2020

Symbolic Construction of Matrix Functions in a Numerical Environment

Evan D. Butterworth
Georgia Southern University

Symbolic Construction of Matrix Functions in a Numerical Environment

An Honors Thesis submitted in partial fulfillment of the requirements for Honors in Mathematics.

By
Evan Butterworth
Under the mentorship of Dr. Yan Wu

ABSTRACT

Within the field of Computational Science, the importance of programs and tools involving systems of differential equations cannot be overemphasized. Many industrial sites, such as nuclear power facilities, are unable to safely operate without these systems. This research explores and studies matrix differential equations and their applications to real computing structures. Through the use of software such as MatLab, I have constructed a toolbox, or collection, of programs that will allow any user to easily calculate a variety of matrix functions. The first tool in this collection is a program that computes the matrix exponential, famously studied and presented by I.E. Leonard and Eduardo Liz. Currently, The Padé approximation is widely adopted to compute matrix exponentials. Unfortunately, this approximation yields significant errors given sufficiently large matrices or ill-conditioned matrices. This program is able to compute these matrix function values with little to no error, providing a distinct advantage over recursive-based methods. Utilizing the results from the matrix exponential, the next programs developed into the study and computation of logarithmic, sinusoidal, and tangential matrix functions. Many of these functions could have numerous uses in the study of dynamical systems that have applications in engineering models.

Thesis Mentor: ____________________________ _

Dr. Yan Wu

Honors Director: _______________________

Dr. Steven Engel

April 2020
Mathematical Sciences
University Honors Program
Georgia Southern University

TABLE OF CONTENTS

NOMENCLATURE

λ – Root of the characteristic polynomial or eigenvalue of a given matrix A

α – Real part of complex number $\alpha+\beta i$

β – Imaginary part of complex number $\alpha+\beta i$

n – Power of t in repeated eigenvalue case. Maximum of n is multiplicity-1

$\lambda(A)$ – The set of eigenvalues of matrix A

INTRODUCTION

In the scientific fields of applied mathematics and computational science, complex systems are often represented through the use of systems of differential equations. Frequently, the matrix exponential, e^{At}, arises in the solution to complex systems of differential equations where A is any square matrix. In addition, there are a number of dynamical systems where the matrix exponential comes into play. For example, the Control-Affine system:

$$\frac{dx}{dt} = Ax + f(x) + \beta(x)u, \; A \in \mathbb{R}^{n\times n}, \; x \in \mathbb{R}^{n}, \; \beta \in \mathbb{R}^{n\times 1}, \; f: \mathbb{R}^{n} \rightarrow \mathbb{R}^{n}$$

where u is a controller (scalar function). There are numerous engineering control systems represented by this type of differential equation. This includes spring pendulums, fluid dynamics, chemical reaction systems, and even has applications in robotics. The solution depends on the matrix exponential, which exists in both the transitive and steady state responses of the system. The computational accuracy of the matrix exponential is essential in order to analyze the performance of the Control-Affine systems. The commonly adopted computational algorithm for calculating the matrix exponential is the Padé approximation, for instance, the built-in function in MatLab. However, this approximation can yield an error that can propagate and grow to an inaccurate solution for some matrices. If we construct e^{At} using the Cayley-Hamilton theorem, the matrix function can be readily written in a finite form. This finite form brings up the interesting connection of deriving other matrix functions in a finite form, exploring the possibilities of logarithmic, sinusoidal, and tangential matrix functions.

THE MATRIX EXPONENTIAL

Before developing a program for the Matrix Exponential, it is critical to understand the definition of the matrix exponential as well as the advantages of a new method over the current status quo, specifically the Padé approximation. We know that the scalar exponential function can be defined using a power series expansion:

$$e^{at} = \sum_{k=0}^{\infty} \frac{a^k t^k}{k!} = 1 + at + \frac{a^2 t^2}{2!} + \cdots$$

The matrix exponential function can be defined similarly:

$$e^{At} = \sum_{k=0}^{\infty} \frac{A^k t^k}{k!} = I + At + \frac{A^2 t^2}{2!} + \cdots$$

Both of these series can be shown to converge to the scalar function or the matrix exponential uniformly. However, this definition for the matrix exponential presents quite the computational problem. Due to the infinite number of terms in the definition, using this in a computational environment would be extremely costly and inefficient. Before continuing, let the following properties hold:

1: $e^0 = e^{A0} = I$

2: $\frac{d}{dt} e^{At} = Ae^{At}$

Via the properties above, we can determine the solution to the initial value problem of the matrix differential equation:

$X' = Ax, X(0) = X_0$

$X(t) = e^{At} X_0$

From the similarities found in the power series expansion and the properties above, we can infer that the matrix and scalar exponential are closely related and may share even more characteristics. The general goal of this program is to compute the matrix exponential analytically or symbolically in a numerical environment. This will allow for the program to attain an extreme level of accuracy, while still maintaining a low computational cost and running in an efficient amount of time. Hence, we must construct a finite definition.

DEFINING THE MATRIX EXPONENTIAL

Theorem 1. Let A be a constant n x n matrix with characteristic polynomial:

$p(\lambda) = \det(\lambda I - A) = \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0$

so

$\Phi(t) = e^{At}$ is the unique solution to the nth order matrix differential equation

$\Phi^{(n)}(t) + c_{n-1}\Phi^{(n-1)} + \cdots + c_1\Phi' + c_0\Phi = 0$

satisfying the initial conditions

$\Phi(0) = I, \Phi'(0) = A, \Phi''(0) = A^2, \ldots, \Phi^{(n-1)}(0) = A^{n-1}$

Theorem 2. Let A be a constant n x n matrix with characteristic polynomial:

$p(\lambda) = \det(\lambda I - A) = \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0$

then

$e^{At} = x_1(t)I + x_2(t)A + x_3(t)A^2 + \cdots + x_n(t)A^{n-1}$

where

$x_i(t), 1 \leq i \leq n \sim x^{(n)} + c_{n-1}x^{(n-1)} + \cdots + c_1x' + c_0x = 0$

satisfying the following initial conditions

$L: \mathbb{R} \rightarrow \mathbb{R}^n$

$$L\big(x_i(0)\big) = \begin{bmatrix} x_i(0) \\ x_i'(0) \\ \vdots \\ x_i^{(n-1)}(0) \end{bmatrix} = e_i\big(i^{th} \text{ column of } I\big)$$

For additional details concerning the proofs of the two theorems above, consult the first reference presented in the references (Leonard 508-509). The above method provides a finite definition for the construction of the matrix exponential through the use of scalar initial value problems, which provides a significantly more accurate and efficient method from a computational perspective. The characteristic polynomial of the matrix is the key to determining the general form for all the $x_i(t)$. The roots of this polynomial, or the eigenvalues of the matrix denoted by λ, can be broken into four distinct cases: distinct real, repeated real, distinct complex, and repeated complex. As the $x_i(t)$ are solutions to a scalar homogeneous n^{th} order differential equation, the eigenvalues are the roots to the corresponding auxiliary equation (figure 1).

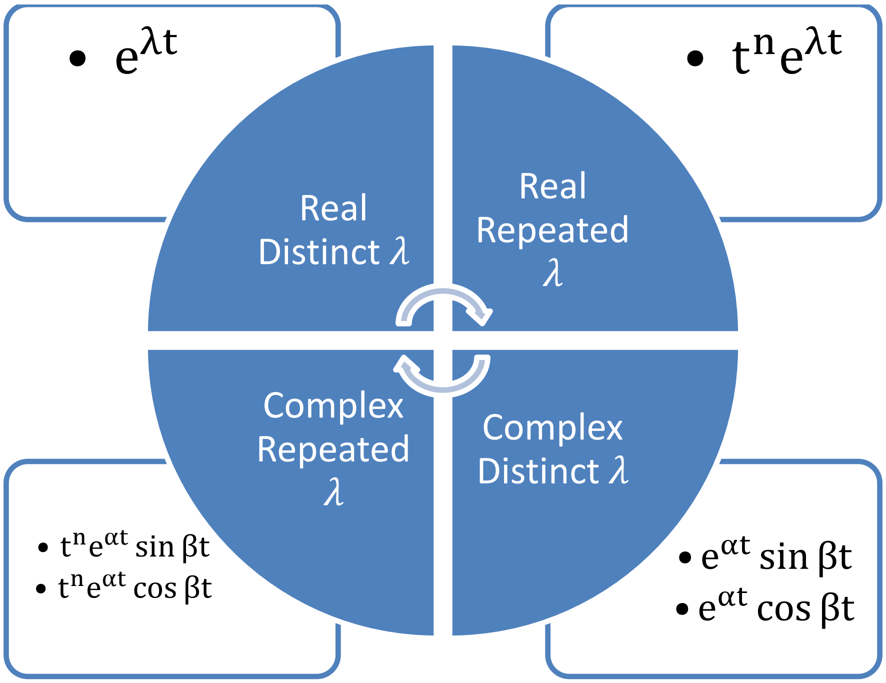

Figure 1

Consider the following example with undetermined coefficients c:

$$\begin{cases} \lambda_{1,2} = 2 \\ \lambda_3 = -1 \\ \lambda_{4,5} = 1 \pm i \end{cases} \Longrightarrow x_n(t) = c_{n,1}e^{2t} + c_{n,2}te^{2t} + c_{n,3}e^{-t} + c_{n,4}e^t \sin t + c_{n,5}e^t \cos t$$

Following this step, we would then need to use the initial values to solve for, in this case, all 25 undetermined coefficients for each $x_i(t), 1 \leq i \leq 5$.

COMPUTATION OF THE MATRIX EXPONENTIAL

A majority of this research was invested in the programming and computation of the matrix exponential. The developed program is broken down into five major steps. To begin with, the program must compute the eigenvalues of the given matrix, say A, and take into account the multiplicity of each eigenvalue. For ease, we separate the table of eigenvalues with respective multiplicities into two cases, real and complex. Next, we construct our general equation, say $X_n(t)$, based on the tables constructed in the first step. Note that we may take multiple higher-order derivatives of a variety of complex functions in a numerical environment. To maintain efficiency, the third overall step of the program is to calculate all necessary derivatives through the use of numeric subroutines (figure 2).

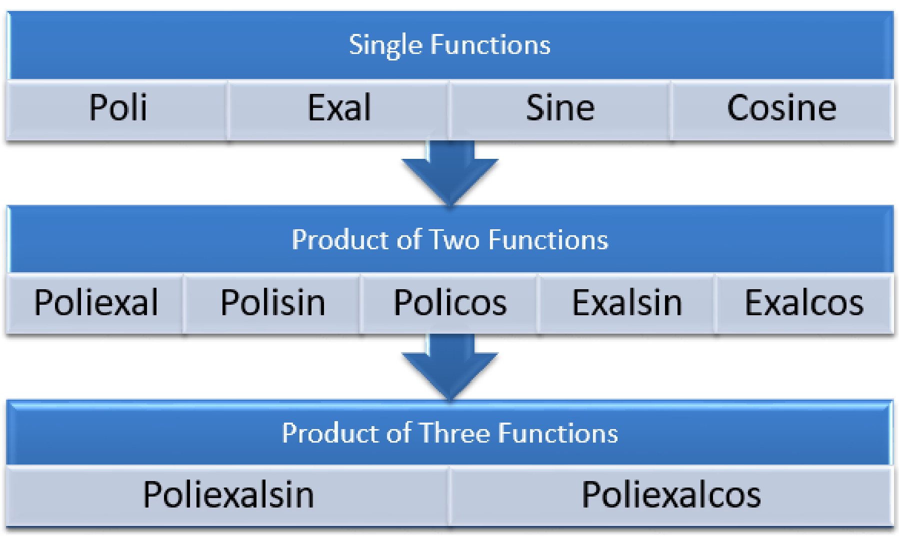

Figure 2

These eleven carefully-constructed subroutines have the ability to calculate any possible derivative presented in our general $X_n(t)$. In order to compute the derivative of a product of two functions, f and g, numerically, we utilize an interesting property. We find the solution by taking the anti-diagonal of a pascal matrix, of one dimension higher than the order of the derivative, as a row vector and multiplying it by a column vector composed of the necessary products of derivatives of f and g. Let A be a square pascal matrix with entries a_{ij} where i and j represent the row and column of A, respectively. The following property is shown below for the general case:

$$(fg)^{(m)} = [a_{m+1,1} \quad a_{m,2} \quad \cdots \quad a_{1,m+1}] \begin{bmatrix} f^{(m)}g^{(0)} \\ f^{(m-1)}g^{(1)} \\ \vdots \\ f^{(0)}g^{(m)} \end{bmatrix}$$

This tool serves as the foundation for the subroutines for the product functions, found in the second and third row in Figure 1. The significance of this tool cannot be understated, as it is an extremely useful method when computing high order derivatives of products of functions, both in terms of efficiency and program speed. Note that these particular subroutines also rely on the single function subroutines to calculate the derivatives of f and g. Hence, all of the single function programs contained in this collection are intertwined and nested in the double function subroutines. Likewise, the double function programs are utilized in the triple function programs. Together as a whole, this collection forms a complex web and essentially makes up the backbone of the main program. After the computation and storage of all necessary derivatives, we penultimately solve for the undetermined coefficients of each $X_n(t)$ using the initial values provided by the L operator as well as the necessary derivatives calculated previously. Utilizing the finite definition of the matrix exponential, we use the completed $X_n(t)$, from 1 to n, to finally compute the desired result. The figure below shows the computation of the matrix exponential for a 6x6 matrix:

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \qquad e^A = \begin{pmatrix} 7.04 & 6.53 & 8.15 & 4.14 & 9.54 & 4.03 \\ 2.17 & 4.11 & 2.79 & 3.08 & 2.85 & 7.89 \\ 3.41 & 4.74 & 7.04 & 3.18 & 7.09 & 4.16 \\ 3.37 & 3.16 & 4.74 & 2.73 & 4.35 & 6.64 \\ 5.54 & 4.55 & 7.53 & 3.10 & 9.92 & 3.73 \\ 2.56 & 1.80 & 3.36 & 2.03 & 4.33 & 8.99 \end{pmatrix}$$

ADDITONAL MATRIX FUNCTIONS

The matrix exponential can be utilized as the foundation for the development of other matrix functions. Consider the following matrix differential equation and note its general solution:

$$X'' + A^2X = 0,$$

$$X = c_1\, sin(At) + c_2 cos(At)$$

Comparable to the matrix exponential, note that the matrix only exists if A is square. The above equation implies the existence of the sinusoidal matrix functions. However, we can use the matrix exponential to compute these values in an efficient and precise manner utilizing the complex definitions of sine and cosine derived from Euler's identity. From the power series expansion of the matrix exponential, we can define the following:

$$e^{iAt} = \sum_{k=0}^{\infty} \frac{(iA)^k t^k}{k!} = I + iAt - \frac{A^2t^2}{2!} + \cdots$$

$$\sin(A) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} A^{2k+1} = \frac{e^{iA} - e^{-iA}}{2i}$$

$$\cos(A) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} A^{2k} = \frac{e^{iA} + e^{-iA}}{2}$$

Consider the following matrix:

$$A = \begin{pmatrix} 1 & 2 & 0 & 1 \\ -1 & 0 & 3 & 0 \\ 1 & -2 & 0 & 0 \\ 0 & 3 & 0 & 4 \end{pmatrix} \Rightarrow \cos(A) = \frac{e^{i\begin{pmatrix} 1 & 2 & 0 & 1 \\ -1 & 0 & 3 & 0 \\ 1 & -2 & 0 & 0 \\ 0 & 3 & 0 & 4 \end{pmatrix}} + e^{-i\begin{pmatrix} 1 & 2 & 0 & 1 \\ -1 & 0 & 3 & 0 \\ 1 & -2 & 0 & 0 \\ 0 & 3 & 0 & 4 \end{pmatrix}}}{2}$$

$$= \begin{pmatrix} 2.3423 & -2.2511 & -3.8084 & -0.6831 \\ -1.8550 & 8.4361 & 0.2657 & 0.7176 \\ -2.5611 & -1.4466 & 7.1667 & 0.1107 \\ 2.1527 & -4.3693 & -3.4763 & -1.1423 \end{pmatrix}$$

From these properties, we can see that the following properties, similar to their scalar counterparts hold for the zero matrix:

Let n be any integer including zero

$\sin(\mathrm{n}\pi\mathrm{I}) = 0$

$\cos(\mathrm{n}\pi I) = \pm I$

$\sin\left(I\left(\frac{\pi}{2} + \pi n\right)\right) = \pm I$

$\cos\left(I\left(\frac{\pi}{2} + \pi n\right)\right) = 0$

From the properties above, we can infer that the other commonly used angles in trigonometry will follow a similar pattern. For the purposes of computation, we have derived a finite method that is faster while attaining higher accuracy compared to the power series approximation with infinite terms. As an aside, the hyperbolic matrix sinusoids can also be defined in a similar way. Since the sinusoids are defined, the next natural connections are the tangential and reciprocal functions. Obviously, division by a matrix is not possible but we can derive the following functions, similar to their scalar counterparts:

$\tan(A) = \sin(A)\,(\cos(A))^{-1}$

$\cot(A) = \cos(A)\,(\sin(A))^{-1}$

$\sec(A) = (\cos(A))^{-1}$

$\csc(A) = (\sin(A))^{-1}$

From the properties above, we can see, for the zero matrix, that:

$\tan(0) = \sin(0)\,(\cos(0))^{-1} = 0I^{-1} = 0$

$\cot(0) = \cos(0)\,(\sin(0))^{-1} = I0^{-1} \Rightarrow undefined$

$\sec(0) = (\cos(0))^{-1} = I^{-1} = I$

$\csc(0) = (sin(0))^{-1} = \ 0^{-1} \Rightarrow$ undefined

We can see that the tangent and secant of matrix A are defined iff the quantity of cos(A) is invertible. Similarly, the cotangent and cosecant of matrix A are defined iff the quantity of sin(A) is invertible. For most of these trigonometric matrix functions, we can create a method to determine an approximation through the use of their respective power series expansions. Through these methods, we can see that these expansions do indeed converge to the definitions above. Next, we explore the inverse function of the matrix exponential. The matrix natural logarithm poses an interesting challenge as the scalar function exists over a strictly positive domain. The first instinct is to then claim that the matrix A must be positive definite in order to be defined. However, this condition is sufficient but not necessary for the existence of the function. Given a real matrix A, ln(A) is a real matrix iff $\lambda(A) \subseteq \mathbb{C}\backslash\mathbb{R}^{-}$. Equivalently, the set of eigenvalues of A contains no real negative elements. If a matrix has distinct and positive eigenvalues, it is diagonalizable. Hence the following property holds for diagonal matrices:

$\ln(A) = Pln(D)P^{-1}$

where D is the diagonal matrix of eigenvalues and P is the eigenvector matrix. For diagonal matrices, the natural log of the matrix is equivalent to the matrix of the natural log of each diagonal entry. For example, the natural log of the identity matrix is the zero matrix. Note that the exponential of the zero matrix is the identity matrix, so the inverse properties of the two functions hold. Consider the following examples:

1) $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \lambda_{1,2} = \pm i \Rightarrow \ln A = P\ln(D)P^{-1}$

$$= P\ln\begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix} P^{-1} = \begin{bmatrix} -i & i \\ 1 & 1 \end{bmatrix}\begin{bmatrix} \ln(i) & 0 \\ 0 & \ln(-i) \end{bmatrix}\begin{bmatrix} \frac{i}{2} & \frac{1}{2} \\ -\frac{i}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\pi}{2} \\ \frac{-\pi}{2} & 0 \end{bmatrix} = B$$

$$e^{\ln A} = e^{B} = e^{\begin{bmatrix} 0 & \frac{\pi}{2} \\ \frac{-\pi}{2} & 0 \end{bmatrix}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = A$$

2) $A = \begin{bmatrix} -4 & 1 & 5 \\ 3 & 2 & -3 \\ -9 & 1 & 10 \end{bmatrix}, \lambda_{1,2,3} = 1,2,5 \Rightarrow \ln A = P\ln(D)P^{-1}$

$$= Pln\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix} P^{-1}$$

$$= \begin{bmatrix} -0.4083 & 0.7071 & 0.5774 \\ 0.4083 & 0 & 0.5774 \\ -0.8165 & 0.7071 & 0.5774 \end{bmatrix}\begin{bmatrix} \ln(1) & 0 & 0 \\ 0 & \ln(2) & 0 \\ 0 & 0 & \ln(5) \end{bmatrix}\begin{bmatrix} 2.4495 & 0 & -2.4495 \\ 4.2426 & -1.4142 & -2.8284 \\ -1.7321 & 1.7321 & 1.7321 \end{bmatrix}$$

$$= \begin{bmatrix} -2.3026 & 0.6932 & 2.3026 \\ 0.9163 & 0.6932 & -0.9163 \\ -3.9120 & 0.6932 & 3.9120 \end{bmatrix} = B$$

$$e^{\ln A} = e^{B} = e^{\begin{bmatrix} -2.3026 & 0.6932 & 2.3026 \\ 0.9163 & 0.6932 & -0.9163 \\ -3.9120 & 0.6932 & 3.9120 \end{bmatrix}} = \begin{bmatrix} -4 & 1 & 5 \\ 3 & 2 & -3 \\ -9 & 1 & 10 \end{bmatrix} = A$$

Theorem 3. Let A be a square matrix. Then e^{A} is invertible.

Proof:

Important Property: Let A,B be *n* x *n* square matrices. Then

$e^{A+B} = e^A e^B$ iff $AB = BA$

so $e^A e^{-A} = e^{A-A} = e^0 = I$ by property 1.

Hence for any e^A, e^{-A} is its inverse. ■

Lastly, we can expand and apply the matrix exponential function to any positive scalar r using the power series expansion:

$$r^A = \sum_{n=0}^{\infty} \frac{(Aln(r))^n}{n!} = I + Aln(r) + \frac{(Aln(r))^2}{2!} + \cdots$$

In order to verify our approximation, we can use the fact that:

$$r^A = (e^{\ln(r)})^A = e^{Aln(r)}$$

An interesting property for diagonal matrices can be seen through the following example:

Let $r \in \mathbb{R}^+$

$$r^I = (e^{\ln(r)})^I = e^{I\ln(r)} = e^{\begin{pmatrix} \ln(r) & 0 & 0 \\ 0 & \ln(r) & 0 \\ 0 & 0 & \ddots \end{pmatrix}} = \begin{pmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & \ddots \end{pmatrix}$$

Hence, for any identity matrix of dimension n and positive scalar r, the above property holds.

CONCLUSIONS

Matrix functions, as a whole, are a useful tool, especially considering their applications to matrix differential equations as well as systems of differential equations. Through the use of Matlab, I have constructed a program that computes the matrix exponential utilizing the methods famously presented by I.E Leonard and Eduardo Liz, providing improvements in speed and accuracy compared to today's commonly used methods. This program involves the use of a number of functions, most importantly the collection of numerical derivative subroutines. Building on this tool, I have constructed a toolbox of programs that utilize the matrix exponential to compute various matrix functions, including the sinusoidal, tangential, logarithmic, and general exponential functions. Future research may branch into even more matrix functions as well as developing additional applications in more interdisciplinary fields.

References

1) Leonard, I. E. “The Matrix Exponential.” *SIAM Review*, vol. 38, no. 3, Sept. 1996, pp. 507–512., doi:10.1137/s0036144595286488.

2) Liz, Eduardo. “Classroom Note:A Note on the Matrix Exponential.” *SIAM Review*, vol. 40, no. 3, Sept. 1998, pp. 700–702., doi:10.1137/s0036144596320752.