technique for computing matrix rotation using the polar decomposition.

1.1. Preliminaries

The polar decomposition forms the hall mark enthusiasm that spurs the development to a method of computing rotation matrix which is obtained free of charge with no extra work. Firstly, we will give a brief exposition of polar decomposition which will fully assist the reader to grasp with the basic tools necessary to understanding the subject matter rotation matrix.

Following [3,6,7], the interest in computing the polar decomposition of a matrix $A = U_d P$ (where $A \in R^{m*n}, m > n$ or $\mathbb{C}^{m\times n}$) was developed. The matrix $U_d \in R^{m\times n}$ is a unitary polar factor while the matrix $P \in R^{n\times n}$ is a Hermittian factor. The study of polar decomposition of a matrix is an important aspect in numerical computing and engineering practices where it is always a necessary task providing the nearness of another matrix to a known matrix. Thus, nearness to the polar decomposition is the difference between matrix A and its polar factor, that is $\|A - U_d P\|$. The polar decomposition of a matrix thus describes [3] information on nearby structured matrices. The matrix gives the nearest Hermittian positive semi-definite matrix to A. Therefore, one often recovers the SVD [2] from computing the polar decomposition of $A = U_d P$ by computing the eigen decomposition of $P \in R^{n\times n}$ such that $P = V\sum V^T$, where $\sum \in D^{n\times n}$ and, $V \in R^{n\times n}$ [6,7,8,9,10,11,12]. Therefore it follows that the polar decomposition narrows closely the

Singular value decomposition (SVD). In a polar decomposition $A = U\sum V^H$, the U and V are the left and right singular vectors respectively while $\sum$ contains the associated singular values of A (i.e., eigenvalues of $(A^T A)$).

Computing the nearness to the polar decomposition [3] involves the following steps. Firstly, compute the polar factor U by yet to be discussed Newton or Halley's iterative method. Then the matrix P is calculated by some technique in the accompany formula. The difference matrix $D = A - U_d P$ is performed. Secondly, calculate the matrix norm D denoted as $\|D\|$. Any of the matrix norms can be used such as infinite norm, the first norm or the Frobenius norm (2-norm of matrix singular values) may be suitable for this purpose. The scaling dynamically Newton's and Halley's method [1,7,13] are a class of methods which use the scaling factors such as matrix condition numbers, the determinant of a matrix or a combination of both scaling equivalent factor to speeding up calculation process in the matrix polar decomposition. This process leads to a dynamically rotated matrix which hitherto has not been considered by anybody in the available literatures.

Of particular interest in this study is the rotation matrix and the accompanying rotation angles. This can be calculated from the numerical results in the polar decomposition $A = U_d P$. The rotation component in the polar decomposition is the unitary matrix U_d. The rotation matrix is the result obtained by taking the logarithm of the component matrix U_d.

2. Derivation of a Class of Iterative Methods

A class of iterative methods for computing polar decomposition is presented. The methods were derived from the iterative methods known to converge globally and quadratically to the desired zeros of polynomials of single variable, $f(x) = 0$.

The polar decomposition is amenable to methods calculating the roots of polynomial equation of single variable. Some relevant literature in this field are [1,3,6,7,8,9,10,12].

The two commonly used methods are the Newton and Halley's iteration. Higher order method exists such as Chevbyshev's method [11]. We shall however dwell on the former two methods of Newton and Halley. They are written as

$$x_{k+1} = x_k - f^{/}(x_k)^{-1} f(x_k), \quad k = 0,1,2,\ldots \quad \text{(Newton's method)} \tag{2.1}$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f^{/}(x_k) - \frac{f(x_k) f^{//}(x_k)}{2 f^{/}(x_k)}}, \quad k = 0,1,2,\ldots \quad \text{(Halley's method)} \tag{2.2}$$

Following [11] the *p*th root formula for the equation $x^p - a = 0, p \geq 2$ with the Newton and Halley's methods are written in the forms:

$$x_{k+1} = \frac{1}{p}\left[(p-1)x_k + a x_k^{1-p}\right],\ x_0 = 1,\ p \geq 2,\ k = 0,1,2,\dots \text{ (Newton method)} \quad (2.3)$$

$$x_{k+1} = x_k\left(\frac{(p-1)x_k^p + (p+1)a}{(p+1)x_k^p + (p-1)a}\right),\ x_0 = 1,\ p \geq 2,\ k = 0,1,2,\dots \text{ (Halley's method)} \quad (2.4)$$

In equations (2.3) and (2.4), we take $p = 2$ for solving matrix square root equation $X^2 - A = O$. This technique introduced the Newton's and Halley's methods as the basic tools for implementation in the matrix square root problem. We therefore extend this approach to the case of polar decomposition and consequently the matrix rotation problem which we shall describe subsequently in the paper. For this, we review in the forms:

Newton's Method:

$$X_{k+1} = \frac{1}{2}\left(X_k + X_k^{-T}\right),\ \ X_0 = A \quad (k = 0,1,2,\dots) \quad (2.5)$$

The Halley's Method:

$$X_{k+1} = X_k\left(3X_k^2 + A\right)^{-1}\left(X_k^2 + 3A\right),\ X_0 = I \quad (k = 0,1,2,\dots) \quad (2.6)$$

It is usual to modify the Newton's and Halley's methods [5,9,11]. Such modifications are the scaled Newton's and scaled Halley's methods and are written as:

The scaled Newton's method:

$$X_{k+1} = \frac{1}{2}\left(c_k X_k + \frac{1}{c_k} X_k^{-1}\right) \quad (k = 0,1,2,\dots). \quad (2.7)$$

The modified Halley's method:

$$X_{k+1} = X_k\left(a_k I + b_k X_k^T X_k\right)\left(I + c_k X_k^T X_K\right)^{-1},\ X_0 = \frac{A}{a},\ a = \|A\|_2 \ \ (k = 0,1,2,\dots). \quad (2.8)$$

Taking for instance, $a_K = \|A\|_2, b_k = \frac{1}{\|A\|_2}, c_k = a_k + b_k - 1$, as parameters in the iterations in equations (2.7) and (2.8) respectively will yield improved iterative results.

Besides the aforementioned methods regarding equations (2.7) and (2.8), by applying the SVD on equation (2.5) we have that

$$X_{k+1} = U\Sigma_k\left(a_k I + b_k \Sigma_k^2\right)\left(I + c_k \Sigma_k^2\right)^{-1} V^T. \quad (2.9)$$

The matrices U and V are orthonormal while $\sum_k = (\sigma_1, \sigma_2, \ldots, \sigma_r, \ldots, \sigma_n)$ are the singular values corresponding to the matrix $A^T A$, where, $\sigma_1 > \sigma_2 > \ldots > \sigma_r > 0$, are ordered according to their multiplicities.

Following equation (2.9) we now describe theoretical basis surrounding the singular values associated with the iterated matrix $\sigma(X_{k+1})$. We give such review here critical for deeper understanding of the subject matter under consideration.

The estimate for $\sigma_i(X_{k+1})$ is given by the equation

$$\sigma_i(X_{k+1}) = g_k(\sigma_i(X_k)), \tag{2.10}$$

where,

$$g_k(x) = x\left(\frac{a_k + b_k}{1 + c_k x^2}\right); \quad (0 < g_k(x) < 1). \tag{2.11}$$

In [1,3,7,9,10] the global minimizers of $g_k(x)$ are well known. Thus setting as

$$a_k = h(\ell_k), \quad b_k = \frac{(a_k - 1)^2}{4}, \tag{2.12}$$

$$h(\ell) = \sqrt{1+d} + \frac{1}{2}\sqrt{8 - 4d + \frac{8(2-\ell^2)}{\ell^2\sqrt{1+d}}}, \tag{2.13}$$

$$d = \sqrt[3]{\frac{4(1-\ell^2)}{\ell^4}}, \tag{2.14}$$

$$\ell_0 = \frac{b}{a}, \;\; \ell_k = \frac{\ell_{k-1}(a_{k-1} + b_{k-1}\ell_{k-1}^2)}{1 + c_{k-1}\ell_{k-1}^2}, \;\; k = 1,2, \ldots \tag{2.15}$$

one computes the value ℓ_k, which is actually equals to the lower bound of the smallest singular value in the phase space matrices in the recurrence relation by the equation:

$$\ell_k = \frac{\ell_{k-1}(a_{k-1} + b_{k-1}\ell_{k-1}^2)}{(1 + c_{k-1}\ell_{k-1}^2)}, \;\; (\ell_0 \leq \sigma(X_0)_{min}). \tag{2.16}$$

Introducing the QR decomposition see, e.g., [3,6,7,8] from its history of evolution, we then compute the following expression in the modified Halley's method in equation (2.9):

$$\begin{bmatrix} c_k X_k \\ I \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R, \tag{2.17}$$

$$X_{k+1} = \frac{b_k}{c_k} X_k + \frac{1}{\sqrt{c_k}}\left(a_k - \frac{b_k}{c_k}\right) Q_1 Q_2, \quad k \geq 0. \tag{2.18}$$

where,

$$Q_1Q_2^T = c_kX_k\left(I + c_k^2X_k^2X_k\right)^{-1}. \tag{2.19}$$

Nevertheless as to be noted in a closing remark in this section the inverse Newton's method in this class of methods discussed above also remains valid [7].

Theorem 2.1. *The Halley's method expressed in equation 2.2 has an order of convergence three and it is globally convergent for any starting values.*

The details about Halley's method can be found in [11].

3. The Main Result: The Rotation Matrix

The basic result in the work is that the polar decomposition described above will be used to obtain the rotation matrix. The rotation matrix is the matrix obtained after calculating with the numerical method to obtain the form $A = U_dP$ where we used the Newton's and Halley's method respectively. Therefore, the primary purpose in the computation is obtaining the rotation component in the polar decomposition as the unitary matrix U_d free of charge without additional work. The angular velocity of Rotation matrix thus becomes a central interest which is obtained by taking the logarithm of the component matrix U_d. Thus the resulting matrix is then the angular velocity. It represents the rate at which the rotation matrix is changing with respect to time. This provides detail information about the rotational motion associated with rotation matrix.

Fundamental to the work is the computation of eigenvalues of rotation matrix which represents the rotation angles in radians. The eigenvectors represent the rotation axis. The eigenvalues are calculated using the Chevbyshev method in the senses of Arnoldi's iteration [6].

There is need for investigation in the relationship existing between two given subspaces in the rotation axis. This includes how close they are, do they intersect? Can one be rotated into another? These questions were answered in [6], as result we omit.

3.1. The methodology

Abstractly, we fast forward the Newton's and Halley's methods earlier described in Section 2 which we are using in the computation of rotation matrix and angles of rotations in the form polar decomposition:

The Newton's method:

Newton's method is an iterative optimization technique to find the polar decomposition of a given matrix. The iteration updates the rotation matrix U using the equation.

$$\hat{X}_{k+1} = \frac{1}{2}\left(X_k + X_k^{-T}\right), \quad X_0 = A \quad (k = 0,1,2,\dots). \tag{3.1}$$

The Halley's method:

We hereby extend Halley's iteration formula expressed in equation (2.2) from the single variable equations for polynomial problems of single variable to the rotation matrix problem. The iteration updates the rotation matrix U using the Halley's algorithm:

Algorithm 3.1 of Halley's method.

$$\hat{A} = A - U * U^T$$
$$dU = 0.5 * \left(U * (U^T) + \left(\hat{A}\right)^{-1}\right)$$
$$d2U = U * (U^T * dU) - dU * U^T$$
$$X_{k+1} = U + 2 * dU^{-1} * \left(\hat{A} - dU * U\right) - 0.5 * (d2U)^{-1} * (U^T * U - I)$$
$$U_{k+1} = X_{k+1} * U_k$$

Here, A is the given matrix, U is the rotation matrix, $\bar{A}$ is a temporary variable, dU is the first derivative, and $d2U$ is the second derivative. As with Newton's method, the iteration continues until the residual norm falls below a specified tolerance or reaches a maximum number of iterations.

Both Newton's and Halley's methods are iterative techniques for the polar decomposition, but Halley's method tends to converge faster due to the use of higher-order derivatives in its iteration. Whereas Newton's method is quadratically convergent, the Halley's method is cubically convergent to the desired solution.

The iteration updates the rotation matrix U in succession until convergence is attained.

3.2. Numerical results

We provide results for the computed rotation matrix and rotation angles from the numerical test problem using Newton's and Halley's methods.

$$\begin{bmatrix} 0.72620613 & 0.24406446 & 0.60207421 & 0.53672703 & 0.42840759 \\ 0.24406446 & 0.86565298 & 0.17261807 & 0.24457816 & 0.2212896 \\ 0.60207421 & 0.17261807 & 0.72270461 & 0.43629414 & 0.45835621 \\ 0.53672703 & 0.24457816 & 0.43629414 & 0.85731306 & 0.45898858 \\ 0.42840759 & 0.2212896 & 0.45835621 & 0.45898858 & 0.8481392 \end{bmatrix}$$

The sample matrices used for the tested problems are named A1, A2, A3. These were generated randomly from the matrix market using Sci-py code.

Sample problem 1. Original symmetric Matrix A1:

Below we presented the computed results with the two methods in the form Tables 1 and 2.

Table 1: Result for rotation matrix R_{N_1} using Newton's method (2.5).

$$\begin{bmatrix} 0.45106428 & -0.4596411 & 0.60498727 & 0.15364719 & -0.41660764 \\ -0.01566061 & -0.67094869 & -0.12845163 & 0.69538118 & 0.25285513 \\ -0.88301794 & -0.12872206 & 0.32477441 & -0.25104823 & -0.1987192 \\ -0.09125429 & 0.40854111 & 0.7187919 & 0.50653646 & -0.25161584 \\ -0.09125429 & 0.40854111 & 0.7187919 & 0.50653646 & -0.25161584 \end{bmatrix}$$

Corresponding rotation angles (radians) to Table 1:

$$(0.45106428 \ -0.4596411 \ 0.60498727 \ 0.15364719 \ -0.41660764).$$

Table 2. Numerical results for the rotation matrix R_{H_1} using Halley's method (Algorithm 2.6).

$$\begin{bmatrix} 0.45106428 & -0.4596411 & 0.60498727 & 0.15364719 & -0.41660764 \\ -0.01566051 & -0.67094869 & -0.12845163 & 0.69538118 & 0.25285513 \\ -0.88301794 & -0.12872206 & 0.32477441 & -0.25104823 & -0.1987192 \\ -0.09125429 & 0.40854111 & 0.7187919 & 0.50653646 & -0.25161584 \\ -0.08388052 & 0.37782237 & -0.20023771 & -0.4247228 & 0.76412346 \end{bmatrix}$$

Corresponding rotation angles (radians) to Table 2:

$$(1.65490707 \ 2.42382623 \ 1.78550429 \ 1.86346698 \ 1.45683618).$$

From Tables 1 and 2 above, both Newton's and Halley's methods yielded same results for the rotation matrix and rotation angles for the given sample symmetric matrix of order 5 that was used as a trial experiment. The meaning is that the rotation matrix signifies the orthogonal component of the polar decomposition while the rotation angles denote the angle of rotation about corresponding rotation axis (as provided by the eigenvectors of rotation matrix).

Sample problem 2. Original symmetric Matrix A2:

$$\begin{bmatrix} 0.58569239 & 0.60901746 & 0.88178832 & 0.89680938 & 0.56957174 \\ 0.60901746 & 0.75796097 & 0.93855484 & 0.6431292 & 0.86418686 \\ 0.88178832 & 0.93855484 & 0.92629921 & 0.72485866 & 0.95057222 \\ 0.89680938 & 0.6431292 & 0.72485866 & 0.51015896 & 0.7947987 \\ 0.56957174 & 0.86418686 & 0.95057222 & 0.7947987 & 0.65775385 \end{bmatrix}$$

Similarly, we also presented results for Newton's and Halley's methods obtained from sample problem 2 in the forms of Tables 3 and 4 below.

Table 3. Numerical values for rotation matrix R_{N_2} (Newton's method).

$$\begin{bmatrix} 0.34785022 & -0.52082796 & -0.25396129 & -0.74284229 & -0.01405999 \\ 0.01411612 & -0.16808971 & 0.94195826 & -0.288004229 & 0.07424814 \\ 0.73308661 & 0.75823571 & 0.00833906 & -0.20810013 & 0.59619799 \\ 0.54815094 & -0.36967593 & -0.15038847 & 0.69957671 & -0.23298519 \\ 0.15354642 & 0.02627212 & 0.16141367 & 0.5038593 & 0.82861912 \end{bmatrix}$$

Table 4. Numerical values for rotation matrix R_{H_2} (Halley's method).

$$\begin{bmatrix} 0.34785022 & -0.52082796 & -0.25396129 & -0.01405999 & \\ 0.01411612 & -0.16808971 & 0.94195826 & -0.28800425 & 0.07424814 \\ 0.73308771 & 0.75823571 & 0.00833906 & -0.20810013 & 0.59619799 \\ 0.54815094 & -0.36967593 & -0.15038847 & 0.69957671 & -0.23298519 \\ 0.15354642 & 0.02627212 & 0.16141367 & 0.5038593 & 0.82861912 \end{bmatrix}$$

Table 5. Residual errors in Newton's and Halley's iteration corresponding to results in Tables 3 and 4.

Iteration i	Error in Newton's method R_{N_2} (Table 3)	Error in Halley's method R_{H_2} (Table 4)
1	0.374003479491028	0.374003479491028
2	0.037175953324314135	0.005104640614007619
3	0.000675636185003302	1.116502012463072e-07
4	3.1742294328015e-08	

The computed rotation matrices in Tables 1-4 are named respectively $R_{N_1}, R_{H_1}, R_{N_2}, R_{H_2}$. The R_{N_1} represents the rotation matrix obtained from using Newton's method displayed in Table 1. The R_{H_1} represents the rotation matrix calculated with Halley's method which is displayed in Table 2. We also took another sample matrix and calculated for the rotation matrices respectively using Newton and Halley's method which we named as R_{N_2} and R_{H_2} as displayed in Tables 3 and 4. Note that in each case we calculated for the corresponding rotation angles to the rotation matrices.

Results in Tables 3 and 4 indicated that the two methods of Newton and Halley iterations are good enough for the requested rotation matrices for the sample test problem on symmetric matrix of order 5. The two methods provided errors per iteration which pointed out clearly the convergence pattern of the methods, see e.g., Table 5. The error significantly reduced with each of the two methods per iteration which was a successful convergence to the polar decomposition. Each of the two methods has its own advantages and disadvantages taking into factor the implementation complexity or computational efficiency.

In another development, we computed with modified Halley's method in equation (2.8) using another randomly generated matrix of order 5 for experiment. Here we present the results in the Table 6.

Sample problem 3. The original randomly generated symmetric matrix A3.

$$\begin{bmatrix} 0.39453252 & 0.5878613 & 0.75682223 & 0.6793194 & 0.44043939 \\ 0.5878613 & 0.2709741 & 0.30744075 & 0.89949971 & 0.76326134 \\ 0.75682223 & 0.30744075 & 0.77684865 & 0.71022001 & 0.56315175 \\ 0.6793194 & 0.89949971 & 0.71022001 & 0.40371184 & 0.3386824 \\ 0.44043939 & 0.76326134 & 0.56315175 & 0.3386824 & 0.69355038 \end{bmatrix}$$

Table 6. Rotation matrix R_{H_m} computed with Equation (2.8).

$$\begin{bmatrix} 0.22071376 & -0.14155046 & -0.43188722 & -0.24287271 & 0.84063489 \\ -0.2708229 & 0.34015211 & -0.72935675 & -0.19462202 & -0.47169573 \\ 0.58557385 & 0.78797297 & -0.04740694 & -0.20399091 & 0.02780601 \\ 0.69246072 & -0.44701119 & -0.00612871 & -0.37874468 & -0.38975511 \\ -0.29576857 & 0.13527976 & -0.52578538 & 0.82872004 & -0.06747766 \end{bmatrix}$$

Computed rotation angles (Radians):

$$(2.07248728 \;\; 1.57111134 \;\; 1.83541351 \;\; 2.39025284 \;\; 1.34826502)$$

Residual norm: 4.559081480334167e-07.

4. Discussion

We discussed a novel approach for computing the rotation matrix and angles of rotation based on the polar decomposition of a matrix. The method used the Newton's and Halley's methods in the formulation. We computed with a randomly generated matrix from the matrix market for a practical experimentation using Sci-py codes in Matlab. For instance, Tables 1 and 2 displayed results for the rotation matrices, and their corresponding rotation angles (in radians) for Newton and Halley's methods.

In Tables 3 and 4, we also computed another randomly generated matrix of order 5 wherein, the errors associated with iterations for both Newton's and Halley's methods were obtained. For instance, in Table 3, it took four iterations of computations for Newton's method to converge to a true solution while it was only three iterations for Halley' method to attain the desired solution in Table 4. Meaning that Halley's method converged to the desired solution at the second iteration while it took Newton's method to converged to the solution at the third iteration. The angles of rotations for each method were computed using the Chevbyshev's method in line with Arnoldi iteration [6].

5. Conclusion

We conclude this paper stating a novel approach for the computation of rotation matrix and angles of rotation (in radian) obtained free of charge without additional work in the course of polar decomposition of a matrix. We have extended the techniques of both Newton's and Halley's methods from iteratively finding zeros of polynomial equation of single variable to the matrix rotation problem. The results computed are of high quality which converged to their desired solutions. We paid special attention to the computational aspects by implementing our numerical calculations for the Newton's and Halley's methods in Python codes for Matlab routines in [5].

Compliance with Ethical standards:

Funding: No funding was received from any agency in the course of study.

Conflicts of interest: The author declares that there is no conflict of interest.

References

[1] Benner, P., Nakatsukasa, Y., & Penke, C. (2022). Stable and efficient computation of generalized polar decompositions. *SIAM Journal on Matrix Analysis and Applications*, *43*(3), 1058-1083. https://doi.org/10.1137/21M1411986

[2] Blanchard, P., Zounon, M., Dongarra, J., & Higham, N. (2019). Parallel numerical linear algebra for future extreme-scale systems. NLAFET, University of Manchester. http://dx.doi.org/10.3030/671633

[3] Bjorck, A. (2009). *Numerical methods in scientific computing*: Volume 2 (2nd ed.). SIAM.

[4] Chen, J., & Chow, E. (2014). A stable scaling of Newton Schulz for improving the sign function of computation of a Hermittian matrix. [Preprint]. ANL/MCS-P5059-0114.

[5] Driscoll, T. A. (n.d.). The Schwartz-Christoffel toolbox.
Retrieved from https://w.w.w.math.udel.edu~driscoli/software/sc/

[6] Golub, G., & Van Loan, C. F. (1983). *Matrix computations*. North Oxford Academic Publishing Co. Ltd.

[7] Higham, N. J. (2008). *Functions of matrices: theory and computation*. SIAM.

[8] Nakatsukasa, Y., Bai, Z., & Gygi, F. (2010). Optimising Halley's iteration for computing the matrix polar decomposition. *SIAM Journal on Matrix Analysis and Applications*, *31*(5), 2700-2720. https://doi.org/10.1137/090774999

[9] Nakatsukasa, Y., & Higham, N. J. (2011). Backward stability of iterations for computing the polar decomposition. [MIMS E print]. 2011.103. The University of Manchester, UK.

[10] Nakatsukasa, Y., & Freund, R. (2016). Computing fundamental matrix decompositions accurately via the matrix sign function in two iterations: The power of Zolotarev's functions. *SIAM Review*, *58*(3), 461-493. http://dx.doi.org/10.1137/140990334

[11] Uwamusi, S. E. (2017). Extracting *p*-th root of a matrix with positive eigenvalues via Newton and Halley's methods. *Ilorin Journal of Science*, *4*(1), 1-16.

[12] Uwamusi, S. E. (2017). Computing square root of diagonalizable matrix with positive eigenvalues and iterative solution to Nonlinear system of equation: The role of Lagrange interpolation formula. *Transactions of the Nigerian Association of Mathematical Physics*, *5*, 65-72.

[13] Wang, X., & Cao, Y. (2023). A numerically stable high-order Chevbyshev-Halley type multipoint iterative method for calculating matrix sign function. *AIMS Mathematics*, *8*(5), 12456-12471. http://dx.doi.org/10.3934/math.2023625

[14] Zielinski, P. (1995). The polar decomposition: properties, applications and algorithms. *Roczniki Polskiego Twarzystwa Mathematycznego Seria III. Matematyka Stosowana XXXVIII*.