On the Accuract and Efficient Approximation of Matrix Exponentials

Emily H. Huang

2023.9-2024.5

A senior honors thesis submitted to the faculty of the University of North Carolina at Chapel Hill in fulfillment of the requirements for the Honors Carolina Senior Thesis in the Department of Mathematics.

Abstract

Abstract Here.

Acknowledge

I extend my sincere gratitude to Professor Yifei Lou. As my research mentor in the UNC Department of Mathematics, she has provided me with insightful feedback, expert guidance, and invaluable support that have been indispensable to the success of my thesis. Her dedication, patience, and encouragement have been a constant source of inspiration and have greatly shaped the quality of my work. She has patiently answered my countless questions, helping me to understand complex theorems and concepts. Her unwavering support and guidance have not only helped me in my academic pursuits but also in my personal life. I feel incredibly fortunate to have had the opportunity to work with such an exceptional mentor, and I will always be grateful for her mentorship.

I also want to express my gratitude to my colleagues who have supported me during my academic journey. Your enthusiasm, advice, and encouragement have been instrumental in keeping me motivated and on track. I am grateful for our discussions, where we share ideas, compare our numerical results, share research resources, and discuss any problems we encounter. The resources and facilities provided by the University of North Carolina at Chapel Hill have also been crucial to the success of my research.

Lastly, I owe a debt of gratitude to my family and friends for their unwavering love, support, and understanding during this rigorous journey. Their encouragement has been invaluable, and I am grateful for their contributions to this work. Thank you for making this journey both meaningful and rewarding.

Contents

1 Introduction

1.1 Examples of Motivation

Figure 1: How to add a picture

1.2 Research Questions

Figure 2: Dependent data

Figure 3: Independent data

1.3 Scientific Challenges

1.4 Explorations and Contributions

2 Matrix Exponential and Operator Splitting

Many physical processes can be decomposed as the combination of simple processes, e.g., many biological processes are modeled by the reaction-diffusion models, and the Navier-Stokes equation contains advection, rection, and diffusion processes. Each process is well studied both analytically and numerically, the challenging research questions are how different components interact/compete with each other to produce complex but interesting phenomena? And how the complex process can be simulated using existing efficient tools built for each individual and well-studied process.

In this research, we consider a matrix version of the challenges.

Assume a physical process is modeled by the exponential of matrix e^{-Ht} where H is a size

$m \times m$ matrix containing contributions from different physical processes and t is the scalar time parameter. Assume H can be decomposed as the sum of two simple matrices

$$H = V + T$$

where V is a low-rank matrix and T is a diagonal matrix. Clearly, both e^{-Vt} and e^{-Tt} are easy to calculate, as

Theorem 1. *The exponential of a diagonal matrix is a diagonal matrix.*

Theorem 2. *The exponential of a low-rank matrix is the Identity matrix plus a low-rank matrix.*

Unfortunately, because matrix multiplication is not commutative, in general

$$e^{-Ht} \neq e^{-Tt}e^{-Vt}.$$

There are different approaches to utilize the simple structures in V and T to compute the more complex e^{-Ht}. One such approach is to use the product of matrix exponentials to achieve higher order approximation of e^{-Ht} []. In this thesis, we explore a polynomial based approach and approximate

$$e^{-Ht} \approx p_n(Ht)$$

where $p_n(x)$ is a polynomial of degree n. Note that as $Y(t) = e^{-Ht}$ satisfies the ordinary different equations

$$\begin{cases} Y'(t) = -H \cdot Y(t) \\ Y(0) = I_{m \times m} \end{cases} \tag{1}$$

As there exist no polynomial with a bounded degree that exactly satisfy the differential equation, we therefore search for a degree n polynomial that satisfies a pseudo-spectral (collocation) formulation by requiring the differential equation is exactly satisfied at $n+1$ collocation points.

Pseudo-spectral Formulation

For a given set of collocation points $\{t_1, t_2, \cdots, t_{n+1}\}$, the pseudo-spectral formulation finds a polynomial matrix $p_n(t)$ which satisfies

$$\begin{cases} p_n'(t_j) = -H \cdot p_n(t_j) \\ p_n(\cdot 0) = I_{m\times m} \end{cases} \tag{2}$$

Unfortunately, the spectral differentiation is an ill-conditioned operator, as demonstrated by the following numerical experiments.

(Pseudo-)Spectral differentiation is ill condition

Problem setting: Giving the function values $f(x_j)$ at $\{t_1, t_2, \cdots, t_{n+1}\}$, one can construct an interpolating polynomial $p_n(x)$. Differentiate the polynomial, one can approximation $f'(x) \approx p_n'(x)$. In this experiment, we compare the analytical $f'(x)$ with its numerical approximation $p_n'(x)$ at the collocation points $\{t_1, t_2, \cdots, t_{n+1}\}$.

The code is written using Mathematica and is available at .

3 Integral Equation Reformulation for Matrix Exponential Problem

Unlike the spectral differentiation operation discussed in the previous section, the spectral and pseudo-spectral integration operators are numerically very stable, as deomonstrated by the following experiment.

(Pseudo-)Spectral integration operator is well-conditioned

Problem setting: Giving the function values $f(x_j)$ at $\{t_1, t_2, \cdots, t_{n+1}\}$, one can construct an interpolating polynomial $p_n(x)$. Differentiate the polynomial, one can approximation $\int_{t=0}^{x} f(t)dt \approx$

$\int_0^x p_n'(t)dt$. In this experiment, we compare the analytical anti-derivative $\int f(x)$ with its numerical approximation $\int p_n(x)$ at the collocation points $\{t_1, t_2, \cdots, t_{n+1}\}$.

The code is written using Mathematica and is available at .

Therefore instead of the original differential equation formulation, we consider the following Picard integral equation reformulation

$$\begin{cases} Y(x) = Y(0) + \int_0^x (-H) \cdot Y(t)dt \\ Y(0) = I_{m\times m} \end{cases} \tag{3}$$

and searching for a polynomial matrix $p_n(t)$ that satisfies the discretized pseudo-spectral formulation

$$\begin{cases} p_n(t_j) = p_n(0) + \int_0^{t_j} (-H) \cdot p_n(\tau)d\tau \\ p_n(0) = I_{m\times m} \end{cases} \tag{4}$$

4 Spectral Deferred Correction Methods

To solve the pseudo-spectral formulation 4, we apply the spectral deferred correction (SDC) approach to improve the efficiency of the algorithm.

Step 1: The first step of the SDC is to find an approximation polynomial solution $\tilde{Y}(t)$ using a low-order method. To demonstration the ideas, we simply apply the first order method from matrix exponentials and compute $\tilde{Y}(t)$ as follows.

$$\tilde{Y}(0) = I_{m\times m}.$$

$$\tilde{Y}(t_{j+1}) = e^{-T(t_{j+1}-t_j)} e^{-V(t_{j+1}-t_j)} \tilde{Y}(t_j)$$

Comment: Instead of the first order approximation, one can also apply higher order approxi-

mations, e.g., the 2nd order Strang spliting. In Rachel's work, many such higher order splittings are developed. An interesting question is how different "low-order" predictors will change the efficiency of the algorithm, which will be studied in some details in this thesis.

Once the approximate solution $\tilde{Y}(t)$ is available and define $Y(x) = \tilde{Y}(x) + \delta(x)$, plug in the Picard integral equation

$$\begin{cases} \tilde{Y}(x) + \delta(t) = Y(0) + \int_0^x (-H) \cdot (\tilde{Y}(x) + \delta(t))dt \\ \delta(0) = 0_{m\times m} \end{cases} \tag{5}$$

one can derive a new set of equations for the error (also called defect) $\delta(t)$

$$\begin{cases} \delta(t) = \int_0^x (-H) \cdot \delta(t)dt + \left(Y(0) + \int_0^x (-H) \cdot \tilde{Y}(x)dt - \tilde{Y}(x)\right) \\ \delta(0) = 0_{m\times m} \end{cases} \tag{6}$$

Defining the residue

$$\epsilon(x) = Y(0) + \int_0^x (-H) \cdot \tilde{Y}(x)dt - \tilde{Y}(x),$$

the error's equation becomes a new Picard integral equation

$$\begin{cases} \delta(t) = \int_0^x (-H) \cdot \delta(t)dt + \epsilon(t) \\ \delta(0) = 0_{m\times m} \end{cases} \tag{7}$$

Note that $\tilde{Y}$ is known, the integral $\int_0^x (-H) \cdot \tilde{Y}(x)dt$ can be accurately evaluated using the very high order (and stable) pseudo-spectral integration matrix.

Step 2: The second step of the SDC method is to apply a low-order method to get a low-order estimate $\tilde{\delta}(t)$ of the analytical error (or defect) $\delta(t)$. Define stepsize $h_j = t_j - t_{j-1}$, $h_0 = t_1 - 0$, and approximate the integral using the Trapezoidal rule, we have

$$\tilde{\delta}(t_1) = (-H)(\frac{h_1}{2}\tilde{\delta}(t_1)) + \epsilon(t_1),$$

$$\tilde{\delta}(t_k) = (-H)\left(\sum_{j=1}^{k} \frac{h_j}{2}(\tilde{\delta}(t_j) + \tilde{\delta}(t_{j-1}))\right) + \epsilon(t_j).$$

Moving all the terms with $\tilde{\delta}(t_k)$ to the left, at each time step, one needs to solve the matrix equation system

$$(I + \frac{h_k}{2}H)\tilde{\delta}(t_k) = RHS\,(t_k) \tag{8}$$

where all the known terms are collected in the term $RHS\,(t_k)$.

To design a low order method which solves Eq. (8) efficiently, we consider the following approximation

$$(I + \frac{h_k}{2}H)^{-1} \approx I - \frac{h_k}{2}H \approx e^{-\frac{h_k}{2}H}.$$

Therefore, the same low-order time splitting schemes can be applied.

Note that the previous approach is only first order accurate, to investigate possible higher order schemes, we consider the differential equation form of the Picard integral equation

$$\begin{cases} \delta'(t) = -H \cdot \delta(t) + \epsilon'(t) \\ \delta(0) = I_{m\times m} \end{cases} \tag{9}$$

The analytical solution is given by

$$\delta(t_{j+1}) = e^{-Ht}\delta(t_j) + \int_{t_j}^{t_{j+1}} e^{-H(t-\tau)}\epsilon'(\tau)d\tau.$$

Applying integration by parts, we have

$$\delta(t_{j+1}) = e^{-Hh_{j+1}}\delta(t_j) + e^{(}-H(t-\tau))\epsilon(\tau)|_{\tau=t_j}^{t_{j+1}} + \int_{t_j}^{t_{j+1}} He^{-H(t-\tau)}\epsilon(\tau)d\tau$$

where $h_{j+1} = t_{j+1} - t_j$. Therefore

$$\delta(t_{j+1}) = e^{-Hh_{j+1}}\delta(t_j) + \epsilon(t_{j+1}) - e^{-Hh_{j+1}}\epsilon(t_j) + \int_{t_j}^{t_{j+1}} He^{-H(t-\tau)}\epsilon(\tau)d\tau.$$

A higher order (but not spectral order) quadrature rule can be applied to evaluate $\int_{t_j}^{t_{j+1}} He^{-H(t-\tau)}\epsilon(\tau)d\tau$. When the trapezoidal rule is applied, the update formula becomes

$$\delta(t_{j+1}) = e^{-Hh_{j+1}}\delta(t_j) + \epsilon(t_{j+1}) - e^{-Hh_{j+1}}\epsilon(t_j) + \frac{h_{j+1}}{2}(He^{-Hh_{j+1}}\epsilon(t_j) + H\epsilon(t_{j+1})).$$

In the numerical implementation, an exponential expansion based low order method can be applied to approxiate $e^{-Hh_{j+1}}$ and be evaluated efficiently.

For the given approximate solution $\tilde{Y}$, the final outcome (output) is the low-order approximation of the error (defect) $\tilde{\delta}(t)$. This can be explictly represented as

$$\tilde{\delta}(t) = ImpFun(\tilde{Y}).$$

Step 3: There are two different approaches to continuous improve the approximate solution. In the first approach, a better estimate of the approximate solution is simply

$$\tilde{Y}_{new} = \tilde{Y}_{old} + \tilde{\delta}$$

and one can repeat step 2 until the iterations are convergent or a maximum number of iterations is reached. In numerical linear algebra language, this fixed-point (stationary) iterations represent a particular Neumann series expansion. The resulting algorithm is referred to as the spectral deferred correction (SDC) method in existing literature.

In the second approach, instead of a naive Neumann series expansion, one can use the terms in the Neumann series to construct a Krylov subspace and search for the optimal solution in the Krylov subspace. The resulting algorithms are well-studied by the numerical linear algebra community. Instead of a detailed review of the mathematical foundation and existing implementations, we refer interested readers to []. In our implementation, as most matrices are in general non-symmetric, existing GMRES, restarted GMRES, Transpose-free QMR, and

BiCGStab have been used to improve the convergence properties of the SDC approach caused by a few bad eigenvalues. The resulting algorithm is referred to as the Krylov deferred correction method (KDC) []. The implementation of KDC is a simple application of existing Krylov subspace methods to the linear equation system $\tilde{\delta}(t) = ImpFun(\tilde{Y})$ where the matrix vector multiplication result is given by $\tilde{\delta}(t)$ and one is search for the root of $ImpFun(Y) = 0$, i.e., when the pseudo-spectral solution becomes the input of the implicit function, the output low order solution should be $\tilde{\delta}(t) = 0$.

4.1 SDC vs. KDC

The advantage of the SDC method is that one only needs the results from previous step (not all historical data) in order to start a new round of refinement, therefore requiring minimal storage. This is normally the right choice when there are no convergence issues (due to a few bad eigenvalues in the Neumann series expansion).

When there are bad eigenvalues, Neumann series may converge slowly (order reduction) or become divergent. In this case, by searching for the optimal solution in the Krylov subspace using least squares, the KDC method will converge more efficiently once the bad eigenvalues are fully resolved. However, this approach requires additional operations and the storage of historical data.

Finding the “optimal" method for a particular problem is always a challenging research topic and the answer depends on a lot of factors, including both the problem properties and computer hardware resources. Some sample factors one needs to consider in the problem discussed in this thesis include the stepsize, the eigenvalue distributions of H, V, and T, the accuracy and efficiency of the low-order scheme, the special structures (low-rank, symmetry) in the matrices, and a lot more.

5 Numerical Experiments

6 Future work