9-13-2016

A New Algorithm for Computing the Square Root of a Matrix

John Nichols
jn7569@rit.edu

A New Algorithm for Computing the Square Root of a Matrix

by

John Nichols

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Applied and Computational Mathematics

School of Mathematical Sciences
College of Science

Rochester Institute of Technology
Rochester, NY

September 13, 2016

Committee Approval:

Matthew J. Hoffman — Date
Director of Graduate Programs, SMS

Manuel Lopez — Date
Thesis Advisor

James Marengo — Date
Committee Member

Anurag Agarwal — Date
Committee Member

Abstract

There are several different methods for computing a square root of a matrix. Previous research has been focused on Newton's method and improving its speed and stability by application of Schur decomposition called Schur-Newton.

In this thesis, we propose a new method for finding a square root of a matrix called the exponential method. The exponential method is an iterative method based on the matrix equation $(X-I)^2 = C$, for C an $n \times n$ matrix, that finds an inverse matrix at the final step as opposed to every step like Newton's method. We set up the matrix equation to form a $2n \times 2n$ companion block matrix and then select the initial matrix C as a seed. With the seed, we run the power method for a given number of iterations to obtain a $2n \times n$ matrix whose top block multiplied by the inverse of the bottom block is $\sqrt{C} + I$. We will use techniques in linear algebra to prove that the exponential method converges to a specific square root of a matrix when it converges while numerical analysis techniques will show the rate of convergence. We will compare the outcomes of the exponential method versus Schur-Newton, and discuss further research and modifications to improve its versatility.

Contents

1 Introduction

Matrices are tables of elements arranged in set rows and columns, and have long been used to solve linear equations [3]. A number of operations can be applied to matrices but the lack of some properties for some operations, such as commutativity with matrix multiplication, are important to note. Matrices are used in a wide number of fields such as cryptography, differential equations, graph theory, and statistics. They also have uses in applications of electronics and imaging science. Understanding how matrix functions, the mapping of one matrix to another, work and the different methods of running them is an important topic. Computing the square root of a matrix is one such matrix function that can be computed in a number of different ways.

Finding the square root of even a 2×2 matrix gets complicated by the fact that the square of $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is $\begin{bmatrix} a^2+bc & ab+bd \\ ac+cd & bc+d^2 \end{bmatrix}$, so directly taking the square root of each element in a matrix does not work in nondiagonal matrices [16]. The lack of commutivity in matrix multiplication and different characterstics of a matrix also affect the outcome. However, certain groups of matrices have known outcomes when finding their matrix square roots. For example, matrices with nonnegative eigenvalues have a square root with positive real parts called the principal square root [3]. Previous research shows that the matrix square root with positive real parts to its eigenvalues is unique [15]. Matrices of $n \times n$ dimensions and n distinct nonnegative eigenvalues have 2^n square roots, while other matrices such as $\begin{bmatrix} 0 & 4 \\ 0 & 0 \end{bmatrix}$ have no square root.

There are a number of different methods to find the square root of a number or matrix but each have associated trade offs. Diagonalization can be used only on a diagonalizable matrix while Newton's method requires an inverse at each step and cannot handle complex entries [1]. We propose a new method called the exponential method which can be used on nondiagonalizable matrices and requires a matrix inverse only at the end. Schur decomposition can be used as a pre-step to make a more manageable matrix for some methods. In fact, using Schur decomposition with Newton's method has been analyzed [2]. The exponential method can be made more versatile in the entries it can solve by introduction of a complex number. On the other hand, Newton's method has quadratic convergence when it converges while the exponential method displays linear convergence.

In this thesis, we will layout and examine the new exponential method for use in finding the square root of a matrix. The derivation of the method and an algorithm for use for real numbers and matrices are first. Proofs on convergence and rate of convergence examines how the exponential method works and how quickly an approximation is reached. Comparisons with other methods and possible expansions to the new method continue the paper. We finish with a conjecture based on the exponential method about how to quickly find the nondominant square roots of a matrix requiring the eigenvalues and eigenvectors of said matrix

2 The Exponential Method for Real Numbers

The exponential method is an alternative method to find the square root of a real number. The derivation of the method comes from sequences. Suppose we have a polynomial of degree n, which has a nonzero term for all degrees from 0 to n. Let $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} + x^n$ be that polynomial. Since we will be interested in finding the roots of $p(x)$, we can assume $p(x)$ is monic. Thus we can define $p(x) = x^K\hat{p}(x)$ and because of $\hat{p}(x)$ we have a nonzero constant term.

Now, suppose we have a sequence $(x_0, x_1, x_2, \cdots)$ so that the limit as $n \to \infty$ of $\frac{x_{n+1}}{x_n} = \alpha \in \mathbb{C}$, a nonzero root of $p(x)$ and $\hat{p}(x)$. So there exists some $K > 0$ such that $\frac{|\alpha|}{2} < \frac{x_{m+1}}{x_m} \leq \frac{3}{2}|\alpha|$ for all $m \geq K$. This root α then $p(\alpha) = 0$ so we can rewrite $p(\alpha) = 0$ as

$$0 = a_0 + a_1 \lim_{m\to\infty} \frac{x_{m+1}}{x_m} + a_2(\lim_{m\to\infty} \frac{x_{m+1}}{x_m})^2 + \cdots + (\lim_{m\to\infty} \frac{x_{m+1}}{x_m})^m \tag{1}$$

From this equation, Lemma 2.1 arises.

Lemma 2.1.

$$(\lim_{m\to\infty} \frac{x_{m+1}}{x_m})^n = \lim_{m\to\infty} \frac{x_{m+n}}{x_m} \tag{2}$$

Proof. If $\alpha \neq 0$ is a root of $\hat{p}(x)$ and thus a root of $p(x)$, then there exists some K such that $K \leq m < \infty$, then $x_m \neq 0$. Then since (x_m) is a sequence, we can make the following observation.

$$\lim_{m\to\infty}\frac{x_{m+1}}{x_m}=\lim_{m\to\infty}\frac{x_{m+2}}{x_{m+1}}=\cdots=\lim_{m\to\infty}\frac{x_{m+n}}{x_{m+n-1}}$$

Therefore,

$$(\lim_{m\to\infty}\frac{x_{m+1}}{x_m})^n$$

$$=(\lim_{m\to\infty}\frac{x_{m+1}}{x_m})(\lim_{m\to\infty}\frac{x_{m+2}}{x_{m+1}})\cdots(\lim_{m\to\infty}\frac{x_{m+n}}{x_{m+n-1}})$$

$$=\lim_{m\to\infty}(\frac{x_{m+1}}{x_m}\frac{x_{m+2}}{x_{m+1}}\cdots\frac{x_{m+n}}{x_{m+n-1}})=\lim_{m\to\infty}\frac{x_{m+n}}{x_m}$$

We can rewrite (1) as $0=a_0+a_1\lim_{m\to\infty}\frac{x_{m+1}}{x_m}+a_2\lim_{m\to\infty}\frac{x_{m+2}}{x_m}+\cdots+\lim_{m\to\infty}\frac{x_{m+n}}{x_m}$. Now if we multiply through by x_m, we get the approximation

$$0\approx a_0x_m+a_1x_{m+1}+a_2x_{m+2}+\cdots+x_{m+n}$$

Shift values and $x_{m+n}\approx -a_0x_m-a_1x_{m+1}-a_2x_{m+2}+\cdots-a_{n-1}x_{m+n-1}$. We can now use this as a recursive property. We seed the recursion with $(x_0,x_1,\cdots x_{n-1})$ and define for $m\geq 1$. Then, for $m\geq K$, we have

$$x_{m+n-1}\approx -a_0x_m-a_1x_{m+1}-a_2x_{m+2}+\cdots-a_{n-2}x_{m+n-2} \tag{3}$$

□

Consider as an example, and starting pointing of this project finding a root to the polynomial $w^2=C$. Arranging the setup of the exponential method for real numbers is as follows.

1. Given $C \in \mathbb{R}^+$ and $C \neq 1$whose square root we are to find.
2. If $w^2 = (x-1)^2 = C$ then $(x-1)^2 - C = 0$.
3. Thus $x^2 - 2x + (1-C) = 0$.
4. Can rearrange and express as $x^2 = 2x^1 + (C-1)x^0$.
5. Thus, $x_{n+2} = 2x_{n+1} + (C-1)x_n$.
6. Let $x_0 = C$ and $x_1 = C$.
7. Run for n iterations.
8. Finally, $x = \frac{x_{n+1}}{x_n}$ for last n and $x - 1 = \frac{x_{n+1}}{x_n} - 1 \approx \sqrt{C}$.

Here is an example for looking for the square root of 5. So $C = 5$ and let $n = 14$. Then $x_0 = 5$ and $x_1 = 5$. With $2x_1 + (5-1)x_0 = 30$, then $x_2 = 30$. If the algorithm ended there, the ratio of x_2 to x_1 and subtracting 1 is $\frac{30}{5} - 1 = 5$. Below is table of values as the exponential method goes through more iterations.

Iteration n =	x_n	$\frac{x_{n+1}}{x_n} - 1$
1	5	0
2	30	5
3	80	1.6667
4	280	2.5
5	880	2.1429
6	2880	2.2727
7	9280	2.2222
8	30080	2.2414
9	97280	2.3404
10	314880	2.2368
11	1018880	2.2358
12	3297280	2.2362
13	10670080	2.2360
14	34529280	2.2361
15	111738880	2.2361

The square root of 5 rounded to four decimal places is $\sqrt{5} \approx 2.2361$. While x_{n+1} increases with each iteration, $\frac{x_{n+1}}{x_n} - 1$ oscillates between values above and below $\sqrt{5}$ until it reaches 2.2361. So after 14 iterations, the exponential method was able to find a close approximation to $\sqrt{5}$.

2.1 Analysis of the Exponential Method

Before proving convergence for the exponential method, Lemma 2.2 will help set up the sequence of x_n. This lemma will show that the sequence $\frac{x_{n+1}}{x_n} - 1$ has an accumulation point when we place this method within the proper context of the power method in the next section. We'll be able to conclude $\lim_{n\to\infty} \frac{x_{n+1}}{x_n} - 1$ exists.

Lemma 2.2. *For $C \in \mathbb{R}^+$, the sequence $x_0 = x_1 = C$ and $x_{n+2} = 2x_{n+1} + (C-1)x_n$ is monotonically increasing and the set $\frac{x_{n+1}}{x_n}$ is bounded.*

Proof. Have that $x_0 = x_1 = C > 0$, then $x_2 = 2x_1 + (C-1)x_0 = x_1 + Cx_0 > x_1$. In general, if $x_{n+1} > x_n$ for $n \geq 2$ then we have $x_{n+2} = 2x_{n+1} + (C-1)x_n > x_{n+1} + Cx_n > x_{n+1}$. Thus x_n is monotonically increasing for $C > 0$.

Now it must be proven that $\frac{x_{n+1}}{x_n}{}_{n=0}^{\infty}$ is bounded above and below. Observe that by the monotonic part, $\frac{x_{n+1}}{x_n} \geq 1$ and $\frac{x_n}{x_{n+1}} < 1$ for all $n \geq 0$. Also for $n \geq 0$, $x_{n+2} = 2x_{n+1} + (C-1)x_n$. If $C > 0$, then $\frac{x_{n+2}}{x_{n+1}} = 2 + (C-1)\frac{x_n}{x_{n+1}} \leq 2 + C - 1$. Therefore, $\frac{x_{n+2}}{x_{n+1}} \leq 2 + C - 1$ and thus $\frac{x_{n+2}}{x_{n+1}} \leq C + 1$. □

2.2 Power Method and the Companion Matrix

The exponential method can be rewritten to use a companion matrix and to more closely resemble the power method. We have the polynomial $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} + x^n$. Polynomial $p(x)$ is monic, it has an $n \times n$ companion matrix. This companion matrix is of the form

$$L(p) = \begin{pmatrix} -a_{n-1} & -a_{n-2} & -a_{n-3} & \cdots & -a_1 & -a_0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

If $p(x) = (x-1)^2 = x^2 - 2x + 1$, then

$$L(p) = \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix}$$

Then the characteristic polynomial of $L(p)$ is

$$\det \begin{pmatrix} 2-\lambda & -1 \\ 1 & -\lambda \end{pmatrix} = \lambda^2 - 2\lambda + 1 \tag{4}$$

Note that $\lambda^2 - 2\lambda + 1$ is the original polynomial, $x^2 - 2x + 1$.

This is the standard setup between polynomial and companion matrix that goes back to at least Marden [22]. There, Marden proceeds to find the roots by using Gerchgorin disks. For the companion matrix, we choose to use the power method to find the dominant eigenvalue. To bring in the power method, select a seed matrix such that $\mathbf{v} = \begin{bmatrix} x_0 \\ 1 \end{bmatrix}$ with $x_0 = c$ and do $\lim_{m\to\infty}(L(p))^m\mathbf{v}$. As $m \to \infty$, $(L(p))^m\mathbf{v}$ will converge to a matrix with columns that are the dominant eigenvector of c and its dominant eigenvalue [14]. The ratio of $\frac{x_{m+1}}{x_m}$ becomes

$$\lim_{m\to\infty} \frac{x_{m+1}}{x_m} \to \frac{L(p)_1\lambda_1^{m+1} + L(p)_2\lambda_2^{m+1}}{L(p)_1\lambda_1^m + L(p)_2\lambda_2^m} = \lambda_1 \frac{L(p)_1\lambda_1^{m+1} + L(p)_2\lambda_2^{m+1}}{L(p)_1\lambda_1^m + L(p)_2\lambda_2^m} = \lambda_1 \tag{5}$$

Note that the roots of a polynomial can be found with the eigenvalues of its companion matrix [8]. As $m \to \infty$, then the eigenvalue of greatest magnitude would dominate the expression. If the dominating eigenvalue is real, then the power method converges to the dominanting eigenvalue and corresponding eigenvector if the starting point does not belong to another eigenspace [20]. Since the exponential method uses the expression $x_{m+1}(x_m)^{-1}$ and has $x_0, x_1 \in \mathbb{R}$, then the matrix square root associated with the real dominating eigenvalue would be found.

2.3 Convergence

From subsection 2.2, the exponential method is shown to be a modified version of the power method. This allows us to use theorems about the power method in the analysis of the new algorithm. One example of this is showing that the exponential method converges.

Theorem 2.3. *Let $C \in \mathbb{R}^+$ be a real positive number for $(x-1)^2 = x^2 - 2x + 1 = C$. The companion matrix for $x^2 - 2x - C + 1 = 0$ is $M = \begin{bmatrix} 2 & C-1 \\ 1 & 0 \end{bmatrix}$. The power method applied to the companion matrix M converges if the dominant eigenvalue is positive and real.*

Proof. The proof for Theorem 2.3 uses matrix powers. Seen in [20] as Theorem 10.3 on pages 590 and 591. Thus the solution to $x^2 - 2x - C + 1 = 0$ is $x = 1 \pm \sqrt{C}$ which are also eigenvalues of the companion matrix. Since the power method finds the eigenvalue with greatest magnitude, $x = 1 + \sqrt{C}$ which is what the exponential method converges to.

□

2.4 Rate of Convergence for Numbers

Iterative methods have a speed at which a sequence approaches a limit if the sequence is convergent. Speed is one of the important characteristics of an method that can result in it being used or not. Comparison between truncation errors of different iterations is often used in matrix series [11]. For this reason, it is vital to find the rate of convergence of the exponential method as one possible means of comparison to other iterative methods.

Theorem 2.4. *Let $C \in \mathbb{R}^+$ be a real number. When $\lim_{n\to\infty} \frac{x_{n+1}}{x_n} = 1 + \sqrt{C}$, the exponential method has linear rate of convergence with order $\alpha = 1$ since that results in a finite limit.*

Proof. As shown in Theorem 2.3, $\lim_{n\to\infty} \frac{x_{n+1}}{x_n} = 1 + \sqrt{C}$. Note that $x_{n+2} = 2x_{n+1} + (C-1)x_n$. By the limit law of reciprocals, $\lim_{n\to\infty} \frac{x_n}{x_{n+1}} = \frac{1}{1+\sqrt{C}}$. Arrange the rate of convergence formula for the exponential method as

$$\lim_{n\to\infty} \frac{|\frac{x_{n+2}}{x_{n+1}} - (1+\sqrt{C})|}{|\frac{x_{n+1}}{x_n} - (1+\sqrt{C})|^{\alpha}} \tag{6}$$

Have $\alpha = 1$. Then

$$\lim_{n\to\infty} \frac{|\frac{x_{n+2}}{x_{n+1}} - (1+\sqrt{C})|}{|\frac{x_{n+1}}{x_n} - (1+\sqrt{C})|}$$

$$= \lim_{n\to\infty} \frac{|\frac{2x_{n+1}+(C-1)x_n}{x_{n+1}} - (1+\sqrt{C})|}{|\frac{x_{n+1}}{x_n} - (1+\sqrt{C})|}$$

$$= \lim_{n\to\infty} \frac{|(1-\sqrt{C}) - (1-C)\frac{x_n}{x_{n+1}})|}{|\frac{x_{n+1}}{x_n}||1-(1+\sqrt{C})\frac{x_n}{x_{n+1}}|}$$

$$= \lim_{n\to\infty} \frac{|1-\sqrt{C}||1-(1+\sqrt{C})\frac{x_n}{x_{n+1}})|}{|\frac{x_{n+1}}{x_n}||1-(1+\sqrt{C})\frac{x_n}{x_{n+1}}|}$$

$$= \frac{|1-\sqrt{C}|}{1+\sqrt{C}}$$

So when $\alpha = 1$, then asymptotic error constant is a finite limit of $\frac{|1-\sqrt{C}|}{1+\sqrt{C}}$. But when $\alpha > 1$ such that $\alpha = 1 + \epsilon$ with $\epsilon > 0$, then

$$\lim_{n\to\infty} \frac{|\frac{x_{n+2}}{x_{n+1}} - (1+\sqrt{C})|}{|\frac{x_{n+1}}{x_n} - (1+\sqrt{C})|^{1+\epsilon}}$$

$$= \frac{|1-\sqrt{C}|}{1+\sqrt{C}} (\lim_{n\to\infty} \frac{1}{|\frac{x_{n+1}}{x_n} - (1+\sqrt{C})|^{\epsilon}}) \tag{7}$$

But note that

$$(\lim_{n\to\infty} \frac{1}{|\frac{x_{n+1}}{x_n} - (1+\sqrt{C})|^{\epsilon}}) \to \infty$$

Thus when $\alpha > 1$, there is no finite limit for the asymptotic error constant for the exponential method. As such, $\alpha = 1$ and that means the exponential method has a linear rate of convergence for real numbers.

□

3 The Exponential Method for Matrices

Just as Newton's method can be applied to the matrix equation $X^2 - C = 0$ for matrices X and C [1], the exponential method can also be applied to the matrix equation $(X - I)^2 - C = 0$ with I the identity matrix. By generalizing the scalar iteration, the exponential method can then be used to find the square root of a matrix. For this, we will utilize an iterative method so that we generate a sequence of matrices $S_0, S_1, S_2, \cdots$ where $S_0 = I$ and $S_1 = \alpha C$ for $\alpha \in \mathbb{C}$ and $\alpha \neq 0$. Then for $k \geq 2$, we have $S_k = 2S_{k-1} + (C - I)S_{k-2}$.

The exponential method for real numbers uses the ratio $\frac{x_{k+1}}{x_k}$ for k iterations, which has a matrix equivalent of $S_{k+1}(S_k^{-1})$. Because we take a matrix inverse as a final step, it is important that there is no singular matrix S_k. A matrix is singular if and only if its determinant is 0. The following lemma is an immediate consequence.

Lemma 3.1. *The matrix C and the matrices S_k, as defined by $S_k = 2S_{k-1} + (C - I)S_{k-2}$ and with $S_0 = I$ and $S_1 = \alpha C$ with $\alpha \in \mathbb{C}$ and $\alpha \neq 0$, share the same eigenspaces.*

Proof. Let $\mathbf{v}$ be an eigenvector of C with corresponding eigenvalue λ. Then $S_2\mathbf{v} = [2\alpha C + (C - I)I]\mathbf{v} = [2\alpha\lambda + \lambda - 1]\mathbf{v}$. Note that $[2\alpha\lambda + \lambda - 1]$ is a scalar multiple.

Now let λ_{k+1} be an eigenvalue of S_{k+1} corresponding to $\mathbf{v}$ and λ_k be an eigenvalue of S_k corresponding to $\mathbf{v}$. Then $S_{k+2}\mathbf{v} = [2S_{k+1} + (C-I)S_k]\mathbf{v} = [2\lambda_{k+1} + (\lambda-1)\lambda_k]\mathbf{v}$. Note that $[2\lambda_{k+1} + (\lambda - 1)\lambda_k]$ is a scalar multiple. Since this is done for $\mathbf{v}$ being some random eigenvector of C with its corresponding eigenvalue, it can be extended to the remaining eigenvectors and eigenvalues. Thus the matrices C and S_k share the same eigenspace □

Lemma 3.1 can be used to show that only finitely many of the S_k may fail to be invertible since the scalar multiple can equal 0 but only finite times as $k \to \infty$.

Lemma 3.2. *For C an invertible matrix, any nonzero $\alpha \in \mathbb{C}$, if we define $S_0 = I$, $S_1 = \alpha C$ and $S_{k+2} = 2S_{k+1} + (C - I)S_k$. Then S_k is invertible for all $k \geq K$ with $K > 1$.*

Proof. Let $S_k\mathbf{v} = \mu\mathbf{v}$, $S_{k+1}\mathbf{v} = \nu\mathbf{v}$, and λ an eigenvalue of C. Then $S_{k+2}\mathbf{v} = (2\nu + (\lambda - 1)\mu)\mathbf{v}$. Note that $2\nu + (\lambda - 1)\mu = 0$ if and only if $\frac{\nu}{\mu} = \frac{1-\lambda}{2}$. That is $2\nu + (\lambda - 1)\mu = 0$, then S_{k+2} is singular. Note that $\nu_k = \mu_{k+1}$. Thus, $\frac{\mu_{k+1}}{\mu_k}$ cannot be $\frac{1-\lambda}{2}$ infinitely often.

If the equality were true infinitely often, then for any $\epsilon > 0$, there exists K such that the complex norm $|(\frac{\mu_{k+1}}{\mu_k} - 1) - \sqrt{\lambda}| < \epsilon$ for all $k \geq K_\lambda$. But there would be a $k_0 \geq K$ so that $\frac{\mu_{k_0+1}}{\mu_{k_0}} = \frac{1-\lambda}{2}$ such that $|\frac{1-\lambda}{2} - 1 - \sqrt{\lambda}| < \epsilon$. Then the following complex norms are of the form

$$
\begin{aligned}
|1 - \lambda - 2 - 2\sqrt{\lambda}| &< 2\epsilon \\
|-(1 + \lambda) - 2\sqrt{\lambda}| &< 2\epsilon \\
|\lambda + 2\sqrt{\lambda} + 1| &< 2\epsilon \\
|(\sqrt{\lambda} + 1)^2| &< 2\epsilon \qquad (8)
\end{aligned}
$$

The process produces $\sqrt{\lambda}$ with positive real part. So $\sqrt{\lambda}+1$ has $\text{Re}(\sqrt{\lambda}+1) > 1$, therefore $|(\sqrt{\lambda}+1)^2| > 1$. This contradicts the above inequality if ϵ is chosen to be $0 < \epsilon < \frac{1}{2}$. So while the exponential method might have a singular S_k, as $k \to \infty$ it becomes very unlikely that we end there.

□

With each S_k being invertible for sufficiently large k, we can apply similar arithmetic to matrices as we did with real numbers. The main difference is that matrix multiplication is not commutative in general. With that fact in mind, the exponential method can be implemented for matrices as follows:

1. Declare some nonsingular matrix C with dimensions (n, n).
2. Initialize i for number of iterations, $S_0 = I$ and $S_1 = C$.
3. Initialize $Z = C - I$.
4. For i iterations or until S_i becomes too ill-conditioned, do $S_{i+1} = 2S_i + (Z)(S_{i-1})$,
5. After iteration steps stop, find S_i^{-1}.
6. Set $n \times n$ matrix $Q = S_{i+1}(S_i^{-1}) - I$.

3.1 Rate of Convergence for Matrices

The rate of convergence of the exponential method for numbers can also be applied when the method is used on matrices. One would expect that the exponential method converges linearly when it converges since it is based on the power method [21]. To prove this, the exponential method will be defined as $S_{n+2} = 2S_{n+1} + (C - I)S_n$. We can modify Theorem 2.4 as follows.

Theorem 3.3. *Let C be an invertible matrix with a square root. When $\lim_{n\to\infty}(S_{n+1})(S_n)^{-1} = I + \sqrt{C}$, the exponential method for matrices has a linear rate of convergence with order $\alpha = 1$.*

Proof. Given that $n \geq K$ as defined in Lemma 3.2, $\lim_{n\to\infty} S_{n+1}(S_n)^{-1} = I + \sqrt{C}$. Note that $S_{n+2} = 2S_{n+1} + (C - I)S_n$. Then, $\lim_{n\to\infty} S_n(S_{n+1})^{-1} = (I + \sqrt{C})^{-1}$. Arrange the rate of convergence formula for the exponential method as

$$\lim_{n\to\infty} |S_{n+2}(S_{n+1})^{-1} - (I + \sqrt{C})|(|S_{n+1}(S_n)^{-1} - (I + \sqrt{C})|)^{-1\times\alpha} \tag{9}$$

With $\alpha = 1$. Then

$$\lim_{n\to\infty} |S_{n+2}(S_{n+1})^{-1} - (I + \sqrt{C})|(|S_{n+1}(S_n)^{-1} - (I + \sqrt{C})|)^{-1}$$

$$= \lim_{n\to\infty} |(2S_{n+1} + (C - I)S_n(S_{n+1})^{-1}) - (I + \sqrt{C})|(|S_{n+1}(S_n)^{-1} - (I + \sqrt{C})|)^{-1}$$

$$= \lim_{n\to\infty} |(2I+(C-I)S_n(S_{n+1})^{-1})-(I+\sqrt{C})|(|I-(I+\sqrt{C})S_n(S_{n+1})^{-1}||S_{n+1}(S_n)^{-1}|)^{-1}$$

$$= \lim_{n\to\infty} |I - \sqrt{C} - (I - C)S_n(S_{n+1})^{-1}|(|I - (I + \sqrt{C})S_n(S_{n+1})^{-1}||S_{n+1}(S_n)^{-1}|)^{-1}$$

$$= \lim_{n\to\infty} |I - \sqrt{C}||I - (I + \sqrt{C})S_n(S_{n+1})^{-1}|(|I - (I + \sqrt{C})S_n(S_{n+1})^{-1}||S_{n+1}(S_n)^{-1}|)^{-1}$$

$$= \lim_{n\to\infty} |I - \sqrt{C}||S_n(S_{n+1})^{-1}|$$

$$= |I - \sqrt{C}|(I + \sqrt{C})^{-1} \quad (10)$$

So when $\alpha = 1$, then asymptotic error constant is a finite limit of $|I - \sqrt{C}|(I + \sqrt{C})^{-1}$. If $\alpha > 1$, then since the real case in the previous chapter amounts to the 1×1 matrix case, this would imply that $\alpha > 1$ in the real case. This has been proven incorrect.

□

3.2 Existence of Matrix Square Roots and Number of Square Roots

Before one starts the task of finding a square root of some sample matrix, it is worthwhile to make sure that the sample matrix even has a square root. As noted before, some matrices do not have a square root such as $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$. One can determine if a square root exists for a matrix by use of Jordan canonical form. In [3], Gordon presents a theorem for the existence of square roots of a matrix. This theorem works on singular and nonsingular matrices and also proves how many distinct square roots a matrix has.

Theorem 3.4. *Let $C \in M_n$, with M_n being the set of all complex matrices. There are two branches depending on C being nonsingular or singular.*

(a) If C is nonsingular and has μ distinct eigenvalues and ν Jordan blocks in its Jordan canonical form, then it has at least 2^μ and at most 2^ν nonsimilar square roots. At least one square root can be expressed as a polynomial in C.

(b) If C is singular and has Jordan canonical form $C = SJS^{-1}$, then let $J_{k_1}(0) + J_{k_2}(0) + \cdots + J_{k_p}(0)$ be the singular part of J. J is arranged such that the blocks are in decreasing order of size k, so $k_1 \geq k_2 \geq \cdots \geq k_p \geq 1$. Let δ be the difference between successive pairs of k_p, such that $\delta_1 = k_1 - k_2$, $\delta_3 = k_3 - k_4$, etc. C has a square root if and only if $\delta_i = 0$ or 1 for $i = 1, 3, 5, \cdots$ and $k_p = 1$ for odd p. A square root that is

a polynomial in C exists if and only if $k_1 = 1$, which is equivalent to rank C = rank C^2.

If C has a square root, its set of square roots lies in finitely many different similarity classes [3].

Proof. The proof for Theorem 3.4 uses the Jordan form theorem. Seen in [3] as Theorem 5.2 on pages 23 and 24. □

3.3 Role of Matrix Inverse

In block-form matrix multiplication, have the matrix product $\begin{bmatrix} M & N \\ O & P \end{bmatrix} \begin{bmatrix} G \\ H \end{bmatrix}$ where M, N, O, P, G, and H are all matrices of size $n \times n$. This is called block-invariant if there exists an $n \times n$ matrix R satisfying $\begin{bmatrix} M & N \\ O & P \end{bmatrix} \begin{bmatrix} G \\ H \end{bmatrix} = \begin{bmatrix} RG \\ RH \end{bmatrix}$.

The particular systems is of the form $A = \begin{bmatrix} 2I & C - I \\ I & 0 \end{bmatrix} \begin{bmatrix} S_2 \\ S_1 \end{bmatrix}$ which iterates starting with $S_2 = C$ and $S_1 = I$. The main result is that when this iteration approaches a left block-invariant matrix $\begin{bmatrix} S_2 \\ S_1 \end{bmatrix}$ then $R = S_2 S_1^{-1}$ and $(R - I)^2 = C$. As the exponential method iterates for k iterations, the output eventually becomes $A^k B$ with A the companion matrix and B the seed matrix. Matrix B is a block matrix with the top half designated S_2 and the bottom half designated S_1. The process below will show how S_1^{-1} is used in the algorithm and that it will lead one to the square root of the initial matrix C.

Let

$$A = \begin{pmatrix} 2I & C - I \\ I & 0 \end{pmatrix}$$

and

$$B = \begin{pmatrix} S_2 \\ S_1 \end{pmatrix}$$

With A and B, set $R = S_2(S_1)^{-1}$ such that after k iterations

$$A^kB = \begin{pmatrix} S_2 \\ S_1 \end{pmatrix}$$

From the exponential method, The next S_2 is a linear recurrence of the two previous terms and this is the same as $(R)S_2$. So $(R)S_2 = 2S_2 + (C-I)S_1$ and $(R)S_1 = S_2$.

$$\begin{aligned}
(R)S_2 &= 2S_2 + (C-I)S_1 \\
(R-I)S_2 &= S_2 + (C-I)S_1 \\
(R-I)(R)S_1 - (R-I)S_1 &= S_2 + (C-I)S_1 - (R-I)S_1 \\
(R-I)(R-I)S_1 &= S_2 + (C-I-(R-I))S_1 \\
(R-I)(R-I)S_1 &= S_2 + (C-R)S_1 \\
(R-I)(R-I)S_1S_1^{-1} &= S_2S_1^{-1} + (C-R)S_1S_1^{-1} \\
(R-I)^2 &= R - (C-R) = C
\end{aligned}$$

With $R = S_2(S_1)^{-1}$, subtract I from it to find the square root of the initial matrix C.

3.4 Continued Fractions and Matrix Square Root

Continued fractions are representations of numbers and matrices represented through an iterative process that involves integers and reciprocals. It was proved that the approximants of Newton's method for a square root can be represented by a continued fraction expansion [6]. One can use continued fractions to prove that the exponential method goes to a square root of a matrix.

Theorem 3.5. *Let $S_{n+2} = 2S_{n+1} + (C-I)S_n$ be an iterative process with C a square matrix with a square root. If the sequence $S_i{}_{i=0}^{\infty}$ is composed entirely of nonsingular matrices, then the expression $S_{n+1}(S_n)^{-1} - I = \sqrt{C}$.*

Proof. Let $Y_i = S_{i+1}(S_i)^{-1}$. Then $S_{n+2} = 2S_{n+1} + (C-I)S_n$ becomes $Y_{n+1} = 2I + (C-I)Y_n^{-1}$ with the substitution. For the sake of easier reading while using continued fractions, let $Y_n^{-1} = \frac{1}{Y_n}$. Now express Y_n as a continued fraction.

$Y_1 = 2I + (C - I)\frac{S_0}{S_1} = 2I + \frac{(C-I)}{C} = 3I - \frac{I}{C}$
$Y_2 = 2I + \frac{C-I}{3I-\frac{I}{C}}$
$Y_2 = 2I + \frac{C-I}{2I+\frac{C-I}{3I-\frac{I}{C}}}$
$\dots$
$Y_n = 2I + \frac{C-I}{2I+\cdots\frac{\cdots}{2I+\frac{(C-I)}{C}}}$

So now Y_n has a continued fraction expression. If the sequence Y_i converges to a matrix F, then $F = 2I + \frac{(C-I)}{F}$. Multiply the right hand side by F to get $F^2 = 2F + C - I$. From there

$F^2 = 2F + C - I$
$F^2 - 2F + I = C$
$(F - I)^2 = C$

Since $(F-I)^2 = C$ and the sequence Y_i converges to F, then $F - I = S_{n+1}(S_n)^{-1} - I = \sqrt{C}$. □

3.5 Specific Square Root and the Power Method

As stated before, the exponential method is based on the power method and shares some of the same characteristics. The power method can fail to converge to the dominant eigenvalue λ_1 if there is another eigenvalue λ_2 such that $\lambda_1 \neq \lambda_2$ but $|\lambda_1| = |\lambda_2|$. The exponential method converges to the quotient $S_{n+1}(S_n)^{-1}$ which is equal to $\sqrt{A} + I$, if it converges. One can use the power method's convergence to explain why the exponential method converges to a matrix square root whose eigenvalues have positive real parts.

Theorem 3.6. *Let $n \times n$ Jordan block matrix A have a square root and has k Jordan blocks A_k with k being the number of distinct eigenvalues of A. Let $2n \times 2n$ matrix L be of the form $\begin{bmatrix} 2I & A-I \\ I & 0 \end{bmatrix}$. Then the exponential method of the companion matrix form applied to L will output a matrix square root of A with eigenvalues that have nonnegative real parts, if it converges.*

Proof. It was shown earlier that the exponential method of the companion matrix is a modified power method and converges to the expression $S_{n+1}(S_n)^{-1}$. The power method converges to the eigenpair where the eigenvalue has the greatest magnitude out of all eigenvalues of the matrix. It was also shown earlier that the companion matrix L has twice the eigenvalues of A and are of the form $\lambda_i = 1 \pm \sqrt{\mu_j}$ with λ_i being the ith eigenvalue of L with $i = 1, 2, \cdots, 2n$ and μ_j being the jth eigenvalue of A with $j = 1, 2, \cdots, n$. So, $\lambda_i = 1 + \sqrt{\mu_j}$ and $\lambda_{i+1} = 1 - \sqrt{\mu_j}$. Note that since A is a Jordan block matrix, operations on the separate blocks are not connected and thus the exponential method would work on each individual block. With this fact and $\lambda_i = 1 \pm \sqrt{\mu_j}$, one can divide the possible outcomes into three instances depending on μ_j.

(1) If μ_j is a positive real number, then $\sqrt{\mu_j} \in \mathbb{R}$. Thus, $|1 + \sqrt{\mu_j}| > |1 - \sqrt{\mu_j}|$ and the exponential method will go towards the eigenvalue $\lambda_i = 1 + \sqrt{\mu_j}$ of the ith Jordan block A_i.

(2) If μ_j is a complex number of the form $a + b\mathrm{i}$ with $a, b \in \mathbb{R}$, then $\sqrt{\mu_j}$ will exist and be in the forms $p + q\mathrm{i}$ and $-p - q\mathrm{i}$ with $p, q \in \mathbb{R}$. However, since $\mu_i \notin (-\infty, 0]$ then $p \neq 0$ and we may assume $p > 0$ and that $\sqrt{\mu_i} = p + q\mathrm{i}$. Make the substitution to get $1 \pm \sqrt{\mu_j} = 1 \pm (p + q\mathrm{i})$. Distribute the minus to get $|1 + p + q\mathrm{i}|$ and $|1 - p - q\mathrm{i}|$. Thus, $|1 + p + q\mathrm{i}| > |1 - p - q\mathrm{i}|$ and the exponential method will go towards the eigenvalue $\lambda_i = 1 + \sqrt{\mu_j}$ of the ith Jordan block A_i.

(3) If μ_j is a negative real number, then $\sqrt{\mu_j}$ is of the form $q\mathrm{i}$ with $q \in \mathbb{R}$. Note that $|1 + q\mathrm{i}| = \sqrt{(1)^2 + (q)^2}$ and $|1 - q\mathrm{i}| = \sqrt{(1)^2 + (-q)^2}$. Thus, $|1 + \sqrt{\mu_j}| = |1 - \sqrt{\mu_j}|$ even though $1 + \sqrt{\mu_j} \neq 1 - \sqrt{\mu_j}$. This situation will cause the power method to fail since there the two eigenvalues of greatest magnitude are not equal to each other and the method oscillates at each iteration. If the $\sqrt{\mu_j} = q\mathrm{i}$ was graphed onto a complex plane, it would lie directly on the y-axis. Thus, the exponential method fails to find the square root of a matrix with a negative real eigenvalue.

In the first and second instances, the eigenvalue of A has a positive real part that is equal to or greater than 1, so $S_{n+1}(S_n)^{-1} - I$ will not result in a negative real part. In the third instance, there are two eigenvalues whose magnitudes are equal to each other but larger than all the other eigenvalues so the exponential method does not converge in that case. Since the eigenvalues of A have a positive real part, they are unique [15]. □

Finally, it can be proven that the exponential method and Newton's method will converge to the same square root of a matrix if they converge.

Theorem 3.7. *Newton's method and the exponential method enacted on a matrix with a square root C will converge to the same square root of C, the unique matrix square root that has positive real parts to its eigenvalues.*

Proof. Higham has shown that Newton's method will converge to the matrix square root that has positive real parts to its eigenvalues if it converges [1]. Theorem 3.6 from earlier shows that the exponential method will converge to the matrix square root with eigenvalues that have positive real parts if it converges. Given that the square root of a matrix with positive real parts to its eigenvalues is in the right complex half-plane, that square root is thus unique [15]. Thus, both Newton's method and the exponential method converge to the same square root of a matrix if they converge. □

3.6 Schur Decomposition and Convergence

It has been proven that the exponential method converges for finding the square roots of real numbers. Schur decomposition can be used as a pre-step to Newton's method to improve stability when the initial matrix is ill-conditioned [2]. The method can also be applied to the eigenvalues of a matrix to determine the square root of said matrix.

Given a square matrix X of order m with m distinct eigenvalues λ, one can find matrix U such that $UXU^{-1} =$

$$\begin{pmatrix} \Lambda_1 & 0 & 0 & \cdots & 0 & \\ 0 & \Lambda_2 & 0 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \Lambda_{m-1} & 0 & \\ 0 & 0 & \cdots & 0 & \Lambda_m & \end{pmatrix}$$

Which is the Jordan normal form of matrix X. Each submatrix Λ_p for $1 \leq p \leq m$ is of the form

$$\begin{pmatrix} \lambda_1 & 1 & 0 & \cdots & 0 & \\ 0 & \lambda_2 & 1 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \lambda_{p-1} & 1 & \\ 0 & 0 & \cdots & 0 & \lambda_p & \end{pmatrix}$$

From the exponential method, let $(X-I)^2 = C$ with C an order m matrix with a square root and I the identity matrix. Then $X^2 - 2X + I = C$ and $X_{n+2} = 2X_{n+1} + (C-I)X_n$.

Let C have two matrices, W and Δ, such that $WCW^{-1} =$

$$\begin{pmatrix} \Delta_1 & 0 & 0 & \cdots & 0 & \\ 0 & \Delta_2 & 0 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \Delta_{p-1} & 0 & \\ 0 & 0 & \cdots & 0 & \Delta_p & \end{pmatrix}$$

Each block matrix Δ_p is of the form

$$\begin{pmatrix} \delta_1 & 1 & 0 & \cdots & 0 & \\ 0 & \delta_2 & 1 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \delta_{p-1} & 1 & \\ 0 & 0 & \cdots & 0 & \delta_p & \end{pmatrix}$$

So matrix C is converted into Jordan normal form and each block Δ_p has a main diagonal with δ_p, the eigenvalues of C. Now apply the exponential method block by block.

$$\begin{pmatrix} \lambda_{n+2} & 1 & 0 & \cdots & 0 & \\ 0 & \lambda_{n+2} & 1 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \lambda_{n+2} & 1 & \\ 0 & 0 & \cdots & 0 & \lambda_{n+2} & \end{pmatrix} =$$

$$\begin{pmatrix} 2\lambda_{n+1} & 2 & 0 & \cdots & 0 & \\ 0 & 2\lambda_{n+1} & 2 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & 2\lambda_{n+1} & 2 & \\ 0 & 0 & \cdots & 0 & 2\lambda_{n+1} & \end{pmatrix} +$$

$$\begin{pmatrix} \delta_n - 1 & 1 & 0 & \cdots & 0 & \\ 0 & \delta_n - 1 & 1 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \delta_n - 1 & 1 & \\ 0 & 0 & \cdots & 0 & \delta_n - 1 & \end{pmatrix} \begin{pmatrix} \lambda_n & 1 & 0 & \cdots & 0 & \\ 0 & \lambda_n & 1 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \lambda_n & 1 & \\ 0 & 0 & \cdots & 0 & \lambda_n & \end{pmatrix}$$

Bring terms together.

$$\begin{pmatrix} \lambda_{n+2} & 1 & 0 & \cdots & 0 & \\ 0 & \lambda_{n+2} & 1 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \lambda_{n+2} & 1 & \\ 0 & 0 & \cdots & 0 & \lambda_{n+2} & \end{pmatrix} =$$

$$\begin{pmatrix} 2\lambda_{n+1} & 2 & 0 & \cdots & 0 & \\ 0 & 2\lambda_{n+1} & 2 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & 2\lambda_{n+1} & 2 & \\ 0 & 0 & \cdots & 0 & 2\lambda_{n+1} & \end{pmatrix} +$$

$$\begin{pmatrix} \lambda_n(\delta_n - 1) & \delta_n - 1 + \lambda_n & 0 & \cdots & 0 & \\ 0 & \lambda_n(\delta_n - 1) & \delta_n - 1 + \lambda_n & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \lambda_n(\delta_n - 1) & \delta_n - 1 + \lambda_n & \\ 0 & 0 & \cdots & 0 & \lambda_n(\delta_n - 1) & \end{pmatrix}$$

Thus, $\lambda_{n+2} = 2\lambda_{n+1} + \lambda_n(\delta_n - 1)$ with each block matrix Λ having their main diagonal of eigenvalues of X be defined by the recurrence relation of the exponential method. Assemble the block matrices to get UXU^{-1}, which is the square root of WCW^{-1}.

3.7 Diagonalization and Schur Decomposition

One issue with the exponential method is that it finds one square root of a given matrix when there can be more. However, if matrix C is diagonalizable then Schur decomposition can be used. Schur decomposition takes the given matrix C and converts it into MUM^{-1}, with U being an upper triangular matrix.

If $C = MUM^{-1}$ and $U^{1/2}$ is the square root of U, then

$(MU^{1/2}M^{-1})^2 = MU^{1/2}(M^{-1}M)U^{1/2}M^{-1} = MUM^{-1} = C$

Thus, $MU^{1/2}M^{-1}$ is a square root of C. If $C = MUM^{-1}$ then $C^{1/2} = MU^{1/2}M^{-1}$ and $U^{1/2}$ the Schur form of C [4]. By finding the upper triangular matrix $U^{1/2}$, one can change the sign of the elements to find other square roots of U and thus of A. Let $U_1^{1/2}$ be the square root of U with positive elements. Thus, if

$$\begin{pmatrix} a & b \\ 0 & d \end{pmatrix}$$

and

$$U_1^{1/2} = \begin{pmatrix} a & b \\ 0 & d \end{pmatrix}$$

then

$$(U_1^{1/2})^2 = \begin{pmatrix} x^2 & xy + yz \\ 0 & z^2 \end{pmatrix} = \begin{pmatrix} a & b \\ 0 & d \end{pmatrix}$$

.

Then $x = a^{1/2}$, $z = d^{1/2}$, and $y = \frac{b}{a^{1/2}+d^{1/2}}$. Suppose the two square roots of a are $a^{1/2}$ and $-a^{1/2}$ and the two square roots of d are $d^{1/2}$ and $-d^{1/2}$. Thus y can be written in different forms depending on the square roots of a and d. This changes the signs of the elements of $U^{1/2}$

$$
\begin{aligned}
y &= \frac{b}{a^{1/2}+d^{1/2}} \\
&= \frac{b}{a^{1/2}-d^{1/2}} \\
&= \frac{b}{-a^{1/2}+d^{1/2}} = \frac{-b}{a^{1/2}-d^{1/2}} \\
&= \frac{b}{-a^{1/2}-d^{1/2}} = \frac{-b}{a^{1/2}+d^{1/2}}
\end{aligned}
$$

The exponential method can then be used on U. Knowing the elements of one square root allows one to find the other square roots. If the square root of d is $-d^{1/2}$ and $z = d^{1/2}$, then $-z = -d^{1/2}$ and $(-z)^2 = (-d^{1/2})^2$. Thus

$$
U_2^{1/2} = \begin{pmatrix} x & y \\ 0 & -z \end{pmatrix} = \begin{pmatrix} a^{1/2} & \frac{b}{a^{1/2}-d^{1/2}} \\ 0 & -d^{1/2} \end{pmatrix}
$$

.

From the above form of $U^{1/2}$, one can multiply by M^{-1} and M to receive a different square root of C with different elements. Multiplying $U_1^{1/2}$ and $U_2^{1/2}$ by -1 gives $U_3^{1/2}$ and $U_4^{1/2}$, respectively.

$$
U_3^{1/2} = \begin{pmatrix} -x & -y \\ 0 & -z \end{pmatrix}
$$

$$
U_4^{1/2} = \begin{pmatrix} -x & -y \\ 0 & z \end{pmatrix}
$$

Thus by finding U, the Schur form of A, we can find the square root of U and manipulate its entries to find other square roots of U and therefore of C. Using the exponential method to directly find other square roots is covered later. The algebra for 2×2 matrices is direct but it becomes more difficult for 3×3 matrices and those with higher orders.

3.8 Uniqueness of Matrix Square Roots

By putting the square roots of a matrix on the complex plane, it is shown that there is a unique square root that exists on the open right half of that plane. In [15], Johnson et al. prove that for any complex matrix there is a square root whose set of eigenvalues exist in the right half of the complex plane and that it is unique.

Theorem 3.8. *Let $C \in M_n$ such that the intersection of set of eigenvalues of C and $(\infty, 0]$ is empty. Then there is a unique $X \in M_n$ such that $X^2 = C$ with the set of eigenvalues of B contained in the open right half of the complex plane [15].*

Proof. The proof for Theorem 3.8 uses Lyapunov's theorem and Schur's theorem. It is found in [15] as Theorem 5 on pages 56 and 57. □

Theorem 3.8 from Johnson et al. tells us that only one square root of a matrix has eigenvalues with positive real components. The exponential method is based on the power method, which finds the dominant eigenvalue and associate eigenvector of a matrix [14]. By using it to find the eigenvalues of the companion matrix $L(C)$ and eigenvectors of C, we can then find other square roots of C with C having order greater than 2. This requires use of the following Theorem 3.9 and finds some more but not all of the remaining square roots.

Theorem 3.9. *Let C be an $n \times n$ matrix with a square root and eigenvalues $\lambda_1, \lambda_2, \cdots \lambda_m$. Then let D be a $2n \times 2n$ block matrix of the form $\begin{bmatrix} 2I & C - I \\ I & 0 \end{bmatrix}$. The eigenvalues of D are of the form $\mu_m = 1 \pm \sqrt{\lambda_m}$.*

Proof. Note that D has four eigenvalues and C has two eigenvalues. Then λ_1 is the largest eigenvalue of C and λ_2 is the next largest eigenvalue. Block $C - I$ has eigenvalues $\lambda_1 - 1$ and $\lambda_2 - 1$. Matrix D is the companion matrix of matrix C and has characteristic polynomial of $X^2 - 2X - (\lambda_1 - 1)$ or $X^2 - 2X - (\lambda_2 - 1)$. For $\lambda_1 - 1$, the solution of the quadratic equation is $1 \pm \sqrt{\lambda_1 - 1 + 1} = 1 \pm \sqrt{\lambda_1}$. For $\lambda_2 - 1$, the solution of the quadratic equation is $1 \pm \sqrt{\lambda_1 - 1 + 1} = 1 \pm \sqrt{(\lambda_2)}$.

Thus, each eigenvalue of C results in two eigenvalues of D of the forms $1 + \sqrt{\lambda_n}$ and $1 - \sqrt{\lambda_n}$. □

3.9 Finding Other Square Roots of a Matrix

Now one can continue on to find other square roots of C using the eigenvalues of the companion matrix $L(C)$, but not all that remain. The simple Theorem 3.10 is based on decomposing the matrix C to $C = VSV^{-1}$ [17].

Theorem 3.10. *Let C be an $n \times n$ matrix with square roots and let $L(C)$ be the block companion matrix formed from the characteristic polynomial of C. Let v_i be the eigenvector associated with λ_i, the ith eigenvalue of C, with $i = 1, 2, \cdots, n$, with descending magnitude. Let matrix V have the form $\begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$. Let μ_k be the eigenvalues of D formed from λ_i and are associated with the same eigenvector of C for $k = 1, 2, \cdots, n-1$, and let matrix S be a diagonal matrix whose entries are μ_k arranged by descending magnitude. Then, $VSV^{-1} - I = \sqrt{C}$.*

Proof. Note that $S = \begin{bmatrix} \mu_1 & 0 & \cdots & 0 \\ 0 & \mu_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \mu_k \end{bmatrix}$, which is equal to

$$\begin{bmatrix} \sqrt{\lambda_1}+1 & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda_2}+1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \sqrt{\lambda_k}+1 \end{bmatrix} \text{ if } \mu_k = \sqrt{\lambda_n} + 1.$$

Note that the columns of V are arranged so that the eigenvectors of C pair with their corresponding eigenvalues of C which form the entries of $L(C)$. So VSV^{-1} represents the decomposition of $\sqrt{C} + I$ since $S = \sqrt{C} + I$. By subtracting the identity matrix I from VSV^{-1}, one revieves $VSV^{-1} - I = \sqrt{C}$. □

What is interesting to note is the exponential method does not need a full set of eigenvalues and eigenvectors of a matrix to find a square root. It is able to return the principal square root with one eigenvalue and associated eigenvector if it converges. The power method can calculate one eigenvalue and one eigenvector at a time until deflation is used to find other eigenpairs [10]. Even with the exponential method being based on the power method, it is surprising and counter-intuitive that it is able to find a square root without complete knowledge of the eigenvalues and eigenvectors of a matrix.

4 Modifications and Observations

One advantage of the exponential method compared to other root-finding algorithms is its customizability. The base method is to use $(x-1)^2 = c$ for real numbers but the $(x+i-1)^2 = c$ can be used if c is a complex number. Similarly, the base method for matrices with real eigenvalues is $(S-I)^2 = C$ but $(S+iI-I)^2 = C$ can be used if C has complex eigenvalues. The modified algorithms are shown below.

4.1 Modification for Negative Numbers

With the necessary modification, the exponential method can find the square root of numbers that Newton's method cannot handle. This modification allows us to find the square roots negative negative and thus imaginary square roots to those numbers. The setup is as follows.

1. Let $w = x + i - 1$ and $w^2 = C$.
2. Then $(x + [i-1])^2 - C = 0$ if $w^2 - C = 0$.
3. Thus $x^2 + 2x(i-1) + (-2i - C) = 0$.
4. Can rearrange and express as to $x^2 = 2(i-1)x^1 + (-2i - C)x^0$.
5. Thus, $x_{n+2} = (-2i+2)x_{n+1} + (2i + C)x_n$.
6. Finally, $x = \frac{x_{n+1}}{x_n}$ for last n and $x + i - 1 = \frac{x_n}{x_{n-1}} + i - 1 = \sqrt{C}$.

4.2 Modification for Negative Eigenvalues

The modification for negative numbers can also be applied to matrices with negative eigenvalues. This modification allows us to find the square roots of matrices with negative eigenvalues and thus imaginary square roots to those eigenvalues. The setup for matrices with negative eigenvalues is as follows.

1. Declare some nonsingular matrix A with dimensions (n, n).
2. Initialize i for number of iterations, $S_1 = C$ and $S_2 = C$.
3. Initialize $Z = 3C - 2I + 2iI$.
4. For i iterations or until S_i is too ill-conditioned, do $S_{i+1} = 2S_i - 2iS_i + (Z)(S_{i-1})$,
5. After step iterations stop, find S_i^{-1}.
6. Set nxn matrix $Q = S_{i+1}(S_i^{-1}) + iI - I$.

4.3 Ratio of Elements in 2×2 Matrices and Graphing

Given a 2×2 matrix C, let S be the square root of C so $S^2 = C$. Let x, y, w, and z be the elements of S and let a, b, c, d be the elements of C. Thus

$$\begin{pmatrix} x & y \\ w & z \end{pmatrix} \begin{pmatrix} x & y \\ w & z \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\begin{pmatrix} x^2 + yw & xy + yz \\ xw + zw & yw + z^2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

From the above matrices

$$\begin{aligned} a &= x^2 + yw \\ b &= xy + yz \\ c &= xw + zw \\ d &= yw + z^2 \end{aligned}$$

Rearrange to get

$$\begin{aligned} P &= x^2 + yw - a = 0 \\ Q &= xy + yz - b = 0 \\ R &= xw + zw - c = 0 \\ S &= yw + z^2 - d = 0 \end{aligned}$$

Since S is the square root of C, the above is true.

For any polynomials α, β, γ, and δ, $\alpha P + \beta Q + \gamma R + \delta S = 0$ at the points solving the four equations. From there, set the polynomials equal to an element of S to knock off a leading term.

If $\alpha = y$ and $\beta = x$, then

$$\alpha P = x^2 y + y^2 w - ay = 0$$
$$\beta Q = x^2 + xyz - bx = 0$$
$$\alpha P - \beta Q = xyz - y^2 w - by + ay = 0$$

The set of equations is to the third degree. If $\beta = w$ and $\gamma = y$, then

$$\beta Q = xyw + ywz - bw = 0$$
$$\gamma R = xyw + ywz - cy = 0$$
$$\beta Q - \gamma R = bw - cy = 0$$

Thus, $bw = cy$ and $w = \frac{c}{b}y$ if $b \neq 0$. So the initial matrix C has a ratio that is preserved in $\sqrt{C}$ for b and c. The exponential method preserves the ratio of b and c so if a seed has $\frac{b_{n+1}}{c_{n+1}} \neq \frac{b_S}{c_S}$, then the algorithm won't return a correct square root.

By getting the ratio $w = \frac{c}{b}y$, then P, Q, R, and S can be rewritten as

$$P = x^2 + \frac{c}{b}y^2 - a = 0$$
$$Q = xy + yz - b = 0$$
$$R = \frac{c}{b}yx + \frac{c}{b}yz - c = 0$$
$$S = \frac{c}{b}y^2 + z^2 - d = 0$$

Note that P, Q, R, and S are in terms of three variables with b and c as constants. With these manipulations, we can get 3d plots of the exponential method with each variable being a plane. Where the planes intersect indicate a square root for the matrix.

The intersection of the three surfaces of the first plot show the square root of the matrix $\begin{bmatrix} 33 & 24 \\ 48 & 57 \end{bmatrix}$. The intersection of the three surfaces of the second plot show the square root of the matrix $\begin{bmatrix} 29 & 20 \\ 20 & 29 \end{bmatrix}$.

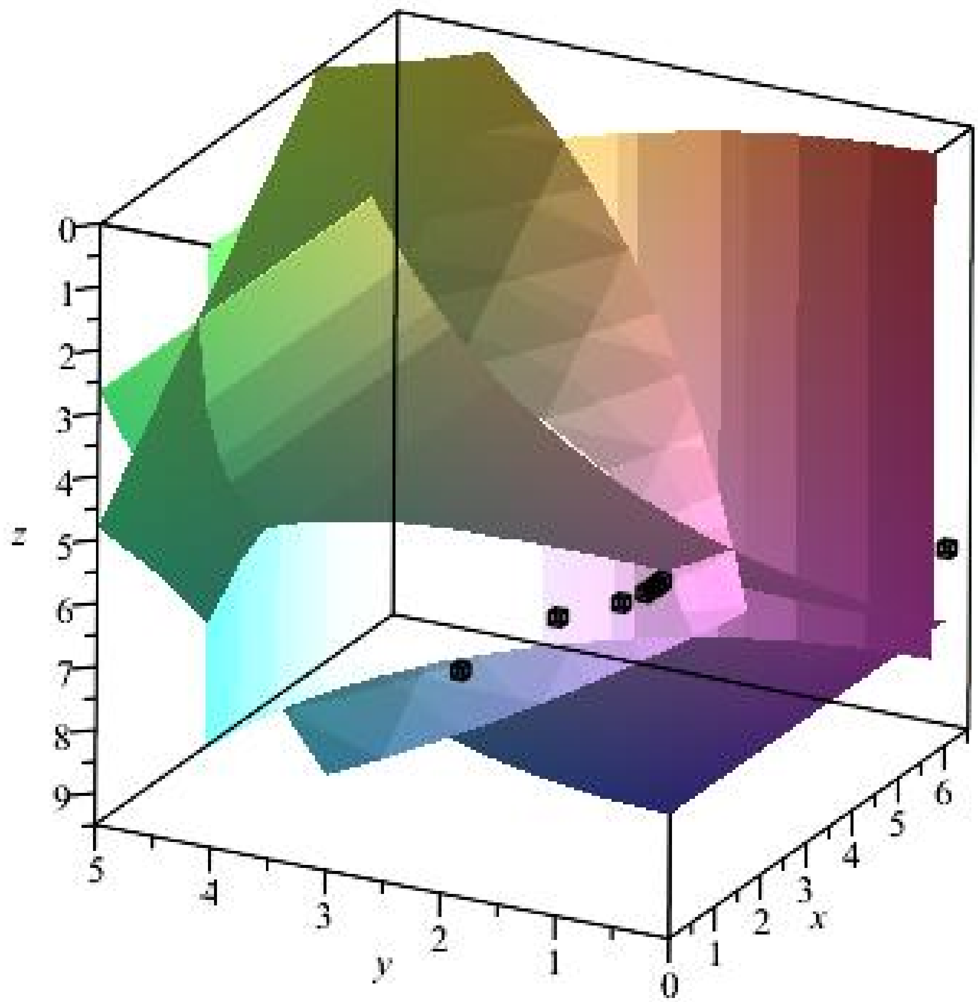

Figure 1: 3d plot of intersection of surfaces for a square root of $\begin{bmatrix} 33 & 24 \\ 48 & 57 \end{bmatrix}$.

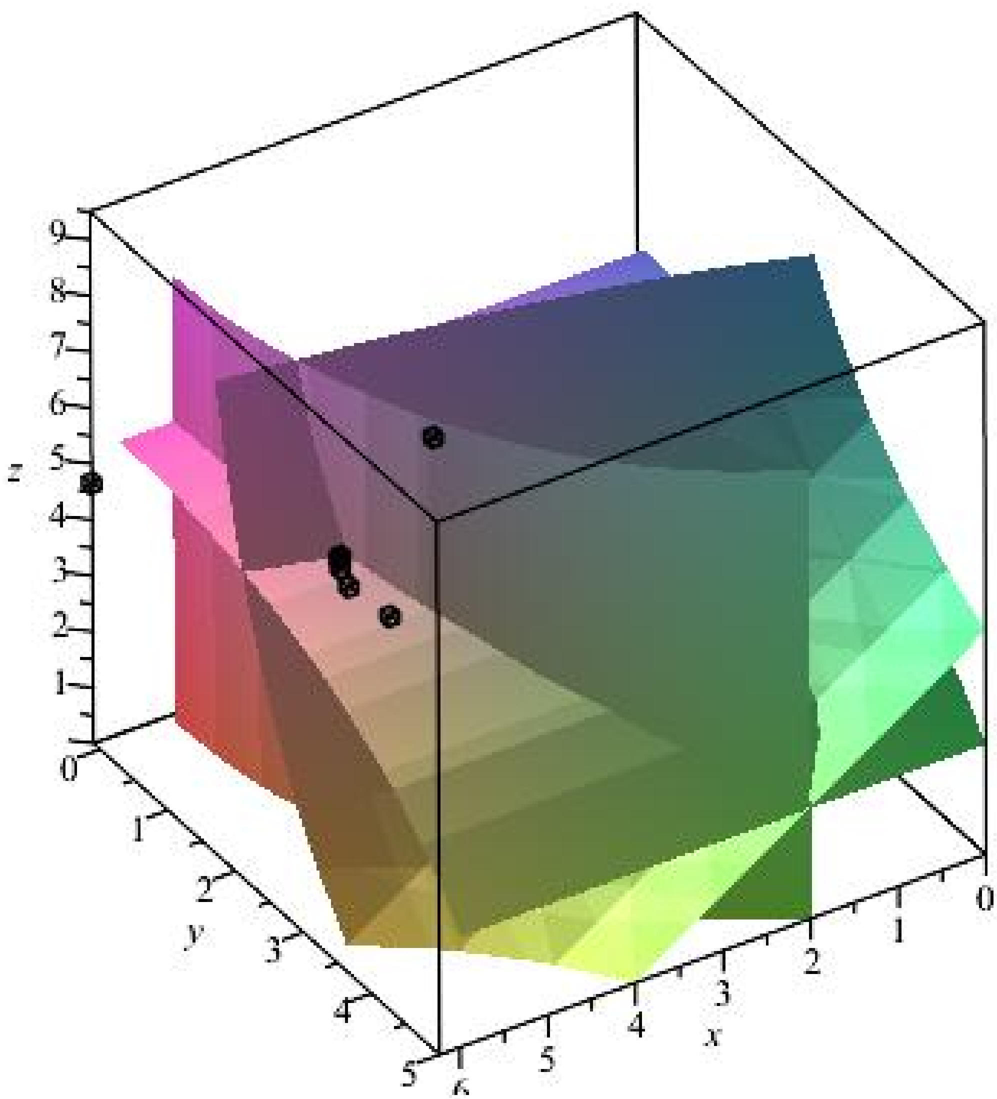

Figure 2: 3d plot of intersection of surfaces for a square root of $\begin{bmatrix} 29 & 20 \\ 20 & 29 \end{bmatrix}$.

4.4 Ratios of Entries Not in Main Diagonal

Another further point of research is the ratio of matrix entries not in the main diagonal for $n \times n$ matrices where $n \geq 3$. It was shown previously the observation that for 2×2 matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, the ratio of $\frac{b}{c}$ is preserved for its square root as long as $b \neq 0$ and $c \neq 0$. This property appears to extend to larger matrices.

Conjecture 4.1. *For 2×2 matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ that has a square root, the ratio of $\frac{b}{c}$ is preserved for its square root. Then for $n \times n$ matrices that have a square root and $n > 2$, the ratio of two entries in the secondary diagonal of a square block is equal between the matrix and its square root as long as none of the entries are in the main diagonal of the matrix and none of them are equal to* 0.

4.5 Heuristic Rule for Ill-Conditioned Matrices

The exponential method computes a matrix inverse once at the end while Newton's method a matrix inverse at every step. While computationally expensive, this means that Newton's method can detect a singular matrix at the iteration it appears and stop there. The exponential method will continue on until at the end the singular matrix is detected. The matrix S_k is invertible at every step as shown in Lemma 3.2, but becomes very ill-conditioned as the size of its entries increases. Thus the program would be unable to find an inverse to S_k due to computer limitations. Some heuristic to tell the program to stop as the matrices become too ill-conditioned would be useful.

Let B be the output matrix of the iterative portion at the current step and let n be the order of the matrix B. To determine if the matrices are becoming too ill-conditioned to continue, calculate at each step $r = \frac{(\max(b_{i,j})^n}{\det B}$. If r becomes large enough that it is represented as a floating point number, then the program is unable to find the inverse of B at that step. For Maple 2016, the algorithm failed when r was to the order of 10^{16} or greater. Thus stopping before that point, when r was to the order of 10^{15}, means the computer can calculate the inverse of the output matrix before it becomes too ill-conditioned.

The calculation $r = \frac{(\max(b_{i,j})^n}{\det B}$ works in that almost singular matrices have determinants that are almost 0. However, $\frac{1}{\det B}$ could be masked by large entries in B. Multiplying by $\max(b_{i,j})^n$ accounts for the largest entry being raised to the nth power.

An example of this is finding the square root of the matrix $\begin{bmatrix} 33.0001 & 24 \\ 48 & 57.0001 \end{bmatrix}$. After 42 iterations, r goes beyond an order of 10^{15} and is represented as a floating point number. S_{42} is very ill-conditioned and the program cannot calculate its inverse because of computer limitations. Thus one can determine when the exponential method is going to run into trouble and find an appropriate cut off point.

5 Results and Comparison

There are a handful of methods that can find the square root of a number and a matrix. The comparison is to Newton's method with Schur decomposition. The exponential method has a linear rate of convergence but Newton's method has a quadratic rate of convergence [4]. While Newton's method returns the square root faster than the exponential method, the latter has a simpler iteration step. This is because Newton's method finds calculates a matrix inverse at every step while the exponential method only needs a matrix inverse at the end. Comparisons between the two methods are shown below. Note that the results are to eight significant figures.

1.

In this case, both methods return a square root of $\begin{bmatrix} 33 & 24 \\ 48 & 57 \end{bmatrix}$. However, Schur-Newton returns the output earlier than the exponential method. Every iteration for $n = 1, 2, \cdots, 10$ is recorded while every fifth iteration is recorded for $n > 10$.

Iteration n =	Schur-Newton	Exponential
1	$\begin{bmatrix} 17 & 12 \\ 24 & 29 \end{bmatrix}$	$\begin{bmatrix} 33 & 24 \\ 48 & 57 \end{bmatrix}$
2	$\begin{bmatrix} 9.42926 & 6.02926 \\ 12.0585 & 15.4585 \end{bmatrix}$	$\begin{bmatrix} 1.85853 & 0.05853 \\ 0.11707 & 1.91707 \end{bmatrix}$
3	$\begin{bmatrix} 6.22525 & 3.20172 \\ 6.40344 & 9.42697 \end{bmatrix}$	$\begin{bmatrix} 11.8665 & 8.00936 \\ 16.0187 & 19.8758 \end{bmatrix}$
4	$\begin{bmatrix} 5.17383 & 2.17374 \\ 4.34749 & 7.34758 \end{bmatrix}$	$\begin{bmatrix} 3.02123 & 0.37417 \\ 0.74834 & 3.39540 \end{bmatrix}$
5	$\begin{bmatrix} 5.00475 & 2.00475 \\ 4.00951 & 7.00951 \end{bmatrix}$	$\begin{bmatrix} 8.05335 & 4.85980 \\ 9.71961 & 12.9131 \end{bmatrix}$
6	$\begin{bmatrix} 5.00000 & 2.00000 \\ 4.00000 & 7.00000 \end{bmatrix}$	$\begin{bmatrix} 3.69227 & 0.78458 \\ 1.56916 & 4.47686 \end{bmatrix}$
7	$\begin{bmatrix} 5.00000 & 2.00000 \\ 4.00000 & 7.00000 \end{bmatrix}$	$\begin{bmatrix} 6.62369 & 3.57645 \\ 7.15290 & 10.2001 \end{bmatrix}$
8	$\begin{bmatrix} 5 & 2 \\ 4 & 7 \end{bmatrix}$	$\begin{bmatrix} 4.12242 & 1.14577 \\ 2.29154 & 5.26819 \end{bmatrix}$
9	$\begin{bmatrix} 5 & 2 \\ 4 & 7 \end{bmatrix}$	$\begin{bmatrix} 5.93797 & 2.92623 \\ 5.85246 & 8.86421 \end{bmatrix}$
10	$\begin{bmatrix} 5 & 2 \\ 4 & 7 \end{bmatrix}$	$\begin{bmatrix} 4.41432 & 1.42017 \\ 2.84034 & 5.83449 \end{bmatrix}$
15	$\begin{bmatrix} 5 & 2 \\ 4 & 7 \end{bmatrix}$	$\begin{bmatrix} 5.52189 & 2.21874 \\ 4.43748 & 7.43767 \end{bmatrix}$
20	$\begin{bmatrix} 5 & 2 \\ 4 & 7 \end{bmatrix}$	$\begin{bmatrix} 4.93160 & 1.93161 \\ 3.86323 & 6.86322 \end{bmatrix}$
25	$\begin{bmatrix} 5 & 2 \\ 4 & 7 \end{bmatrix}$	$\begin{bmatrix} 5.02275 & 2.02275 \\ 4.04550 & 7.04550 \end{bmatrix}$
30	$\begin{bmatrix} 5 & 2 \\ 4 & 7 \end{bmatrix}$	$\begin{bmatrix} 4.99258 & 1.99258 \\ 3.98516 & 6.98516 \end{bmatrix}$

2.

Here, the methods look for a square root of the Hilbert matrix

$$\begin{bmatrix} 1 & 0.5 & 0.33333 & 0.25 \\ 0.5 & 0.33333 & 0.25 & 0.2 \\ 0.33333 & 0.25 & 0.2 & 0.166666 \\ 0.25 & 0.2 & 0.166666 & 0.142857 \end{bmatrix}.$$

Schur-Newtons fails to return a square root while the exponential method succeeds. Every fifth iteration is represented after the first iteration, up to iteration 25.

Iteration n =	Schur-Newton			
1	1	0.5	0.33333	0.25
	0.5	0.33333	0.25	0.2
	0.33333	0.25	0.2	0.166666
	0.25	0.2	0.166666	0.142857
5	0.91150	0.33870	0.19352	0.13074
	0.33870	0.35574	0.24155	0.18408
	0.19325	0.24155	0.25516	0.19947
	0.13074	0.18408	0.19947	0.22889
10	0.91146	0.33903	0.19271	0.13098
	0.33903	0.35285	0.24745	0.18069
	0.19271	0.24745	0.24110	0.20855
	0.13098	0.18069	0.20855	0.22260
15	0.91146	0.33903	0.19271	0.13098
	0.33903	0.35285	0.24745	0.18069
	0.19271	0.24745	0.24110	0.20855
	0.13098	0.18069	0.20855	0.22260
20	0.91146	0.33903	0.19271	0.13098
	0.33903	0.35285	0.24745	0.18069
	0.19271	0.24745	0.24110	0.20855
	0.13098	0.18069	0.20855	0.22260
25	−47.9369	550.364	−1324.05	861.119
	−27.5129	313.962	−754.801	491.092
	−19.6778	223.986	−538.435	350.440
	−15.4097	175.166	−421.089	274.138

Iteration n =	Exponential			
1	1	0.5	0.33333	0.25
	0.5	0.33333	0.25	0.2
	0.33333	0.25	0.2	0.16666
	0.25	0.2	0.166666	0.142857
5	0.90638	0.34800	0.19465	0.12847
	0.34800	0.32286	0.25167	0.20089
	0.19465	0.25167	0.23208	0.20647
	0.12847	0.20089	0.20647	0.19700
10	0.91055	0.34267	0.19207	0.12812
	0.34267	0.33726	0.25170	0.19174
	0.19207	0.25170	0.23524	0.21043
	0.12812	0.19174	0.21043	0.20940
15	0.91104	0.34081	0.19229	0.12964
	0.34081	0.34488	0.25058	0.18535
	0.19229	0.25058	0.23571	0.21115
	0.12964	0.18535	0.21115	0.21515
20	0.91126	0.33989	0.19243	0.13042
	0.33989	0.34875	0.24995	0.18214
	0.19243	0.24995	0.23607	0.21140
	0.13042	0.18214	0.21140	0.21819
25	0.91136	0.33972	0.19201	0.13116
	0.33945	0.35073	0.24926	0.18087
	0.19249	0.24969	0.23617	0.21160
	0.13079	0.18075	0.21127	0.21960

The square of the exponential method's output is the initial matrix. Schur-Newton gets close to the matrix square root but then diverges as the iterations continue.

3.

In this instance, Schur-Newton is used while the exponential method incorporates the modification to handle complex numbers from above. The initial matrix is $\begin{bmatrix} -9 & 1 \\ 0 & -4 \end{bmatrix}$ and has eigenvalues -9 and -4. With the modification, Schur-Newton fails to converge while the modified exponential method succeeds. Every fifth iteration after the first is shown, up to iteration 30.

Iteration n =	Schur-Newton
1	$\begin{bmatrix} -4 & 0.50000 \\ 0 & -1.50000 \end{bmatrix}$
5	$\begin{bmatrix} -2.52459 & 0.84153 \\ 0 & 1.68306 \end{bmatrix}$
10	$\begin{bmatrix} 7.20264 & -2.40088 \\ 0 & -4.80176 \end{bmatrix}$
15	$\begin{bmatrix} 47.72390 & -15.9079 \\ 0 & -31.8159 \end{bmatrix}$
20	$\begin{bmatrix} -1.40549 & 0.46849 \\ 0 & 0.93699 \end{bmatrix}$
25	$\begin{bmatrix} 0.35253 & -0.11751 \\ 0 & -0.23502 \end{bmatrix}$
30	$\begin{bmatrix} -4.36977 & 1.45659 \\ 0 & 2.91318 \end{bmatrix}$

Iteration n =	Exponential
1	$\begin{bmatrix} -8+i & 1 \\ 0 & -3+i \end{bmatrix}$
5	$\begin{bmatrix} 0.18622 - 2.51671i & -0.00566 + 0.07176i \\ 0 & 0.15789 - 2.51578i \end{bmatrix}$
10	$\begin{bmatrix} 0.018886 - 2.98165i & -0.00454 + 0.19638i \\ 0 & -0.00382 - 1.99970i \end{bmatrix}$
15	$\begin{bmatrix} 0.00115 - 2.99955i & -0.00022 + 0.19991i \\ 0 & 0.000053 - 1.99957i \end{bmatrix}$
20	$\begin{bmatrix} 0.00005 - 3.00000i & -0.00001 + 0.20000i \\ 0 & -1.85847 * 10^{-7} - 2.00000i \end{bmatrix}$
25	$\begin{bmatrix} 2.47090 * 10^{-6} - 3.00000i & -4.96647 * 10^{-7} + 0.20000i \\ 0 & -1.23320 * 10^{-8} - 1.99999i \end{bmatrix}$
30	$\begin{bmatrix} 8.55423 * 10^{-8} - 3.00000i & -1.70311 * 10^{-8} + 0.20000i \\ 0 & 3.83062 * 10^{-8} - 2.00000i \end{bmatrix}$

The exponential method's output when squared is approximately the square root of the initial matrix. However the Schur-Newton's output when squared is $\begin{bmatrix} 19.0949 & 1.45659 \\ 0 & 2.91318 \end{bmatrix}$ and not close to the initial matrix.

6 Possible Future Research

In the previous sections, we analyzed the inner workings of the exponential method. Now we look outwards to find applications of the new algorithm to other fields. One such field is Clifford algebra, a form of associative algebra with more dimensions representing planes and volumes that has useful properties. There is the quadratic form formed of variables x, y which form a vector $[xy]^T$ as part of the exponential method. In general, we have $f : V \to \mathbb{F}$, where $\mathbb{F}$ is a field that cannot have characteristic of 2. Thus, $f(\lambda, X) = \lambda^2 f(X)$.

A Clifford map is a vector space homomorphism $\phi : V \to \mathbb{F}$ and $(\phi v)^2 = f(v)$ as vectors. If $\phi(\lambda v) = \lambda\phi(v)$ so $\phi(\lambda v)^2 = \lambda^2\phi(v)^2 = \lambda^2 f(v) = f(\lambda v)$. From there, we can define $\phi(v) = \sqrt{f(v)}$ by the exponential method.

Now, we can define a map $f : V \to \mathbb{F}^{nxn}$ with $f(\lambda v) = \lambda^2 f(v)$. We will call this a higher-order quadratic form. Define the corresponding higher-order Clifford map as $\phi : V \to \mathbb{F}^{nxn}$ so that $(\phi(v))^2 = f(v)$. Then we can construct $\phi(v)$ to be the dominant left invariant block $\begin{bmatrix} SN_2 \\ SN_1 \end{bmatrix}$ of the matrix $\begin{bmatrix} 2I & [f(v)] - I \\ I & 0 \end{bmatrix}$. Then we have to consistently choose square roots with Clifford maps for all v. Thankfully, Theorem 3.6 and Theorem 3.8 allows one to accomplish this and give a constructive approach to $\phi(v)$. Finally, such a left invariant block exists for every v we could define $\phi(v) = SN_2(SN_1)^{-1} - I$.

By linking the exponential method to Clifford algebra and Clifford maps, it can be possibly expanded to applications in various topics. Clifford algebra is used in physics for the study of spacetime and in imaging science for the study of computer vision.

7 Concluding Remarks

The exponential method can calculate the dominant square root of a matrix when it converges with a linear rate. If it converges, it converges to the matrix square root with nonnegative real parts to its eigenvalues. What is interesting to note is that the new method does not appear to need the full set of eigenvalues and eigenvectors to find a matrix square root. While the exponential method converges slower than

Schur-Newton to the same matrix square root, it can be easily modified to converge for a wider group of matrices.

From previous research, we know whether a given matrix has a square root and the number of square roots. Furthermore, the uniqueness of the matrix square root in the right half of the complex plane gives insight into why the exponential method converges to that particular square root. Matrix functions are an important part of linear algebra and research is complicated by some matrix operations lacking certain properties. Stable and versatile ways to implement matrix functions benefit a variety of theoretical and applied fields.

8 Appendix

Provided below is code for implementation of the base exponential method as it should be used in Maple.

A := input (Enter an $n \times n$ matrix) ;

count := input (Enter the number of iterations) ;

n := Dimension(A) ;

Id := IdentityMatrix(n) ;

SN1 := A ;

SN2 := A ;

Z := A - Id ;

for i from 1 to count do temp := SN2 ; SN2 := (2SN2) + Multiply(Z,SN1) ; SN1 := temp ; end do :

ISN1 := Matrix Inverse(SN1) ;

SR := Multiply(SN2,ISN1) - Id ;

ANS := Multiply(SR,SR) ;

The following is the code as implemented in Maple for the exponential method with a companion matrix.

A := input (Enter an $n \times n$ matrix) ;

count := input (Enter the number of iterations) ;

n := Dimension(A) ;

Id := IdentityMatrix(n) ;

Z := A - Id ;

L := <<2Id, Id> | <Z, 0Id>> (L is a $2n \times 2n$ matrix) ;

seed:= <<A, Id>> ;

for i from 1 to count do seed := Multiply(L,seed) ; end do ; (seed is a $2n \times n$ matrix)

SN2 := <<seed(1,1), ..., seed(1,n)> | <..., ..., ...> | <seed(n,1), ..., seed(n,n)>> ;

SN1 := <<seed(n+1,1), ..., seed(n+1,n)> | <..., ..., ...> | <seed(2n,1), ..., seed(2n,n)>> ;

ISN1 := MatrixInverse(SN1) ;

SR := Multiply(SN2,ISN1) - Id ;

ANS := Multiply(SR,SR) ;

References

[1] Nicholas J. Higham. *Newton's Method for the Matrix Square Root*. Mathematics of Computation, 46(174): 537-549, April 1986.

[2] Ake Björck and Sven Hammarling. *A Schur Method for the Square Root of a Matrix*. Linear Algebra and its Applications, 52(53): 127-140, 1983.

[3] Crystal Monterz Gordon. *The Square Root Function of a Matrix*. Thesis, Georgia State University, 2007. `http://scholarworks.gsu.edu/math_theses/24`

[4] Nicholas J. Higham. *Computing Real Square Roots of a Real Matrix.* Linear Algebra and its Applications, 88(89): 405-430, 1987.

[5] Thab Ahmad Abd AL-Baset AL-Tamimi. *The Square Roots of 2x2 Invertible Matrices.* International Journal of Difference Equations, 6(1): 61-64, 2011.

[6] Andrej Dujella. *Newton's formula and continued fraction expansion of* $\sqrt{d}$. Experiment. Math. 10 (2001), 125-131, 2000.

[7] Mohammed A. Hasan. *A Power Method for Computing Square Roots of Complex Matrices.* Journal of Mathematical Analysis and Applications, 213: 393-405, 1997.

[8] Alan Edelman and H. Murakami. *Polynomial Roots from Companion Matrix Eigenvalues.* Mathematics of Computation, 64(210): 763-776, April 1995.

[9] Gianna M. Del Corso. *Estimating an Eigenvector by the Power Method with a Random Start.* SIAM Journal of Matrix Analysis and Applications, 18(4): 913-937, 1997.

[10] E. Pereira and J. Vit 'oria. *Deflation of Block Eigenvalues of Block Partitioned Matrices with an Application to Matrix Polynomials of Commuting Matrices.* Computers and Mathematics with Applications, 42(8): 1177-1188, 2001.

[11] N. J. Young. *The Rate of Convergence of a Matrix Power Series.* Linear Algebra and its Applications, 35: 261-278, February 1981.

[12] Chun-Hua Guo. *Convergence Rate of an Iterative Method for a Nonlinear Matrix Equation.* SIAM Journal of Matrix Analysis and Applications, 23(1): 296-302, 2001.

[13] Pentti Laasonen. *On the Iterative Solution of the Matrix Equation* $AX^2 - I = 0$. Mathematical Tables and Other Aids to Computation, 12(62): 109-116, April 1958.

[14] Maysum Panju. *Iterative Methods for Computing Eigenvalues and Eigenvectors.* The Waterloo Mathematics Review, 1: 9-18, 2011.

[15] Charles R. Johnson, Kazuyoshi Okubo, Robert Reams *Uniqueness of matrix square roots and an application.* Linear Algebra and its Applications, 323: 51-60, 2001.

[16] Same Northshield. *Square Roots of 2x2 Matrices.* SUNY Plattsburgh, Class Notes, 2000

[17] Ming Yang. *Matrix Decomposition.* Northwestern University, Class Notes, 2000

[18] Tien-Hao Chang. *Numerical Analysis.* National Cheng Kung University, University Lecture, 2007

[19] Steven J. Leon. *Linear Algebra with Applications.* Pearson Prentice Hall, New Jersey, 8th edition, 2010.

[20] Ron Larson and David C. Falvo. *Elementary Linear Algebra.* Houghton Mifflin Harcourt Publishing Company, Massachusetts 6th edition, 2009.

[21] John H. Mathews and Kurtis D. Fink. *Numerical Methods Using MATLAB.* Pearson Prentice Hall, New Jersey 3rd edition, 1999.

[22] Morris Marden. *Geometry of Polynomials* American Mathematical Society, Rhode Island 2nd edition, 1969.