Two Quantum Algorithms for a Non-linear Reaction-Diffusion Equation using Chebyshev approximation method

A Thesis

Submitted For the Degree of

Master of Technology

in the Faculty of Engineering

by

Manish Kumar

Instrumentation and Applied Physics

Indian Institute of Science

BANGALORE – 560 012

June 2024

Manish Kumar

June 2024

TO

All the person in my life
who encouraged me to learn to catch a FISH
rather than simply asking for a FISH.

Acknowledgements

I acknowledge that the code written by **Herman Kolden** (for viscous Berger's equation) has been repurposed to reproduce the results on the forward Euler method.

I thank (**IQTI, IISc Bangalore**) for providing me with the computing facility for the numerical simulation.

I firmly believe the continuous feedback and support of Prof. **Apoorva D. Patel** have influenced this project in a significant proportion. □

Abstract

We present **two** quantum algorithms for reaction-diffusion equations(RDE) based on a truncated Chebyshev series method. In the first algorithm, we adapt the quantum algorithm for matrix exponentiation using Chebyshev expansion (Patel et al. 2017) [PP17]. For the second algorithm, we repurpose the Quantum pseudo-spectral method to solve an ODE (Childs et al. 2019) [CL19].

We employ the Carleman embedding technique to linearize the non-linear equation in both cases. Additionally, we derive a sufficient condition for the Carleman embedding matrix to be diagonalizable, which is indispensable for designing both quantum algorithms. We give an iterative procedure to construct the similarity transformation matrix (say, V) that diagonalizes the Carleman matrix (i.e., $A = V\Lambda V^{-1}$).

Our first algorithm has gate complexity of $O(d\ \log(d),\ T\ \text{polylog}(T),\ \text{poly}(\log(\frac{1}{\varepsilon})))$. Where d is the size of the Carleman matrix, T is simulation time, and ε is the approximation error. The distinct feature of the first algorithm is that it doesn't rely on any Quantum linear system algorithm (QLSA).

The second algorithm achieves a nearly optimal dependency on parameters d, T, and ε. The gate complexity asymptotically scales as $O(\text{polylog}(d),\ T\ \text{polylog}(T),\ \text{polylog}(\frac{1}{\varepsilon}))$. We further compare our algorithms with two existing methods for the Reaction-Diffusion equation: the forward Euler method (Liu et al. 2023) [Liu23] and the truncated Taylor series method (Costa et al. 2023) [CSM23].

We have considered a specific one-dimensional quadratic RDE throughout the paper. It is mainly for convenience in numerical simulation and theoretical analysis. We briefly discuss how it can be generalized to cubic RDE.

Our algorithms produce a quantum state vector that encodes the final solution. It is less informative than an explicit solution. It is a well-known caveat to all the known quantum algorithms for this problem [DLW22]. However, we briefly discuss some advantages of pursuing this goal. We conclude this by summarizing the complexity-theoretic hardness of the problem and why the algorithms can't be *dequantized* if conjecture $BQP \neq P$ is true. □

Organization of the content & Our contribution

In **Chapter 1**, we briefly introduce the problem and its scope.

In **Chapter 2**, we formally describe the problem for which we seek an algorithm.

In **Chapter 3**, we briefly summarize the two previous algorithms for the problem. We explicitly describe their method and key results while omitting the proof.

In **Chapter 4**, we summarize the analytic results on the stability of the numerical solution. It includes errors due to Carleman truncation, central difference discretization, and numerical integration schemes (Euler and Taylor). These results are borrowed from prior work. We utilize our MATLAB codes for numerical simulations to validate them.

In **Chapter 5**, we present our analysis of the diagonalization of the Carleman matrix.

In **Chapter 6**, we utilize the results of the previous chapter to explore the Chebyshev series-based method to solve the Carleman ODE emerging from the Reaction-diffusion equation. We repurpose two known quantum methods to solve the Carleman ODE.

In **Chapter 7**, we briefly mention the complexity-theoretic hardness of the problem and its known lower bounds. We compare the gate complexity of all the algorithms described in the paper. We conclude with some open problems related to the new algorithms that can be pursued in the future. □

All the simulation results of this paper can be found in our GitHub repository [here].

Contents

Chapter 1

Introduction

1.1 Quantum algorithms for linear ODEs

Quantum Hamiltonian simulation has been considered a promising avenue for achieving a computational advantage over existing classical simulation methods. [Fey82] [Lyo96]. The problem can be viewed as solving the Schrodinger equation, a linear ordinary differential equation that governs the dynamics of closed quantum systems.

ODEs modeling classical systems are ubiquitous in science and engineering. Several classical numerical techniques have been developed over a century [Kre98]. Recently, numerous efforts have been made to design a quantum algorithm for some classical systems [BCO17][DLW22]. Contrary to the classical algorithms, these quantum algorithms output a quantum state vector that encodes the solution instead of explicitly describing the state vector.

1.2 Quantum algorithms for Non-linear ODEs

Several natural systems, such as chemical reactions, fluid transport, and population dynamics, are well explained by non-linear equations. For example, the Reaction-Diffusion equation (RDE) for chemical systems and the Navier-Stokes equation (NSE) for fluid flow are widely studied in non-linear systems.

As per the finite difference method, spatial discretization of the above two PDEs will give non-linear ODEs [GRM07]. Recently, this has been viewed as an opportunity to explore quantum algorithms for such non-linear systems by designing an algorithm for a non-linear ODE. However, the quantum algorithmic toolkit developed for linear ODEs could not be directly imported to non-linear ODEs, mostly because quantum dynamics is a linear theory. Nevertheless, **a special class** of non-linear ODEs

for a certain range of parameters has been efficiently computed on a quantum machine [Liu21]. These non-linear ODEs have polynomial non-linearity. For example, the below non-linear ODE has quadratic non-linearity:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = au + bu^2 \tag{1.2.1}$$

An efficient quantum algorithm exists for a specific range of parameters, a and b. It is achieved by Carleman linearization, a technique that embeds a non-linear ODE into an infinite system of linear ODEs. Now, the linear system of ODE can be solved using existing quantum methods.

1.3 Reaction-Diffusion Equation (RDE)

RDE successfully models various biological phenomena such as leaf venation, tumor growth, and Turing patterns in tissues and organs [Gri96]. A one-dimensional RDE is mathematically represented as below:

$$\frac{\partial u(x,t)}{\partial t} = D\Delta u(x,t) + f(u(x,t)) \tag{1.3.1}$$

Here, $D \in \mathbb{R}$ is called the diffusion coefficient. The term $f(u(x,t))$ is called the reaction term, accounting for all local reactions. Three of its special cases are:

1. $f(u) = au + bu^2$ is called the KPP–Fisher equation.
2. $f(u) = au + bu^3$ is called Allen-Kahn equation.
3. $f(u) = au^2 + bu^3$ is called Zeldovich equation.

Throughout the paper, we take the KPP-Fisher equation as a prototypical problem to design both the algorithm and demonstrate its effectiveness via numerical simulation.

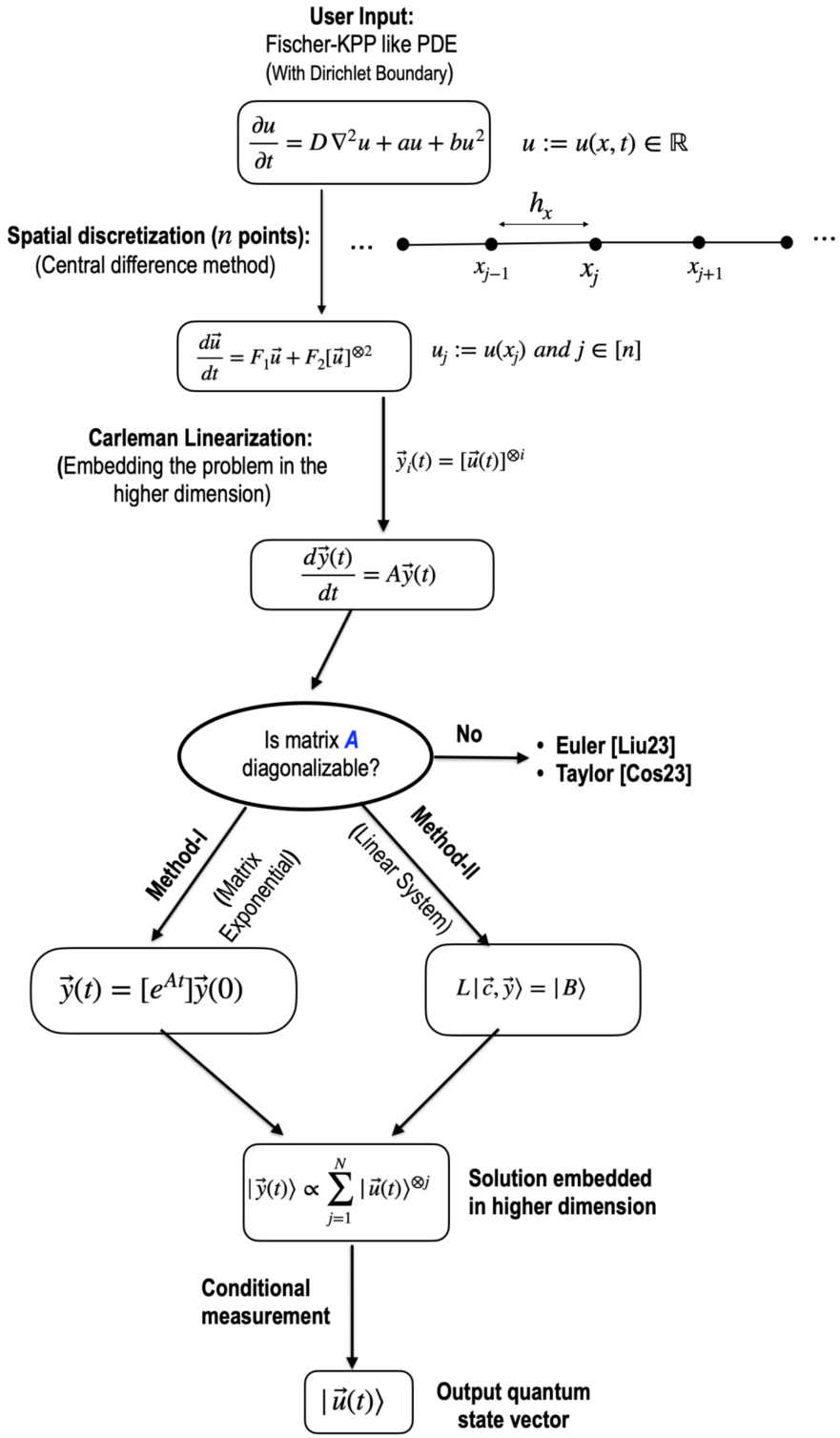

Figure 1.1: Algorithmic steps to produce a quantum state encoding the solution of Fisher-KPP equation. Like previous works, we utilize uniform discretization and Carleman linearization techniques. We have investigated the possibility and advantage of using the Chebyshev series method to solve the homogeneous ODE arising from Carleman linearization.

Chapter 2

Problem Description

2.1 The Quadratic ODE Problem

2.1.1 The equation and its parameter description

We focus on an initial value problem described by the one-dimensional quadratic ODE.

$$\frac{\mathrm{d}u}{\mathrm{d}t} = F_2 u^{\otimes 2} + F_1 u, \qquad u(0) = u_{\mathrm{in}}. \tag{2.1.1}$$

Here $u = [u_1, \ldots, u_n]^T \in \mathbb{R}^n$, $u^{\otimes 2} = [u_1^2, u_1u_2, \ldots, u_1u_n, u_2u_1, \ldots, u_nu_{n-1}, u_n^2]^T \in \mathbb{R}^{n^2}$, each $u_j = u_j(t)$ is a function of t on the interval $[0, T]$ for $j \in [n] \coloneqq \{1, \ldots, n\}$, $F_2 \in \mathbb{R}^{n \times n^2}$, $F_1 \in \mathbb{R}^{n \times n}$ are time-independent matrices. The symbol $\|\cdot\|$ denotes the spectral norm unless stated otherwise.

2.1.2 Problem Description

The main computational problem we consider is as follows.

Problem 1 *([Liu21], [CSM23]) We consider a one-dimensional quadratic ODE as in (2.1.1). We assume F_2, and F_1 are s-sparse (i.e., have at most s nonzero entries in each row and column), F_1 is diagonalizable, and that the eigenvalues λ_j of F_1 satisfy $\mathrm{Re}\,(\lambda_n) \leqslant \cdots \leqslant \mathrm{Re}\,(\lambda_1) < 0$. We parametrize the problem in terms of the quantity*

$$R \coloneqq \frac{\|u_{\mathrm{in}}\|\|F_2\|}{|\,\mathrm{Re}\,(\lambda_1)|}. \tag{2.1.2}$$

We assume we are given oracles O_{F_2} and O_{F_1} that provide the locations and values of the nonzero entries of F_2 and F_1, respectively, for any desired row or column. We are also given the value $\|u_{\mathrm{in}}\|$ and an oracle O_x that maps $|00\ldots0\rangle \in \mathbb{C}^n$ to a quantum state proportional to u_{in}. We aim to produce

an ε-approximate quantum state proportional to $u(T)$ (say, $|u(T)\rangle$) for some given $T > 0$ within some prescribed error tolerance $\epsilon > 0$.

Two variants of the problem: A variant of problem-1 exists. It is mentioned in [Liu23] and asks for the solution vector $u(t)$ value at different time steps between $t \in [0,\ T]$. Assume the time domain has m equally spaced points such that $h = T/m$.

Figure 2.1: Uniform time steps between $[0,\ T]$

Now, the problem is formally defined as:

Problem 2 *([Liu23]) For the problem-1, instead of producing quantum state $|u(T)\rangle$, produce a quantum state that is a superposition of the solution at different timesteps as follows*

$$|u_{\text{evo}}\rangle = u(0)\,|0\rangle + u(h)\,|1\rangle + u(2h)\,|2\rangle + \cdots + u(mh)\,|m\rangle = \sum_{k=0}^{m} u(kh)|k\rangle \tag{2.1.3}$$

It is slightly different than problem 1. We use the following terminology to distinguish the problem:

- *Problem-1*: Final state (quantum) encoding problem
- *Problem-2*: History state (quantum) encoding problem

In Chapter 3, we will see that the algorithms for both problems are almost identical. They only differ at the final step of the algorithm, called the conditional measurement step. In Chapter 5, we work on problem 1 to design both algorithms.

Interpretation of parameter R: The parameter R defined in problem 1 plays a pivotal role in designing an efficient algorithm for the problem. Liu et al. have interpreted the quantity $R = \frac{\|u_{\text{in}}\|\|F_2\|}{|\operatorname{Re}(\lambda_1)|}$ as qualitatively similar to the Reynolds number, which characterizes the ratio of the (non-linear) convective forces to the (linear) viscous forces within a fluid. More generally, R quantifies the strength of the non-linearity relative to dissipation.

2.2 Reduction of Fischer-KPP equation to the Quadratic ODE

We focus on the specific Reaction-Diffusion equation called the Fischer-KPP equation:

$$\frac{\partial u}{\partial t} = D\Delta u + au + bu^2, \tag{2.2.1}$$

where, $u := u(x,t)$. Let the domain of the problem be $x \in [0,1]$ and $t \in [0,T]$. It is a non-linear PDE. On spatial discretization, we get a non-linear ODE. Here, we employ the central difference scheme to discretize the spatial domain as below.

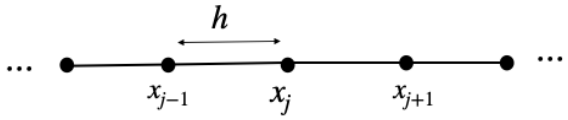

Figure 2.2: figure caption

On defining $u_j := u(x_j,t)$ for $j \in [n]_1$, we have $u = [u_1,\ldots,u_n]^T \in \mathbb{R}^n$ and $u^{\otimes 2} = [u_1^2, u_1u_2,\ldots,u_nu_{n-1},u_n^2]^T \in \mathbb{R}^{n^2}$. The result of the discretization is a system of non-linear ODEs as below:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = DL_hu + au + \mathbf{b}u^{\otimes 2}, \tag{2.2.2}$$

Here, L_h is the one-dimensional discrete Laplacian operator. For homogeneous Dirichlet boundary conditions, L_h is

$$L_h := (n+1)^2 \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}_{n\times n}. \tag{2.2.3}$$

Equation (2.2.2) can be equivalently written as:

$$\frac{\mathrm{d}U}{\mathrm{d}t} = (DL_h + aI)u + \mathbf{b}u^{\otimes 2}, \tag{2.2.4}$$

Comparing it with (2.1.1) implies

$$F_1 = (DL_h + aI) \in \mathbb{R}^{n\times n}$$

and

$$F_2 = \mathbf{b} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n^2}}$$

The above analysis implies that a numerical scheme for the above Quadratic ODE can be used to predict the temporal dynamics of the Fisher-KPP equation. The rest of the document contains different

quantum algorithms to solve it

2.2.1 Properties of matrix F_1 and F_2

Lemma 1 *The following two properties hold for matrix* F_1

1. *It is a real symmetric matrix, thus Hermitian.*
2. *It has real and negative eigenvalues only if parameters* D, a, *and* n *are related by*

$$4D(n+1)^2 sin^2(\frac{j\pi}{2(n+1)}) \geq a \tag{2.2.5}$$

Proof: The eigenvalues of L_h are given by

$$\lambda_j(L_h) = -4(n+1)^2 sin^2(\frac{j\pi}{2(n+1)});\ where\ j \in \{1, ..., n\} \tag{2.2.6}$$

Since $F_1 = DL_h + aI$, its eigenvalues are given by

$$\lambda_j(F_1) = -4D(n+1)^2 sin^2(\frac{j\pi}{2(n+1)}) + a;\ where\ j \in \{1, ..., n\} \tag{2.2.7}$$

Thus $\lambda_j(F_1) \leq 0$ if

$$4D(n+1)^2 sin^2(\frac{j\pi}{2(n+1)}) \geq a \tag{2.2.8}$$

Usually, the diffusion coefficient $D \geq 0$. Hence, $a < 0$ makes the above inequality true unconditionally. But certain values of $a > 0$ are too allowed, as given by the above equation. □

Another interesting property is possessed by matrices F_1 and F_2. They are sparse matrices whose sparsity parameter (say, s) is independent of their size, or $s = O(1)$.

Lemma 2 *Matrix* F_1 *has sparsity parameter* $s = 3$, *independent of its size* n. *While for* F_2, *sparsity* $s = 1$.

Proof: The definition of sparsity parameter s is a maximum number of non-zero entries along rows or columns. For F_1, $s = 3$ due to the construction of discrete laplacian matrix L_h. For F_2, it is due to the construction prescribed by its definition. □

Matrix sparsity plays an important role in designing an efficient quantum algorithm. We use these results in the later parts.

Chapter 3

Existing methods to solve the Quadratic ODE: Survey

One of the most common strategies to explore a quantum algorithm for this problem can be summarized as a three-stage process:

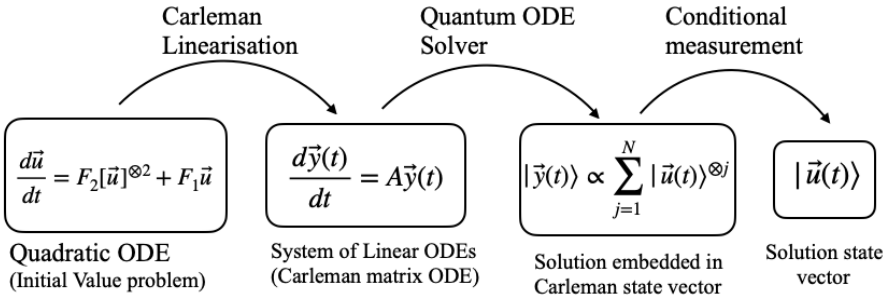

Figure 3.1: Three key steps to solve the quadratic ODE problem. Different algorithms attempt to solve the Linear ODE (step 2) differently.

1. Linearize the quadratic ODE into a system of linear ODEs.
2. A quantum algorithm will solve the above linear ODE system.
3. Apply conditional measurement on the output of the previous step. It gives the desired state vector.

For quantum algorithms, mainly two linearization schemes have been explored, namely- (I) Carleman and (II) Lie-Koopman linearization. We exclusively utilize the Carleman linearization method in the paper.

As we see in the next section, the linearization gives a matrix differential ODE as below:

It is more concisely represented as

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = A\mathbf{y}, \qquad \mathbf{y}(0) = \mathbf{y}_{\text{in}} \tag{3.0.1}$$

Two previously explored algorithms to solve the above equation are- (I) the Euler method and (II) the Truncated Taylor method. We briefly highlight these algorithms in section 3.2.

Two algorithms exist to solve the ODE via the Chebyshev approximation method: (I) Matrix exponentiation and (II) pseudo-spectral method. Unlike Euler and Taylor's methods, matrix A is required to be diagonalizable. Our key contribution is to show that the matrix is diagonalizable for a wide range of parameters. Then, we repurpose the existing Chebyshev-based ODE solver to solve the Fisher-KPP problem.

3.1 Carleman Linearization for Quadratic ODE

Now, we elaborate on Carleman's linearization of the quadratic ODE problem. Given

$$\frac{\mathrm{d}u}{\mathrm{d}t} = F_1 u + F_2 u^{\otimes 2}, \qquad u(0) = u_{\text{in}}. \tag{3.1.1}$$

where, $u := \vec{u}(t) \in \mathbb{R}^n$, $F_1 = (DL_h + aI) \in \mathbb{R}^{n\times n}$ and $F_2 = \mathbf{b} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n^2}}$.

The Carleman embedding is given by initializing a new set of variables $y_j := u^{\otimes j}$ for $j = 1, 2\cdots\infty$. It provides a system of infinite linear ODE in new variables:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_{N-1} \\ y_N \\ \vdots \end{pmatrix} = \begin{pmatrix} A_1^1 & A_2^1 & & & & & & \\ & A_2^2 & A_3^2 & & & & & \\ & & A_3^3 & A_4^3 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & & A_{N-1}^{N-1} & A_N^{N-1} & & \\ & & & & & A_N^N & \ddots & \\ & & & & & \ddots & \ddots & \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_{N-1} \\ y_N \\ \vdots \end{pmatrix}. \tag{3.1.2}$$

Where the block matrices along the diagonal position are defined as:

$$A_1^1 = F_1 \tag{3.1.3}$$

$$A_2^2 = (F_1 \otimes I) + (I \otimes F_1) \tag{3.1.4}$$

$$A_3^3 = (F_1 \otimes I \otimes I) + (I \otimes F_1 \otimes I) + (I \otimes I \otimes F_1) \tag{3.1.5}$$

$$and, A_j^j = F_1 \otimes I^{\otimes j-1} + I \otimes F_1 \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_1, \tag{3.1.6}$$

Similarly, the block matrices above the diagonal are defined as:

$$A_2^1 = F_2 \tag{3.1.7}$$

$$A_3^2 = (F_2 \otimes I) + (I \otimes F_2) \tag{3.1.8}$$

$$A_4^3 = (F_2 \otimes I \otimes I) + (I \otimes F_2 \otimes I) + (I \otimes I \otimes F_2) \tag{3.1.9}$$

$$and, A_{j+1}^j = F_2 \otimes I^{\otimes j-1} + I \otimes F_2 \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_2, \tag{3.1.10}$$

In numerical analysis, we truncate the above infinite-dimensional system of linear ODEs at order N, thereby obtaining a finite system with the upper triangular block structure

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix} = \begin{pmatrix} A_1^1 & A_2^1 & & & & & & \\ & A_2^2 & A_3^2 & & & & & \\ & & A_3^3 & A_4^3 & & & & \\ & & & & \ddots & & & \\ & & & & & \ddots & & \\ & & & & & & A_{N-1}^{N-1} & A_N^{N-1} \\ & & & & & & & A_N^N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix}. \tag{3.1.11}$$

It is more concisely represented as

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = A\mathbf{y}, \qquad \mathbf{y}(0) = \mathbf{y}_{\mathrm{in}} \tag{3.1.12}$$

Recall, $y_j = u^{\otimes j}$, $\mathbf{y}_{\mathrm{in}} = [u_{\mathrm{in}}, u_{\mathrm{in}}^{\otimes 2}, \ldots, u_{\mathrm{in}}^{\otimes N}]$, and $A_j^j \in \mathbb{R}^{n^j \times n^j}$, $A_{j+1}^j \in \mathbb{R}^{n^j \times n^{j+1}}$ for $j \in [N]$.

We now refer to the truncated Carleman matrix (up to order N) as **the Carleman matrix** unless stated otherwise.

We prove two useful lemmas on the sparsity and size of the Carleman matrix as follows

Lemma 3 *The N-th order truncated Carleman matrix A is an $(3N)$-sparse matrix, where $s = 3$ is the sparsity of F_1 and F_2.*

Proof: Due to the block structure of the matrix A, the sparsity depends on the sparsity of the last blocks A_N^N and A_N^{N-1}. These block matrices are constructed using the Kronecker product of identity matrix I with F_1 (or F_2). We have discussed F_1 and F_2 have sparsity $s = 3 \in O(1)$. As a consequence, the sparsity of block matrix A_N^N is $3N \in O(N)$. □

The dimension of the Carleman matrix ((3.0.1)) is crucial for analyzing the algorithms' complexity.

Lemma 4 *The size of the Carleman matrix depends on the size of matrix F_1 and truncation order N as*

$$d := \Delta := n + n^2 + \cdots + n^N = \frac{n^{N+1} - n}{n-1} = O(n^N). \tag{3.1.13}$$

Proof: Compute the size of each of the block matrices A_j^j. Now, use the fact that the sum of their size is equal to the size of the Carleman matrix. □

This finite order truncation is bound to introduce error. For parameter $R < 1$ (equation (2.1.2)), ([Liu21]) have proved that the truncation error (exponentially) converges as truncation order N is increased. The stability of the numerical solution is formally stated in Chapter 4. Our MATLAB simulation results could be interpreted as validating this result.

3.2 Two existing techniques to solve the Carleman ODE

We briefly summarize the existing methods to solve the Carleman ODE. They will serve as a benchmark for our algorithms discussed in the next chapter.

3.2.1 Forward Euler Method (Liu et al. 2023)

[Liu23] solved the history-state quantum encoding problem (i.e., problem 2). They use the forward Euler method to solve the Carleamn ODE arising from a reaction-diffusion equation.

The Carleman ODE is a first-order homogenous ODE.

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = A\mathbf{y}, \qquad \mathbf{y}(0) = \mathbf{y}_{\text{in}} \tag{3.2.1}$$

Forward Euler method: The time interval $[0, T]$ is uniformly discretized in $m = T/h_t$ sub-intervals to solve using the forward Euler method.

Figure 3.2: Uniform time steps between $[0,\ T]$

The recurrence relation is given by (assume $h := h_t$)

$$y^{k+1} = [I + Ah]y^k \tag{3.2.2}$$

where component $y^k \in \mathbb{R}^\Delta$ approximates the value of $\mathbf{y}(t = kh_t)$ for each $k \in [m+1]_0 \coloneqq \{0, 1, \ldots, m\}$. The base case $\mathbf{y}^0 = \mathbf{y}(t = 0) := \mathbf{y}_{\text{in}}$. This gives an $((m+1)\Delta) \times ((m+1)\Delta)$ sized linear system

$$\begin{pmatrix} I & & & & \\ -(I+Ah) & I & & & \\ & \ddots & \ddots & & \\ & & -(I+Ah) & I & \\ & & & -(I+Ah) & I \end{pmatrix} \begin{pmatrix} y^0 \\ y^1 \\ \vdots \\ y^{m-1} \\ y^m \end{pmatrix} = \begin{pmatrix} y_{\text{in}} \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}. \tag{3.2.3}$$

Quantum encoding of the linear system: This is encoded in quantum form as

$$L|\mathbf{Y}\rangle = |B\rangle, \tag{3.2.4}$$

where

$$L = \sum_{k=0}^{m+1} |k\rangle\langle k| \otimes I - \sum_{k=1}^{m+1} |k\rangle\langle k-1| \otimes (I + Ah). \tag{3.2.5}$$

$$|\mathbf{Y}\rangle = \frac{1}{\||\mathbf{Y}\rangle\|} \sum_{k=0}^{m} y^k |k\rangle = \frac{1}{\|\mathbf{Y}\|} \sum_{k=0}^{m} \sum_{j=1}^{N} y_j^k |j\rangle|k\rangle \tag{3.2.6}$$

where the normalization factor satisfies $\||\mathbf{y}\rangle\|^2 = \sum_{k=0}^m \|y^k\|^2 = \sum_{k=0}^m \sum_{j=1}^N \|y_j^k\|^2$. Also

$$|B\rangle = \mathbf{y}_{\text{in}} \, |k = 0\rangle \tag{3.2.7}$$

The quantum (encoded) linear equation (3.2.4) encodes the recurrence relation (3.2.2).

The state encoding can be explicitly written as:

$$\mathbf{Y} = [y^0,\ y^1, \cdots, y^m]^T \tag{3.2.8}$$

$$where,\ y^k = [y_1^k,\ y_2^k, \cdots, y_N^k]^T \quad \forall k \in \{0, ..., m\} \tag{3.2.9}$$

$$and\ y_j^k = [u(t = kh)]^{\otimes j} \tag{3.2.10}$$

The QLSA solver: This is solved using the QLSA mentioned in [CKS15], whose output is a quantum state vector encoding the solution $\mathbf{Y}$ of the linear system ((3.2.4)).

Conditional measurement: The goal is to produce the history state encoding

$$|u_{\text{evo}}\rangle = \sum_{k=0}^{m} u(kh)|k\rangle = \sum_{k=0}^{m} y_1^k\,|j = 1\rangle\,|k\rangle \tag{3.2.11}$$

It is achieved by conditional measurement on register $|j\rangle$. If the outcome of the register $|j\rangle$ is one, success is declared. Otherwise, the algorithm is repeated. □

Reason: Eq. (3.2.8) implies the measurement outcome of $j = 1$ would collapse the quantum state $\mathbf{y}$ to the desired state

$$|u_{\text{evo}}\rangle := |y_1^k\rangle = |u(0)\rangle + |u(h)\rangle + \cdots + |u(mh)\rangle = \sum_{k=0}^{m} u(kh)|k\rangle \tag{3.2.12}$$

Success probability on conditional measurement: It is crucial to figure out how probable the event of getting $j = 1$ is on conditional measurement. The theorem 6 ([Liu23]) provides a lower bound on the probability of measuring the state corresponding to $j = 1$ as follows.

Theorem 1 *Then the probability of measuring a quantum state* $|u_{\text{evo}}\rangle = \sum_{k=0}^{m} |y_1^k\rangle$ *satisfies*

$$P_{(\text{j=1})} \geqslant \frac{2G^2}{16 \max_{t\in[0,T]} \|\mathbf{y}(t)\|^2 + G^2}. \tag{3.2.13}$$

Where

$$G \coloneqq \sqrt{\frac{\sum_{k=0}^{m} \|\mathbf{y}_1(kh)\|^2}{m+1}}, \tag{3.2.14}$$

and due to equation 4.54 in [Liu23]

$$\max_{t\in[0,T]} \|\mathbf{y}(t)\|^2 \leq ||u_{\text{in}}||^{4N} \tag{3.2.15}$$

□

Amplitude implication: This is required to improve the success of measuring the desired state after

conditional measurement. The last theorem implies

$$\mathcal{O}(\sqrt{1/P_{(\mathrm{j}=1)}}) = O\left(\frac{u_{\mathrm{in}}^{2N}}{G}\right)$$

Thus, this many rounds of amplitude amplification subroutine on $|\mathbf{y}\rangle$ is sufficient to boost the success of measuring $j = 1$ to nearly 100 percent. □

Claim: The gate complexity of the whole algorithm is given by

$$O(C_{amp}C_{QLSA}) \tag{3.2.16}$$

Reason: We can see QLSA as a subroutine to produce quantum state $|\mathbf{y}\rangle$. The amplitude amplification has improved the change of measuring the desired output $|u_{\mathrm{evo}}\rangle$ (subspace) on the conditional measurement on $|\mathbf{y}\rangle$ (space). □

Computational cost:

- The gate complexity of amplitude amplification is known to be optimal.
- Different variants of QLSA are known to have different gate complexity.

[Liu23] use theorem 5 from [CKS15] whose gate complexity is

Theorem 2 *([CKS15]) Let $M|x\rangle = |b\rangle$ be an s-sparse quantum linear system problem with matrix size n and condition number κ. Assume oracles for matrix M and b are provided. Then, there exists a quantum algorithm that produces a state ε approximate to the analytic solution state $|x\rangle$ with gate complexity*

$$O(s\kappa poly(log(s\kappa/\varepsilon))(log(n) + poly(log(\kappa/\varepsilon)))) \tag{3.2.17}$$

Since there is an upper bound on $P_{(j=1)}$ (Theorem 1) and condition number κ (equation 4.29 and 4.51 [Liu23]) in terms of system parameters. The final complexity of the problem is given in the following theorem.

Theorem 3 (Theorem 4.1 in [Liu23]) *For given quadratic ODE problem-1 with parameter $R < 1$, there exists a quantum algorithm that produces a state that approximates $u_{\mathbf{evo}}$ succeeding with probability $\Omega(1)$ with the query complexity (to the oracles O_{F_1}, O_{F_2}, and $O_{u_{\mathrm{in}}}$)*

$$\frac{1}{G^2\epsilon} sT^2D^2n^4N^3\|u_{\mathrm{in}}\|^{2N} \cdot \mathrm{poly}\left(\log\frac{aDnNsT}{G\epsilon}\right). \tag{3.2.18}$$

The gate complexity is larger than its query complexity by logarithmic factors. □

Where, notation

- G is normalization factor (equation (3.2.14)),
- s is the sparsity of F_1 matrix,
- T is simulation time,
- n is the number of spatial discretization points,
- N is Carelman truncation order
- $||u_{\text{in}}||$ is the norm of the initial solution (at t=0)
- a and b are parameters of the Fischer-KPP equation

Conclusion: While the gate complexity scales polynomials in several parameters, the factor of $(\frac{1}{\varepsilon})$ is not desirable. Also parameter N appears in power to $||u_{in}||$. In the next algorithm by Costa23 improves it to poly$(\log(\frac{1}{\varepsilon}))$. It also removes the N appearing in the exponent. In the next section, we see these details.

3.2.2 Truncated Tylor method (Costa et al. 2023)

The runtime of the last algorithms is improved by Costa et al.([CSM23]). They introduced two key changes as

1. Rescaling the quadratic ODE problem before linearization (say by a factor $1/\gamma$)
2. Solving the Carleman ODE by truncated Taylor series method

At last, the algorithm's output is multiplied by the inverse of the scaling factor γ to retrieve the desired solution.

It should be noted that the paper [CSM23] solves the another variant of the problem (see Problem 2) that aims to output quantum vector $|u(T)\rangle$ rather than the history state

$$|u_{\text{evo}}\rangle = \sum_{k=0}^{m} u(kh)|k\rangle.$$

As we see alter, the key difference is in how conditional measurement is done on the output of the QLSA subroutine (i.e., $|\mathbf{y}\rangle$).

Remark: In chapter 6, we develop our algorithm to solve the same problem 1.

Definition 1 (Rescaled Quadratic ODE problem) *Consider a nonlinear ODE system of the form* $\frac{\mathrm{d}u}{\mathrm{d}t} = F_1 u + F_2 \mathbf{u}^{\otimes 2}$ *as in Eq.* (5.1.1)*. Then, using a variable transformation in the form of a rescaling*

$\widetilde{u} = u/\gamma$ with $\gamma > 0$, we obtain another system in the rescaled variable

$$\frac{\mathrm{d}\widetilde{u}}{\mathrm{d}t} = \widetilde{F}_1\widetilde{u} + \widetilde{F}_2\widetilde{u}^{\otimes 2}, \tag{3.2.19}$$

with $\widetilde{F}_1 = F_1$ and $\widetilde{F}_2 = \gamma F_2$. □

Good choice of the scaling factor: Equation 55 of [CSM23] prescribes taking the scaling factor as per

$$\gamma \leq \frac{||u_{\text{in}}||}{R}. \tag{3.2.20}$$

Scaled Carleman ODE: The scaling factor slightly alters the Carleamn ODE problem. We get

$$\widetilde{A}_j^j = A_j^j \text{ and } \widetilde{A}_{j+1}^{(j)} = \gamma A_{j+1}^j \tag{3.2.21}$$

$$\widetilde{\mathbf{y}} = [\widetilde{u},\ \widetilde{u}^{\otimes 2}, \cdots,\ \widetilde{u}^{\otimes N}] \tag{3.2.22}$$

As a result, we can write the rescaled Carleman linearization as

$$\frac{\mathrm{d}\widetilde{\mathbf{y}}}{\mathrm{d}t} = \widetilde{A}\widetilde{\mathbf{y}}, \tag{3.2.23}$$

where

$$\widetilde{A} = \begin{bmatrix} A_1^1 & \gamma A_2^1 & \cdots & 0 & & 0 & \cdots & 0 \\ 0 & A_2^2 & \gamma A_3^2 & 0 & 0 & 0 & 0 & \vdots \\ \vdots & 0 & \ddots & & & 0 & \ddots & 0 \\ & & \ddots & \ddots & & & \ddots & 0 \\ & & & & \ddots & & & 0 \\ & & & & \ddots & \ddots & & \vdots \\ \vdots & & & & & 0 & A_{N-1}^{N-1} & \gamma A_{N-1}^N \\ 0 & 0 & \cdots & & & \cdots & 0 & A_N^N \end{bmatrix}. \tag{3.2.24}$$

Note: We drop the tilde (symbol) above matrix A and vector $\mathbf{y}$ for convenience.

The QLSA solver: This homogenous and time-independent ODE is solved using the Taylor series method as described in reference [BC22]. This output the quantum vector $|\widetilde{\mathbf{y}}\rangle$.

Conditional measurement: Since the goal of the problem 2 is to output $|u(T)\rangle$. This is recovered by conditional measurement on

$$\widetilde{\mathbf{y}} = \sum_{j=1}^{N} \widetilde{y_j}\,|j\rangle \ \ for\ j \in \{1, ..., N\}. \tag{3.2.25}$$

You measure the register $|j\rangle$ and declare success only if you measure $j = 1$. Otherwise, the algorithm is repeated until successful.

Probability of the success on conditional measurement: [CSM23] gives an estimate of how probable this event is in the following theorem

Theorem 4 *(Lemma 2 in [CSM23]) For the scaled Carleamn ODE with the truncation number N and parameter $R < 1$, the probability of getting $j = 1$ is given by $P(\widetilde{y_1}) \geq \frac{1}{N}$*

Note 1: We have skipped the internal details of the ODE solver [BC22]. As a matter of fact, they approximate the solution using the K-th order Taylor series. It gives a recurrence relation similar to the forward Euler method, i.e., a linear system. Then, they use QLSA solver [CAS22] (instead of [CKS15]).

Note : The gate complexity of the solver [CAS22] depends on the gate complexity of the block encoding of the matrices. They have computed the cost of such an operation in terms of input parameters.

Theorem 5 (Lemma 5 in [CSM23]) *For given quadratic ODE problem-2 with parameter $R < 1$, there exists a quantum algorithm that produces a state that approximates $u(T)$ succeeding with probability $\Omega(1)$ with the query complexity to the oracles O_{F_1} and O_{F_2}*

$$\mathcal{O}\left(\frac{1}{\sqrt{1-R^2}}\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}(|a|+(4\pi^2/3)Dn^2)TN\log\left(\frac{N}{\varepsilon}\right)\log\left(\frac{N(|a|+(4\pi^2/3)Dn^2)T}{\varepsilon}\right)\right), \quad (3.2.26)$$

and query complexity to oracles for $\mathbf{u}_{\text{in}}$ (say, $O_{u_{\text{in}}}$),

$$\mathcal{O}\left(\frac{1}{\sqrt{1-R^2}}\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}(|a|+(4\pi^2/3)Dn^2)TN^2\log\left(\frac{N}{\varepsilon}\right)\right), \quad (3.2.27)$$

*and an **additional** gate of order*

$$\mathcal{O}\left(\frac{1}{\sqrt{1-R^2}}\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}(|a|+(4\pi^2/3)Dn^2)TN^2\log\left(\frac{N}{\varepsilon}\right)\log\left(\frac{N(|a|+(4\pi^2/3)Dn^2)T}{\varepsilon}\right)^2\log n\right), \quad (3.2.28)$$

is required. The Carleman truncation order scales as

$$N = \mathcal{O}\left(\frac{\log(1/\varepsilon)}{\log(1/R)}\right). \quad (3.2.29)$$

With the assumption of $\lambda_{F_2}/\|F_2\| = \mathcal{O}(\lambda_{F_1}/\|F_1\|)$ for the block encodings of F_1 and F_2.

Where, notation

- D is diffusion coefficient
- a is the coefficient of $u(x,t)$ in Fisher-KPP equation
- R is non-linearity parameter (see (2.1.2))

- T is simulation time
- n is the number of spatial discretization point
- N is Carelman truncation order
- $||u_{\text{in}}||$ is norm of initial solution (at t=0)
- λ_{F_1} is λ-value for the block encoding of matrix F_1 (refer to QLSA solver in [CAS22])

Conclusion: They have improved the ε dependency from poly$(1/\varepsilon)$ to poly$(\log(1/\varepsilon))$. The term containing parameter N in the exponent has been pulled off due to rescaling the equation.

Chapter 4

Error Analysis and Numerical Simulation

In the next chapter, we discuss the stability of the solution due to the introduction of various approximation techniques ranging from Carleman truncation, central difference discretization, and numerical integration schemes (Euler and Taylor). We give our MATLAB simulation results for a specific Fisher-KPP equation with the Dirichlet boundary condition.

4.1 Carleman truncation error

The error due to finite truncation of Carleman ODE, originating from the reaction-diffusion equation, was first studied by [Liu23]. For $R < 1$, they proved the truncation error exponentially converges as the truncation number increases. (assume the vector norm $||.||$ is l_2 norm)

Theorem 6 *(Theorem 3.2 in [Liu23]) Let the eigenvalues of matrix F_1 in quadratic ODE be all real and negative, and λ_1 be the largest eigenvalue. Then for any $j \in [N]$, the truncation error $\eta_j(t) \coloneqq u^{\otimes j}(t) - y_j(t)$ satisfies*

$$\|\eta_j(t)\| \leqslant \|u_{\text{in}}\|^j R^{N+1-j}(1 - e^{\lambda_1 t}). \tag{4.1.1}$$

For $j = 1$, we have the tighter bound

$$\|\eta_1(t)\| \leqslant \|u_{\text{in}}\| R^N \big(1 - e^{\lambda_1 t}\big). \tag{4.1.2}$$

Later on [CSM23] improved the error bound as follow

Theorem 7 *(Lemma 4 in [CSM23]) In the same context as theorem Theorem 7, the truncation error is given as*

$$\|\eta_j(t)\| \leqslant \|u_{\text{in}}\|^j R^{N+1-j} f_N(\lambda_1 t). \tag{4.1.3}$$

For $j = 1$, we have the tighter bound

$$\|\eta_1(t)\| \leqslant \|u_{\text{in}}\| R^N f_N\big(\lambda_1 t\big). \tag{4.1.4}$$

Where $f_N(\lambda_1 t) \leq (1 - e^{\lambda_1 t})$.

The above theorems are characterized by parameter R, which depends on the highest eigenvalue of matrix F_1. Recently, a new regime for convergence has been explored by [WWX24]. It is quantified in terms of a No-resonance parameter that depends on a slightly different property of the eigenvalues of matrix F_1. We discuss it in the next chapter when we define the No-resonance condition.

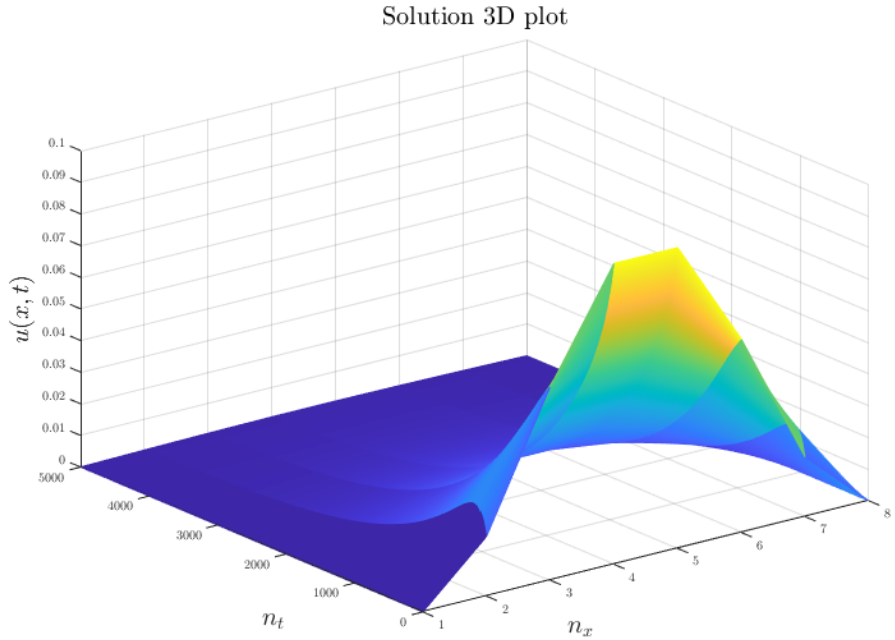

Figure 4.1: (Euler method) Solution for reaction diffusion equation for $D = 0.2,\ a = 0.4$ *and* $b = -1$. The initial distribution $u(x, t = 0) = 0.1sin^2(\pi x)$. Position domain $x \in [0, 1]$ is discretized into 8 points while time $t \in [0, 3]$ is discretized into 5000 points

.

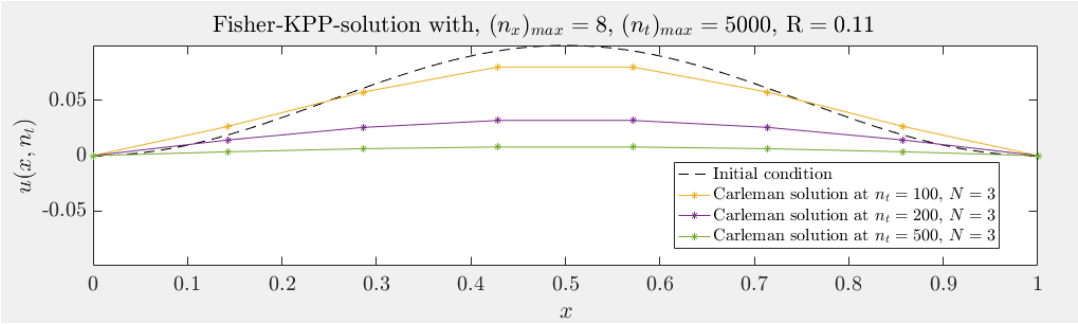

Figure 4.2: (Euler method) Time snapshots at three different times reveal how the solution decays in magnitude as time progresses. Position domain $x \in [0, 1]$ is discretized into 8 points while time $t \in [0, 3]$ is discretized into 5000 points

4.2 Error due to ODE solver: Euler & Taylor method

This type of error is well-studied in numerical methods for ODE. For the Euler method, the step size is the key determining parameter.

Theorem 8 *(Lemma 4.3 in [Liu23]) In the context of problem 1 with parameter $R < 1$, if we select the time step as*

$$h \leq \frac{1}{N^2||F_1||} \tag{4.2.1}$$

then the error due to the Euler method is bounded as

$$\varepsilon_{\text{euler}} \leq N^2 T h(||F_1|| + ||F_2||)^2 (max_{t\in[0,T]}||\mathbf{y}(t)||) \tag{4.2.2}$$

For the Fisher-KPP equation, the values of these matrix norms are as follows

$$||F_1|| = 4D(n+1)^2 + a \tag{4.2.3}$$

$$||F_2|| = b \tag{4.2.4}$$

For K-th order truncated Taylor series method, the time propagation error is given as

$$\varepsilon_{Taylor} \in \left(\frac{(||A||h)^{K+1}}{(K+1)!} ||\mathbf{y}(0)|| \right) \tag{4.2.5}$$

The spectral norm of the Carleman matrix is known to be bounded by

$$||A|| \leq N(||F_1|| + ||F_2||) \tag{4.2.6}$$

It is combined to get the truncation order K as a function of ε and system parameters.

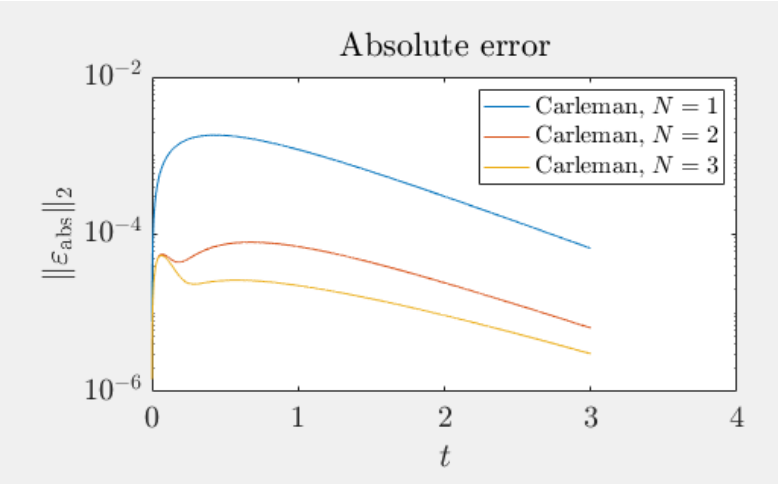

Figure 4.3: (Euler method: absoulte error vs time) As time progresses, the error eventually starts decreasing. Time $t \in [0, 3]$ is discretized into 5000 points

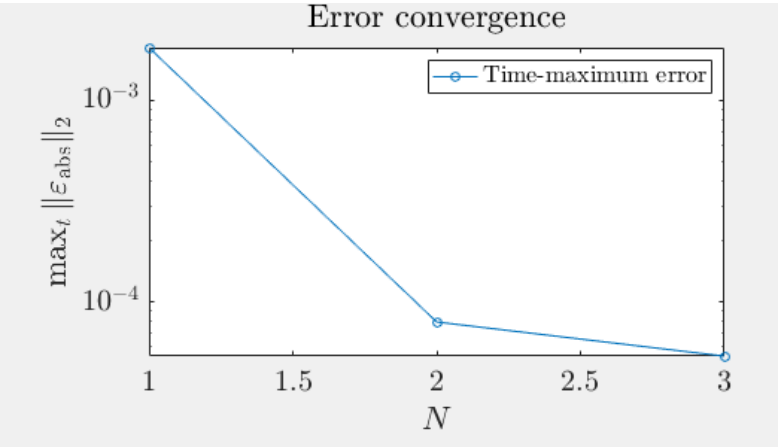

Figure 4.4: (Euler method: time-maximum error vs. Carleman truncation order) As expected, the error decreases as the truncation order increases.

4.3 Error due to spatial discretization: 3-point stencils

[CSM23] estimated the error due to uniform finite difference discretization.

Theorem 9 *Using a finite difference discretization with* 3 *stencil points in one dimension, the solution of the Fisher-KPP equation at time* $T > 0$ *has an error due to spatial discretization when* $a < 0$ *and* $|a| > |b|\|\mathbf{u}_{\text{in}}\|$ *bounded as*

$$\|\varepsilon_{\text{disc}}(T)\| = \mathcal{O}\left(n^{-1/2}\|\frac{d^3(u(x,t))}{d^3x}\|\frac{1-\exp\{(a+|b|\|\mathbf{u}_{\text{in}}\|)\,t\}}{|a+|b|\|\mathbf{u}_{\text{in}}\||}\right), \tag{4.3.1}$$

where n *is the number of grid points used.*

Proof: It is derived from lemma 7 in [CSM23], where it is given in general for a d-dimensional system using $(2k+1)$ stencils. In our case, we have a one-dimensional problem with a 3-point stencil. Putting d=1 and k=1 yields the above theorem. □

Remark: The results on Carleman truncation error and finite difference discretization are applicable in our case, too. We will refer to these results in Chapter 6 while discussing the sources of error for our algorithm.

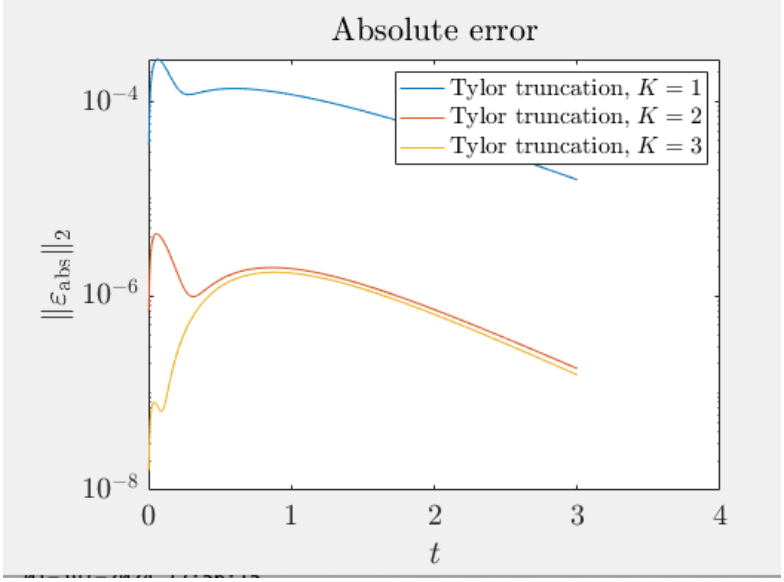

Figure 4.5: (Taylor method: absoulte error vs time graph) They corresponds to three different orders of Taylor truncation. We have fixed the Carleman truncation order to $N = 3$. Time step size: the time domain $t \in [0, 3]$ is uniformly divided into 2000 parts for numerical simulation.

Numerical Simulation details: We are solving the Fisher-KPP equation

$$\frac{\partial u(x,t)}{\partial t} = 0.2\Delta u(x,t) + 0.4u(x,t) - u^2(x,t) \tag{4.3.2}$$

with initial distribution

$$u(x, t=0) = 0.1sin^2(\pi x) \tag{4.3.3}$$

Proxy for the analytic solution: Errors are defined with respect to the analytic solution. We use MATLAB Runge-Kutta 45 ODE solver (MATLAB:ode45) to solve the quadratic ODE problem. We take this solution as a proxy for the analytic solution. It is one of the best ODE solvers available in MATLAB.

Then, the same quadratic ODE is solved using either

- Carleman linearization + Forward Euler method (fig: 4.1 to 4.4)

- Carleman linearization + Taylor series method (fig: 4.5 to 4.6)

Methodology: It is a classical numerical result. The quantum algorithm is supposed to solve the Euler/Taylor method by solving the corresponding linear system equation. We solved it iteratively per the Euler/Taylor method without embedding it into a linear system. It is sufficient to see how stable the solution is due to Carleman linearization and numerical schemes (the Euler/Taylor method)

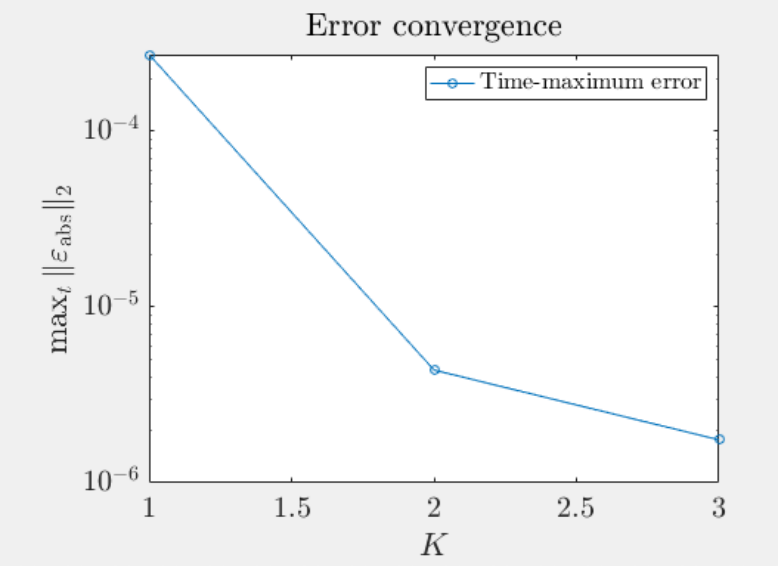

Figure 4.6: Taylor method) As the Taylor truncation order increases, the time-maximum error decreases rapidly.

Chapter 5

Diagonalization of the Carleman matrix

Prelude: In chapter 3, we have seen the forward Euler and Taylor series method to solve the Carleman ODE. In approximation theory literature, they are called local approximation schemes. They closely approximate the function at the initial point, but it gets worse as we move far from the initial point.

Meanwhile, Chebyshev's approximation connects variables globally. As a result, the error is nearly uniform throughout the domain. The series is known to converge faster than the Taylor series. A comprehensive analysis of the Chebyshev series can be found in [Pow81].

It has motivated us to use the Chebyshev approximation to solve the Carleman ODE (a homogenous and time-dependent ODE)

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{A}\mathbf{y}(t). \tag{5.0.1}$$

But the key hurdle is that it requires the Carleman matrix $\mathbf{A}$ to be diagonalizable. In this chapter, we derive sufficient conditions for the diagonalization of the Carleman matrix.

Key results: The Carleman matrix is not a Hermitian matrix. It is not even a normal matrix [MS1]. Thus, the method of its diagonalization is not very apparent, as we have in the case of normal matrices. Nevertheless, we exploit its bidiagonal/triangular block structure along with its special eigen-properties to design an iterative method to diagonalize it.

We start by summarizing the two important results derived in this chapter.

Theorem 10 *All the eigenvalues of the Carleman matrix are real and negative if diffusion constant D and parameter a are related by*

$$4D(n+1)^2 sin^2(\frac{j\pi}{2(n+1)}) \geq a \tag{5.0.2}$$

Theorem 11 *The sufficient conditions for diagonalization of the Carleman matrix are:*

1. *all the block matrix A_j^j are diagonalizable*
2. *the Carleman matrix fulfills the **No-resonance** condition*

5.1 Sufficient condition for diagonalization of the matrix

For the Fisher-KPP equation $\partial^t u = D\Delta_x u + au + bu^2$, we get the following quadratic ODE on n-point discretization,

$$\frac{\mathrm{d}u}{\mathrm{d}t} = F_1 u + F_2 u^{\otimes 2}, \qquad u(0) = u_{\text{in}}. \tag{5.1.1}$$

where, $F_1 = (DL_h + aI) \in \mathbb{R}^{n\times n}$ and $F_2 = \mathbf{b} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n^2}}$. Finally, Carleman linearization gives the ODE

$$\frac{\mathrm{d}y}{\mathrm{d}t} = Ay, \qquad y(0) = y_{\text{in}} \tag{5.1.2}$$

Now, we prove why the eigenvalues of matrix F_1 are sufficient to compute the eigenvalue of the Carleman matrix $\mathbf{A}$. Or, due to the upper triangular nature of the matrix A, the off-diagonal block matrices **don't** contribute to the eigenvalues of $\mathbf{A}$.

In Lemma 1 we derived that all the eigenvalues of the matrix F_1 are given by

$$\lambda_j = -4D(n+1)^2 sin^2\left(\frac{j\pi}{2(n+1)}\right) + a;\ j \in \{1, ..., n\} \tag{5.1.3}$$

We begin with observations on the spectral properties of $\mathbf{A}$. Since,

$$\mathbf{A} = \begin{pmatrix} A_1^1 & A_2^1 & & & & & \\ & A_2^2 & A_3^2 & & & & \\ & & A_3^3 & A_4^3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & A_{N-1}^{N-1} & A_N^{N-1} & \\ & & & & & A_N^N & \ddots \\ & & & & & \ddots & \ddots \end{pmatrix} \tag{5.1.4}$$

Where the block matrices along the diagonal position are defined as:

$$A_1^1 = F_1 \tag{5.1.5}$$

$$A_2^2 = (F_1 \otimes I) + (I \otimes F_1) \tag{5.1.6}$$

$$A_3^3 = (F_1 \otimes I \otimes I) + (I \otimes F_1 \otimes I) + (I \otimes I \otimes F_1) \tag{5.1.7}$$

$$and, A_j^j = F_1 \otimes I^{\otimes j-1} + I \otimes F_1 \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_1, \tag{5.1.8}$$

Similarly, the block matrices above the diagonal are defined as:

$$A_2^1 = F_2 \tag{5.1.9}$$

$$A_3^2 = (F_2 \otimes I) + (I \otimes F_2) \tag{5.1.10}$$

$$A_4^3 = (F_2 \otimes I \otimes I) + (I \otimes F_2 \otimes I) + (I \otimes I \otimes F_2) \tag{5.1.11}$$

$$and, A_{j+1}^j = F_2 \otimes I^{\otimes j-1} + I \otimes F_2 \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_2, \tag{5.1.12}$$

Lemma 5 *Assume that block matrix A_1^1 has eigenvalues $\{\lambda_1, \cdots, \lambda_d\}$, then the set of eigenvalues of block matrix $\mathbf{A_j^j}$ is,*

$$\lambda[A_j^j] = \left\{ \sum_{k=1}^{d} m_k \lambda_k \mid m_k \in \{0, ..., j-1\} \text{ and } \sum_{k=1}^{d} m_k = j \right\}.$$

For example, the set of eigenvalues of A_2^2 is

$$\lambda[A_2^2] = \{(\lambda_1 + \lambda_1), (\lambda_1 + \lambda_2), ..., (\lambda_1 + \lambda_d), (\lambda_2 + \lambda_1), ..., (\lambda_d + \lambda_d)\}.$$

proof: As described earlier, $A_1^1 = F_1$. For the block matrix

$$A_j^j = F_1 \otimes I^{\otimes j-1} + I \otimes F_1 \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_1 \tag{5.1.13}$$

each of the matrices in the summation is normal and commutes pair-wise. Thus, the eigenvalues of A_j^j are given as the sum of all possible combinations of the eigenvalues of the matrices in the summation [MS5]. The eigenvalues of a summand in Eq. (5.1.14) are

$$\lambda(I \otimes \cdots \otimes F_1 \otimes \cdots \otimes I) = \lambda(F_1) \tag{5.1.14}$$

due to the property of Kronecker's product. There are j summands in Eq. (5.1.14). It leads to the

desired expression. □

Theorem 12 *The set of eigenvalues of the Carleman matrix A is given by*

$$\lambda[A] = \left\{ \sum_{k=1}^{d} m_k \lambda_k \mid m_k \in \{0, ..., N\} \text{ and } \sum_{k=1}^{d} m_k \leq N \right\}. \tag{5.1.15}$$

Proof: Eigenvalues of a block upper triangular matrix is the union of the eigenvalues of the block matrices in the main diagonals [MS2]. The eigenvalues of each block matrices A_j^j are given in Lemma 5. Taking their union would give the desired set. □

Definition 2 ***(No-resonance condition)*** *Let the set of eigenvalues of the block matrix A_j^j be denoted by $\{\lambda(A_j^j)\}$, then the matrix A follows the No-resonance condition if and only if*

$$\{\lambda(A_1^1)\} \cap \{\lambda(A_2^2)\} \cap ... \cap \{\lambda(A_N^N)\} = \{\Phi\} \tag{5.1.16}$$

Or, any two block matrices along the diagonal of **A** should not have any common eigenvalues. □

There exists an equivalent (algebraic) definition to characterize *No-resonance* condition as below.

Definition 3 ***(No-resonance condition)*** *Given A_1^1 has eigenvalues $\{\lambda_1, \cdots, \lambda_d\}$, Then the matrix* **A** *is said to follow the No-resonance condition if $\forall i \in [d]$*

$$\lambda_i \neq \sum_{j=1}^{n} m_j \lambda_j, \quad \forall m_j \in \{0, 1, ...\} \text{ and } \text{ s.t. } 2 \leq \sum_{j=1}^{n} m_j \leq N. \tag{5.1.17}$$

This definition has been mentioned in equation 9 of [WWX24].

We derive a closed-form expression to determine if the Carleman matrix from the Fisher-KPP equation owes the No-resonance condition.

Theorem 13 *Checking No-resonance for the Carleman matrix amounts to proving the below trigonometric equation has no solution*

$$-4D(n+1)^2 sin^2\left(\frac{j\pi}{2(n+1)}\right) + a \neq \sum_{l=1}^{d}(m_i)\left(4D(n+1)^2 sin^2\left(\frac{l\pi}{2(n+1)}\right) + a\right) \tag{5.1.18}$$

where $m_i \in \{0, 1, ..., n\}$ and $2 \leq \sum m_i \leq N$.

Proof: Take the value of λ_j from Lemma 1 and use it in Theorem 12. It gives the above trigonometric relation. □

5.2 Numerical evidence for the matrix diagonaliziblity

Finally, we have proved that the diagonalization of the Carleman matrix reduces to checking this No-resonance condition. It depends on four parameters.

- Fisher-KPP system parameter: D and a. (This input from a user can't be changed.)
- Hyperparameter N (truncation order) and n (no. of grid points) that depends on allowed approximation error

Hypothesis 1 *The Carleman Matrix satisfies the No-resonance condition for several choices of the Carleman truncation N and the number of discretization points d.*

Numerical test: Although the No-resonance is equivalent to checking the feasibility of the equation Eq. (5.1.18). It is not immediate to us how to prove or disproof it for all values of the parameters. Since we are designing a numerical solution, the numerical test below for the no-resonance indicates that it is likely that the no-resonance condition holds for a wide range of parameters.

Method: We start with Fisher-KPP equation parameters (say, $D = 0.2$ and $a = 0.4$). Now, we check for various combinations of Carleman truncation order and the number of discretization points (grid) in space. The result of the No-resonance condition is tabulated in the next table.

Conclusion: We infer that the cases where the No-resonance condition fails must be rare/isolated. Thus, if in a certain choice of parameter N and n, the No-resonance condition does not hold, then change this parameter to the next integer (say, $n+1$ grid point instead of n). Simulating a system with $n+1$ grid points instead of n won't impact the simulation result's usefulness.

No. of grid point (n)	Carleman truncation order (N)	No-resonance condition
4	5	Yes
8	3	Yes
8	4	Yes
8	5	Yes
16	3	Yes
16	4	Yes
32	3	Yes

5.3 Procedure and computational cost of the diagonalization

Since the Carelamn matrix is diagonalizable, it can be written as

$$A = V\Lambda V^{-1} \tag{5.3.1}$$

where $\Lambda = diag(\lambda_1, ..., \lambda_d)$. This section gives an explicit iterative method to construct the matrix V and V^{-1}. We use the following property of the block upper triangular matrix for this purpose.

Lemma 6 *Let P be a block upper triangular matrix*

$$P = \begin{bmatrix} P_{11} & P_{12} \\ & P_{22} \end{bmatrix} \tag{5.3.2}$$

where block matrices $P_{1,1} \in R^{p\times p}$ and $P_{2,2} \in R^{q\times q}$ are known to be diagonalizable. Assume they satisfy $P_{11}\vec{e}_i = \lambda_i\vec{e}_i$ for $i \in [p]$ and $P_{22}\vec{g}_j = \mu_j\vec{g}_j$ for $j \in [q]$. Then, the set of linearly independent eigenvectors of matrix P is given by

$$\left\{ \begin{bmatrix} e_1 \\ 0^q \end{bmatrix}, ..., \begin{bmatrix} e_p \\ 0^q \end{bmatrix}, ..., \begin{bmatrix} x_1 \\ g^1 \end{bmatrix}, ..., \begin{bmatrix} x_q \\ g_q \end{bmatrix} \right\} \tag{5.3.3}$$

*where **vector** $x_j = (P_{11} - \mu_j I)^{-1} P_{12} g_j$. The notation 0^q implies that the last q entries of the column vector are zero.*

Let the matrix V diagonalize matrix P by similarity transformation. i.e., $P = V\Lambda V^{-1}$. The matrix T is constructed by stacking these linearly independent vectors in the columns as below.

$$V = \left[\begin{bmatrix} e_1 \\ 0^q \end{bmatrix} ... \begin{bmatrix} e_p \\ 0^q \end{bmatrix} \begin{bmatrix} x_1 \\ g^1 \end{bmatrix} ... \begin{bmatrix} x_q \\ g_q \end{bmatrix} \right] \tag{5.3.4}$$

Proof: The proof follows very similar to as given in [MS2]. We notice

$$\begin{bmatrix} P_{11} & P_{12} \\ & P_{22} \end{bmatrix} \begin{bmatrix} e_i \\ 0^q \end{bmatrix} = \lambda_i \begin{bmatrix} e_i \\ 0^q \end{bmatrix} \tag{5.3.5}$$

It implies that the eigenvectors of P_{11} are post-padded with extra zeros (0^q) to give the whole matrix P eigenvectors. Now we see how the eigenvectors of P_{22} are pre-padded with a vector x_j to get the remaining eigenvectors for P. Notice,

$$\begin{bmatrix} P_{11} & P_{12} \\ & P_{22} \end{bmatrix} \begin{bmatrix} x_j \\ g_j \end{bmatrix} = \mu_j \begin{bmatrix} x_j \\ g_j \end{bmatrix} \tag{5.3.6}$$

Solving this eigenvector equation gives $x_j = (P_{11} - \mu_j I)^{-1} P_{12} g_j$. Thus, pre-padding x_j would give the remaining eigenvectors. It can be checked that these eigenvectors are linearly independent, too. Due to the Spectral theorem, having a set of linearly independent eigenvectors is sufficient to diagonalize a matrix. One needs to pad these eigenvectors column-wise to get the similarity transformation matrix

V. □

The vector x_j is well defined if $(P_{11} - \mu_j I)^{-1}$ is defined. Or, the eigenvalues of P_{11} don't coincide with any of the eigenvalues of P_{22}. One can see this is similar to the No-resonance condition. It was first observed by [TL89] and recently utilized by [WWX24] for the quantum algorithm for quadratic ODE.

Suppose the block matrix P_{11} and P_{22} are diagonalized like

$$P_{11} = V_{P_{11}} \Lambda_{P_{11}} V_{P_{11}}^{-1} \tag{5.3.7}$$

$$P_{22} = V_{P_{22}} \Lambda_{P_{22}} V_{P_{22}}^{-1} \tag{5.3.8}$$

Then, for matrix P the below construction holds for V and V^{-1} (let $P = V\Lambda V^{-1}$)

$$V = \begin{bmatrix} V_{P_{11}} & [X_j] \\ & V_{P_{22}} \end{bmatrix} \tag{5.3.9}$$

where $[X_j] = [x_1 \cdots x_j]$ The inverse of V can be computed using the result on the block triangular matrix.

Lemma 7 *The inverse of a matrix*

$$\begin{bmatrix} V_{P_{11}} & [X_j] \\ & V_{P_{22}} \end{bmatrix} \tag{5.3.10}$$

is given by

$$\begin{bmatrix} (V_{P_{11}})^{-1} & -(V_{P_{11}})^{-1}[X_j](V_{P_{22}})^{-1} \\ & (V_{P_{22}})^{-1} \end{bmatrix} \tag{5.3.11}$$

Proof: The proof is given in [MS3]. It can be verified by multiplying the matrix V and V^{-1}, which gives the identity matrix. □

An Iterative method for Diagonalization: Until now, we have seen how to diagonalize the simplest possible case of matrix P. We apply the process iteratively for larger block bi-diagonal matrices, like the Carelman matrix.

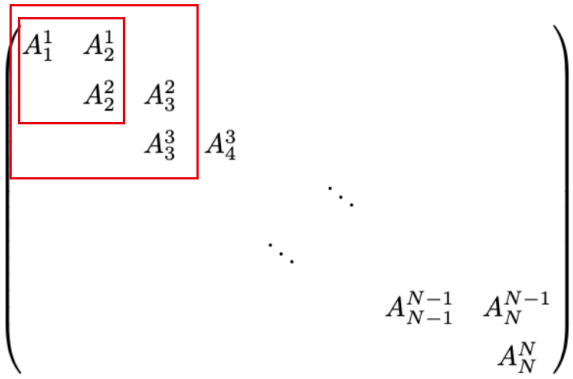

Figure 5.1: An iterative procedure to diagonalize the Carleman matrix begins with the uppermost three block matrices and uses Lemma 6. The process terminates at the last block A_N^N.

- We start from the top three block matrices that together form a matrix like P of Lemma 6. (Note: No-resonance guarantee that it can be diagonalized. We compute its V matrix.)
- We add two more blocks to our analysis, as shown in the below picture. Then we apply Lemma 6.
- We terminate the process after processing the last block A_N^N.

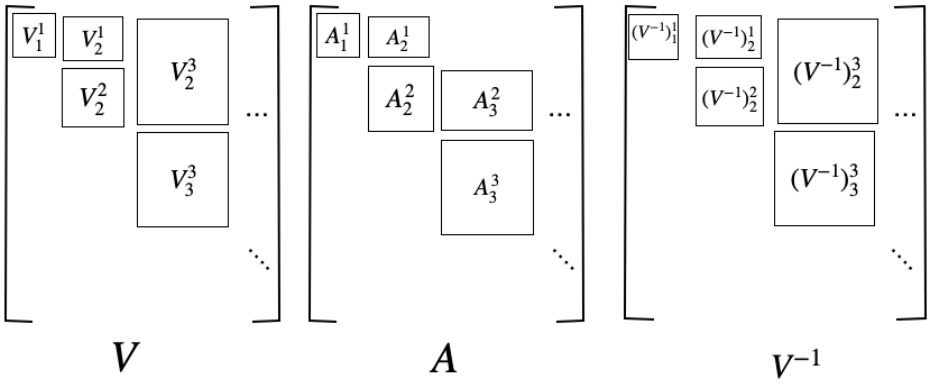

Figure 5.2: Although matrix **A** is a block bi-diagonal matrix, the matrix V and V^{-1} are block upper triangular. Computing diagonal block matrices V_j^j is less expensive than computing off-diagonal blocks V_j^{j+1} as described next.

Remark: A more rigorous discussion on diagonalization has been given in the recent work by [WWX24]. We have provided an iterative procedure to carry out the construction.

The computational cost of diagonalization

Computing matrices V_j^j of matrix V: It is relatively less expensive. We give a procedure to find them.

$$\text{Given } A_1^1 = F_1^1 = W\Lambda_1 W^{-1} \tag{5.3.12}$$

$$\textit{then, } A_2^2 = F_1 \otimes I + I \otimes F_1 = (W \otimes W)\Lambda_2(W^{-1} \otimes W^{-1}) \tag{5.3.13}$$

$$\cdots \tag{5.3.14}$$

$$A_j^j = (W \otimes ... \otimes W)\Lambda_j(W^{-1} \otimes ... \otimes W^{-1}) \tag{5.3.15}$$

Thus, $V_j^j = W \otimes ... \otimes W$ (Kronecker product of W with itself j times), and $[V_j^j]^{-1} = W^{-1} \otimes ... \otimes W^{-1}$.

Computing matrices V_{j+1}^j of matrix V : It is more expensive to compute the off-diagonal block matrices V_{j+1}^j. We have seen they are constructed by pre-padding the vectors as $V_{j+1}^j = [X_j] = [x_1 \cdots x_j]$ as in Lemma 6.

Computing $[X_j]$ requires

- Computing pre-padding vector $x_i = (A_{i-1}^{i-1} - \mu_i I)^{-1} A_{i-1}^j g_i$.
- It requires computing the inverse of $(A_{i-1}^{i-1} - \mu_i I)$.

$(A_{i-1}^{i-1} - \mu_i I)^{-1}$ is equivalently written as:

Theorem 14 *If the matrix $A_{i-1}^{i-1} = \sum_k \lambda_k g_k g_k^T$, then $(A_{i-1}^{i-1} - \mu_i I)^{-1} = \sum_k \frac{1}{(\lambda_k - \mu_i)} g_k g_k^T$.*

Let the cost of the above operation be $\mathcal{C}_{\text{inv}}$.

If $(A_{i-1}^{i-1} - \mu_i I)^{-1}$ is known, then the cost of computing $x_i = (A_{i-1}^{i-1} - \mu_i I)^{-1} A_{i-1}^j g_i$ is the same as computing two consecutive matrix-vector products.

As we have seen the matrix $(A_{i-1}^{i-1} - \mu_i I)^{-1}$ is not necessarily sparse. Thus, the maximum quantum advantage is limited. The cost of different types of matrix-vector products is discussed in the next chapter Theorem 17.

Conclusion: In computing the matrix V, the most expensive step appears to be the dense upper triangular **matrix-vector product.**

Chapter 6

Truncated Chebyshev series method for Carleman ODE

The Diagonalization of the Carleman matrix ($A = V\Lambda V^{-1}$) opens two new ways to explore the solution of the Carleman ODE.

- **Method I:** Estimating matrix exponentiation using the Chebyshev series. (It requires the matrix V and V^{-1} as a part of its algorithm.)
- **Method-II:** Chebyshev Psuedo spectal method. (It doesn't explicitly require V or V^{-1} in the algorithm. But the condition number of matrix V appears in the gate complexity).

6.1 Solving Carleman ODE using Matrix exponentiation

We are seeking an algorithm for the Fisher-KPP equation

$$\frac{\partial u}{\partial t} = D\Delta u + au + bu^2, \tag{6.1.1}$$

where the domain of the problem be $x \in [0, 1]$ and $t \in [0, T]$. It is converted into the quadratic ODE problem upon discretization in space.

6.1.1 Revisiting the scaled quadratic ODE

We have seen an advantage of problem rescaling in improving the success probability of measuring $u(T)$ on the conditional measurement on the Carleman vector $\mathbf{y}(T)$.

Definition 4 (Rescaled Quadratic ODE problem) *Consider a nonlinear ODE system of the form* $\dfrac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = F_1 u + F_2 \mathbf{u}^{\otimes 2}$ *as in Eq.* (5.1.1). *Then, using a variable transformation in the form of a rescaling* $\widetilde{u} = u/\gamma$ *with* $\gamma > 0$, *we obtain another system in the rescaled variable*

$$\frac{\mathrm{d}\widetilde{\mathbf{u}}}{\mathrm{d}t} = \widetilde{F}_1 \widetilde{\mathbf{u}} + \widetilde{F}_2 \widetilde{\mathbf{u}}^{\otimes 2}, \tag{6.1.2}$$

with $\widetilde{F}_1 = F_1$ *and* $\widetilde{F}_2 = \gamma F_2$. □

Good choice of the scaling factor: Equation 55 of [CSM23] prescribes taking the scaling factor as per

$$\boxed{\gamma \leq \frac{||u_{\mathrm{in}}||}{R}}. \tag{6.1.3}$$

It implies $\|\widetilde{\mathbf{u}}(t)\| < 1$ if $R < 1$.

Scaled Carleman ODE: The scaling factor slightly alters the Carleamn ODE problem. We get

$$\widetilde{A}_j^j = A_j^j \text{ and } \widetilde{A}_{j+1}^{(j)} = \gamma A_{j+1}^j \tag{6.1.4}$$

$$\widetilde{\mathbf{y}} = [\widetilde{u},\ \widetilde{u}^{\otimes 2}, \cdots,\ \widetilde{u}^{\otimes N}] \tag{6.1.5}$$

As a result, we can write the rescaled Carleman linearization as

$$\frac{\mathrm{d}\widetilde{\mathbf{y}}}{\mathrm{d}t} = \widetilde{A}\widetilde{\mathbf{y}}, \tag{6.1.6}$$

where

$$\widetilde{A} = \begin{bmatrix} A_1^1 & \gamma A_2^1 & \cdots & 0 & & 0 & \cdots & 0 \\ 0 & A_2^2 & \gamma A_3^2 & 0 & 0 & 0 & 0 & \vdots \\ \vdots & 0 & \ddots & & & 0 & \ddots & 0 \\ & & \ddots & \ddots & & & \ddots & 0 \\ & & & & \ddots & & & 0 \\ & & & & \ddots & \ddots & & \vdots \\ \vdots & & & & & 0 & A_{N-1}^{N-1} & \gamma A_{N-1}^N \\ 0 & 0 & \cdots & & & \cdots & 0 & A_N^N \end{bmatrix}. \tag{6.1.7}$$

Two important results: We derive an important result of measuring $\tilde{u}(T)$ by conditional measurement on $\widetilde{\mathbf{y}}(T)$. Recall the (rescaled) Carleman vector would be

$$\widetilde{\mathbf{y}}(t) = [\widetilde{\mathbf{u}}(t), \widetilde{\mathbf{u}}^{\otimes 2}(t), ..., \widetilde{\mathbf{u}}^{\otimes N}(t)] \tag{6.1.8}$$

Thus

$$\|\tilde{\mathbf{y}}(t)\|^2 = \sum_{l=1}^{N} \|\tilde{\mathbf{u}}(t)\|^l = \|\tilde{\mathbf{u}}(t)\|^2 \left(\frac{1 - \|\tilde{\mathbf{u}}(t)\|^{2N}}{1 - \|\tilde{\mathbf{u}}(t)\|} \right) \tag{6.1.9}$$

For $\|\widetilde{\mathbf{u}}(t)\| < 1$,

$$\left(\frac{1 - \|\tilde{\mathbf{u}}(t)\|^{2N}}{1 - \|\tilde{\mathbf{u}}(t)\|} \right) < N \tag{6.1.10}$$

Thus,

$$\boxed{\|\tilde{\mathbf{y}}(t)\|^2 < \|\tilde{\mathbf{u}}(t)\|^2 N} \tag{6.1.11}$$

It is a remarkable result as it relates the norm of the Carleman vector to the norm of $\tilde{\mathbf{u}}(t)$ (the solution of the quadratic ODE). The probability of measuring $\tilde{\mathbf{u}}(t) := \tilde{\mathbf{y}}_1(t)$

$$\boxed{P(\tilde{\mathbf{u}}(t)) = \frac{\|\tilde{\mathbf{u}}(t)\|^2}{\|\tilde{\mathbf{y}}(t)\|^2} \geq \frac{1}{N}} \tag{6.1.12}$$

Thus, $O(1/(\sqrt{P(\tilde{\mathbf{u}}(t))}))$ rounds of amplitude amplification would boost the change of measuring $\tilde{\mathbf{u}}(t)$ to nearly 100 percent. We will extensively use Eq. (6.1.12) and Eq. (6.1.11) in the analysis of the runtime of the algorithms.

Note: We drop the tilde (symbol) above matrix A and vector $\mathbf{y}$ for convenience.

6.1.2 Method-I: matrix exponentiation method

Carleman ODE is a homogenous first-order matrix ODE with a time-independent matrix coefficient:

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = A\mathbf{y}, \qquad \mathbf{y}(0) = \mathbf{y}_{\text{in}} \tag{6.1.13}$$

Thus, a formal solution exists as follows (chapter 3 [MS11]):

$$\boxed{\mathbf{y}(t) = e^{At}\mathbf{y}(0)} \tag{6.1.14}$$

The two key insights from the algorithm are

- to efficiently approximate $e^{\mathbf{A}t}$ using the Chebyshev series
- to look for quantum speedup in the sparse matrix-vector multiplication

In the context of a Hermitian matrix H (say), Patel et al. [PP17] observed that a faster quantum algorithm does exist to approximate the expression $e^{-Ht}\mathbf{b}$. Here, the matrix H can be viewed as an operator in the Hilbert space. There are three requirements for an efficient algorithm:

1. The Hilbert space is a tensor product of many small components
2. The components have only local interactions that make H sparse and
3. Both matrix H and the vector b are specified in terms of a finite number of efficiently computable functions. Or, equivalently, oracles for matrix H and vector b are provided. □

Extending the algorithm for the Carleman ODE: Now we extend it to the Carleman matrix $\mathbf{A}$ having the following properties:

- a non-Hermitian matrix
- (3N) sparse matrix with local structure (see equation Eq. (5.1.4) to equation Eq. (5.1.12))
- explicit construction of the matrix V (i.e., $A = V\Lambda V^{-1}$) is possible (No-resonance

condition).

We adopt the [PP17] algorithms for the Carleman matrix. For this, we state the three crucial lemmas.

Lemma 8 *The Carleman matrix A can be linearly transformed to map its eigenvalues to the interval $[-1,\ 1]$.*

$$A \mapsto \frac{A - \frac{(\lambda_{\max}+\lambda_{\min})}{2} I}{\frac{\lambda_{\max}-\lambda_{\min}}{2}} \tag{6.1.15}$$

Lets define $\alpha := \frac{(\lambda_{\max}+\lambda_{\min})}{2}$ and $\beta := \frac{(\lambda_{\max}-\lambda_{\min})}{2}$. The scaling of time goes as

$$t \mapsto \beta t \tag{6.1.16}$$

It should be noted that the above scaling doesn't alter the diagonalization procedure due to the following lemma.

Lemma 9 *The Carleman matrix A is diagonalized as $A = V\Lambda V^{-1}$, then the same similarity transformation matrices V and V^{-1} will diagonalize the scaled matrix.*

$$A \mapsto \frac{A - \frac{(\lambda_{\max}+\lambda_{\min})}{2} I}{\frac{\lambda_{\max}-\lambda_{\min}}{2}} \tag{6.1.17}$$

Proof: $V(A/\beta - \alpha I/\beta)V^{-1} = V(A/\beta)V^{-1} - V(\alpha I/\beta)V^{-1} = \Lambda/\beta - \alpha I/\beta$ which is the diagonal matrix. □

Consequently, we use the same matrices V and V^{-1} without worrying about the scaling.

Lemma 10 *The minimum and maximum value of the Carleamn matrix are*

$$\lambda_{\max} = -4D(n+1)^2 sin^2\left(\frac{\pi}{2(n+1)}\right) + a \tag{6.1.18}$$

$$\lambda_{\min} = -N\left(4D(n+1)^2 sin^2\left(\frac{n\pi}{2(n+1)}\right) + a\right) \tag{6.1.19}$$

Proof: We work in a parameter regime with negative eigenvalues. Thus, the maximum eigenvalue of matrix A is the one with the **minimum magnitude** eigenvalue of $A_1^1 = F_1$. The maximum eigenvalue of matrix A is the one with the **maximum magnitude**

eigenvalue of A_N^N. □

Since $A = V\Lambda V^{-1}$, we express the analytic equation of the Carleman matrix as

$$e^{At} = Ve^{\Lambda}V^{-1} \tag{6.1.20}$$

Here,

$$e^{\Lambda t} = \begin{pmatrix} e^{\lambda_1 t} & & & & \\ & e^{\lambda_2 t} & & & \\ & & \ddots & & \\ & & & e^{\lambda_{n-1} t} & \\ & & & & e^{\lambda_d t} \end{pmatrix} \tag{6.1.21}$$

Express each of the exponential functions along the diagonal as Chebyshev series

$$e^{\lambda_i t} = \sum_{k=0}^{\infty} C_k(t) T_k(\lambda_i) \tag{6.1.22}$$

These are two crucial results regarding the series

Theorem 15 *In the Chebyshev series*

$$e^{\lambda_i t} = \sum_{k=0}^{\infty} C_k(t) T_k(\lambda_i) \tag{6.1.23}$$

The coefficients are given by the modified Bessel function of the first kind $I_k(x)$:

$$C_0 = \frac{1}{\pi}\int_0^{\pi} e^{-tcos\theta} d\theta = I_0(t), \tag{6.1.24}$$

$$C_{k>0} = \frac{2}{\pi}\int_0^{\pi} e^{-tcos\theta} cos(k\theta) d\theta = 2(-1)^k I_k(t) \tag{6.1.25}$$

Where,

$$I_{k(t)} = \sum_{m=0}^{\infty} \frac{(t/2)^{k+2s}}{s!(k+s)!} \tag{6.1.26}$$

Proof: It has been derived in Mathews & Walker's book [MW71] using complex analysis.□

Theorem 16 *([PP17]) Given a simulation time of t and an error tolerance of ε, the maximum truncation order is given by*

$$r = \frac{e^{5/4t}}{2} + ln(1/\varepsilon) = O(t + log(1/\varepsilon)) \tag{6.1.27}$$

Proof: It is proven in the paper. It is a direct consequence of Theorem 15. □

Algorithmic steps:

Step I: Rescale the Carleman matrix to bring its eigenvalue between $[-1,\ 1]$. It is the domain of the definition of Chebyshev polynomials. The method for rescaling is given in Lemma 8.

Step II: Construct the matrix V and V^{-1} using the method described in Chapter 4.

Step III: Based on user input of allowed error ε and simulation time, truncate the series up to order r using Theorem 16. It gives

$$e^{\lambda_i t} \approx [e^{\lambda_i t}]_r = \sum_{k=0}^{r} C_k(t) T_k(\lambda_i) \tag{6.1.28}$$

. The notation for truncated series is

$$exp(\Lambda T) \approx [exp(\Lambda T)]_r \tag{6.1.29}$$

Step IV: Do the quantum implementation of

$$\boxed{\mathbf{y}(T) = T[exp(\Lambda t)]_r T^{-1}\mathbf{y}(0)} \tag{6.1.30}$$

for determining the solution at some given time T.

Step V: Now, $O(1/(\sqrt{P(\tilde{\mathbf{u}}(t))})) = O(\sqrt{N})$ rounds of amplitude amplification (see Eq. (6.1.12)). The conditional measurement on $\mathbf{y}(T)$ would give $\mathbf{u}(T)$ with high probability.

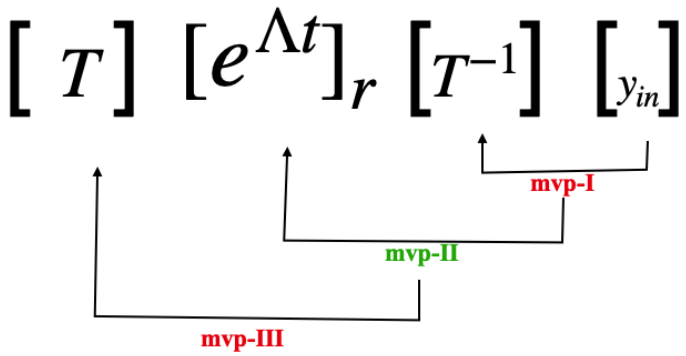

Figure 6.1: Step IV involves three matrix-vector products (mvp) where matrices can be viewed as operators acting on a state vector. There is a known quantum advantage in sparse matrix-vector structure. The mvp-II is efficient (see Theorem 17). However, it is unclear if we can have similar efficiency in computing mvp-I and II, where matrices V and V^{-1} are (dense) upper triangular matrices.

□

Two key results for estimating the cost of the matrix-vector product:

Theorem 17 *([PP17]) Let d be the size of the Carleamn matrix. The error tolerance is ε, and the Chebyshev truncation order is r. The quantum cost of implementing the matrix-vector product (II), i.e., $[e^{\Lambda t}]\boldsymbol{b}$ is*

$$O\left(log(d)\left(t + log\left(\frac{1}{\varepsilon}\right)\right) log^3\left(\frac{(t + log(1/\varepsilon))}{e^{-t}\varepsilon}\right)\right) \tag{6.1.31}$$

Proof: Use equation 59 of [PP17] on the sparse matrix-vector multiplication. The paper uses d for sparsity and $N = 2^n$ for matrix size. In our case, sparsity is one as $[e^{\Lambda t}]_r$ is the diagonal matrix. The size of the Carleman matrix is d. Thus substituting n=log(d) in equation of [PP17] gives the desired result. □

Theorem 18 *The computational cost of multiplying an upper triangular matrix M of size $d \times d$ with a vector $\boldsymbol{b}$ is*

1. *Quantum cost: $O(dlog(d))$*
2. *Classical cost: $O(d^2)$*

Proof: In the classical case, the vector **b** is multiplied by each d row of M. Each such operation requires d multiplication and addition. Hence, the complexity of O(d^2).

In the quantum case, For each d row and vector b, multiplication requires $\log(d)$ instructions. It is a keen to single-instruction-multiple-data (SIMD) paradigm of parallel computing. Here, the same instruction (gates) is given to the qubits (data) in superposition. See [PP17] for an elaborate discussion. □

Computational cost of each step:

- Step I:(Matrix rescaling) The matrix rescaling has a relatively nominal cost because there is a closed-form expression for computing the minimum and maximum eigenvalue of the Carleman matrix due to Lemma 10.
- Step II:(Diagonalization) The cost of finding the matrix V and V^{-1} is discussed in Chapter 5. The most expensive step is to compute the off-diagonal block elements. (See Fig. 5.2). We don't immediately see any advantage in computing this step on a quantum computer. Thus, it can be computed on a classical computer as part of classical pre-processing.
- Step III:(Finding the truncation order) Estimating the Chebyshev series truncation order r has a relatively nominal cost.
- Step IV:(matrix-vector product) As per Theorem 18 and Theorem 17, the leading order contribution of various parameters in the gate complexity are $O(t \text{ polylog}(t))$, $O(\text{dlog}(d))$ and $O(\text{ploylog}(1/\varepsilon))$. **Note:** This will output the quantum Carleman vector $|\mathbf{y}(T)\rangle$.
- Step V:(Amplitude amplification) As per Eq. (6.1.12), $O(1/(\sqrt{P(\tilde{\mathbf{u}}(t))})) = O(\sqrt{N})$ rounds of amplitude amplification would boost the change of measuring $\tilde{\mathbf{u}}(t)$ (by the conditional measurement on $\mathbf{y}(T)$) to nearly 100 percent.

Conclusion: We present method-I as a hybrid Classical-Qauntum algorithm. The scaling of the gate complexity on parameters t and ε is considered good. The scaling of the gate complexity with matrix size is $d(\log(d))$. Currently, We can't find any local sub-structures in the matrix V that a quantum algorithm could harness to reduce the gate complexity polylogarithmic in d.

6.2 Method-II: Psuedo Spectral Method

Prelude: The high-level description of the algorithm is very similar to the other Quantum linear system-based algorithms of Chapter 3. The key difference is the Carleman ODE is solved using the Chebyshev series-based method rather than the Taylor series or the Euler method.

6.2.1 Main result

Outline:

- We state the main theorem of [CL19]. It is given in the broader context of a first-order matrix differential equation. If certain conditions are fulfilled, the theorem provides quantum gate complexity for solving such an ODE.
- Then we prove why the Carelman ODE fulfills all the theorem requirements. Thus, we can use the Psuedo-spectral algorithm that outputs the desired quantum encoding of the solution $|\mathbf{y}(T)\rangle$.
- We use the conditional measurement to get $|\mathbf{u}(T)\rangle$ from the Carleman vector $|\mathbf{y}(T)\rangle$

Theorem 19 *(Adapted from theorem 1 of [CL19]) Consider a homogenous ODE problem*

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = A\mathbf{y}; \quad \mathbf{y}(0) = \mathbf{y}_{\text{in}} \tag{6.2.1}$$

Assume a s-sparse matrix $A \in \mathbb{R}^{d\times d}$ can be diagonalized as $A = V\Lambda V^{-1}$ where $\Lambda = diag(\lambda_1, \ldots, \lambda_d)$ with $\mathrm{Re}(\lambda_i) \leqslant 0$ for each $i \in d$. Let the condition number of the matrix V be κ_V. Then, there exists a quantum algorithm that produces ϵ-close quantum state $|\mathbf{y}(T)\rangle/\|\mathbf{y}(T)\|$ succeeding with probability $\Omega(1)$ using

$$O\left(\kappa_V s\|A\|Tq\,\mathrm{poly}\left(\log\left(\frac{\kappa_V s\|\mathbf{y}_{\text{in}}\|\|A\|T}{\epsilon g}\right)\right)\right) \tag{6.2.2}$$

queries to oracles O_A and $O_{\mathbf{y}_{\text{in}}}$. And parameters

$$g := \|\mathbf{y}(T)\|, \qquad q := \max_{t\in[0,T]} \frac{\|\mathbf{y}(t)\|}{\|\mathbf{y}(T)\|}. \tag{6.2.3}$$

The overall gate complexity is larger than the query complexity by a factor of

$$\text{poly}\left(\log\left(\frac{\kappa_V s(d)\|A\|\|\mathbf{y}_{\text{in}}\|T}{\epsilon}\right)\right) \tag{6.2.4}$$

Proof: This follows from Theorem 1 of [CL19]. The paper is in the context of a time-dependent matrix $A(t)$. In our case, the matrix A is time-independent. We have replaced the parameter

$$g' := \max_{t\in[0,T]} \max_{n\in\mathbb{N}} \|\hat{x}^{(n+1)}(t)\| \tag{6.2.5}$$

with the following bound coming from Lemma 3 and equation 4.21 in [CL19]

$$\max_{t\in[0,T]} \max_{n\in\mathbb{N}} \|\hat{x}^{(n+1)}(t)\| \leq \kappa_V \|\mathbf{y}_{\text{in}}\| \tag{6.2.6}$$

to get the above result. □

As we have seen, the Carleman Matrix fulfills the requirements mentioned in the theorem.

- All its eigenvalues are non-positive (due to Theorem 10)
- It is Diagonalizable ($A = V\Lambda V^{-1}$) (due to Theorem 11)
- It is (3N)-sparse where N is the truncation order and s is the sparsity of matrix F_1. (due to Lemma 3)

The gate complexity demands the computation of the following parameters of the problem in terms of the parameters of the quadratic ODE problem.

- Spectral norm : $\|A\|$
- s: sparsity of A
- d : the size of A
- condition number κ_V
- $g := \|\mathbf{y}(T)\|$, and
- $q := \max_{t\in[0,T]} \frac{\|\mathbf{y}(t)\|}{\|\mathbf{y}(T)\|}$.

We estimate them as follows:

Estimation of spectral norm of A: $\|A\|$ can be simplified in terms of system parameters. In the context of the reaction-diffusion equation, [Liu23] has estimated $\|A\|$ as follows.

Lemma 11 *(Equation 4.19 and 4.20 in [Liu23]) For the Carleman matrix A, its spectral norm is bounded by*

$$\|A\| \leq N(\|F_1\| + \|F_2\|) \tag{6.2.7}$$

The spectral norm of F_1 and F_2 are given as

$$\|F_1\| = 4D(n+1)^2 + a \tag{6.2.8}$$

$$\|F_2\| = b \tag{6.2.9}$$

Estimation of the sparsity of A: The sparsity of the Carleman A is given in terms of the sparsity of the matrix F_1 and the Carleman truncation order as

Lemma 12 *(Equation 4.19 and 4.20 in [Liu23]) For an N-th order truncated Carleman matrix A, its sparsity $s = 3N$.*

Proof: It is discussed in Lemma 3. The sparsity of matrix $F_1 = 3L_h + aI$ is three due to the sparsity of the discrete Laplacian matrix.

The size of A: It is given by Lemma 4. It scales as $d = O(n^N)$, where n is no. of discretization point while, N is the Carleman truncation order.

Estimating parameter g: We estimate it using the Eq. (6.1.11). Since, $g := \|\mathbf{y}(T)\|$, putting $t = T$ in Eq. (6.1.11) implies

$$\boxed{g := \|\mathbf{y}(T)\| \leq \|\mathbf{u}(T)\|\sqrt{N}} \tag{6.2.10}$$

Estimating parameter q: We again estimate it using the Eq. (6.1.11). Since,

$$q := \max_{t\in[0,T]} \frac{\|\mathbf{y}(t)\|}{\|\mathbf{y}(T)\|} \leq \max_{t\in[0,T]} \frac{\|\mathbf{u}(t)\|}{\|\mathbf{u}(T)\|} \tag{6.2.11}$$

Since the system is dissipative (negative eigenvalues), $\mathbf{u}(t) < \mathbf{u}_{\text{in}}$. Combining it with the

above results gives

$$\boxed{q := \max_{t\in[0,T]} \frac{\|\mathbf{y}(t)\|}{\|\mathbf{y}(T)\|} \leq \max_{t\in[0,T]} \frac{\|\mathbf{u}(t)\|}{\|\mathbf{u}(T)\|} \leq \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}} \tag{6.2.12}$$

Compiling the results: We put all the above-estimated quantities into the gate complexity of Theorem 19. It implies the number of Oracle calls to O_A and $\mathbf{y}_{\text{in}}$

$$O\left(\kappa_V(3N)[N(4D(n+1)^2+a+b)]T\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}\,\text{poly}\log\left(\frac{\kappa_V(3N)(\|\mathbf{u_{in}}\|\sqrt{N})[N(4D(n+1)^2+a+b)]T}{\epsilon\mathbf{u}(T)}\right)\right. \tag{6.2.13}$$

and the gate complexity is larger by a factor of

$$\text{poly}\log\left(\frac{\kappa_V(3N)(n^N)[N(4D(n+1)^2+a+b]\|\mathbf{u}_{\text{in}}\|\sqrt{N}T}{\epsilon}\right) \tag{6.2.14}$$

Overall scaling: The leading order contribution of various parameters in the gate complexity are $O(T\ \text{polylog}(T))$, $O(N^2\text{poly}(\log(N^{1.5}n^N)))$ and, $O(\text{ploy}(\log(1/\varepsilon)))$.

Remark: A bound on the condition number of V in terms of system parameter would also be very useful as it appears in the expression on the gate complexity. We have not found a conclusive upper bound. But we approached as follows.

An estimate on the condition number of matrix V:

Approach: In Chapter 5, the iterative construction of matrix $V \in \mathbb{R}^{d\times d}$ is by computing a set of linearly independent vectors for the Carleman matrix A. Let $V = [e_1, ..., e_d]$. We have an explicit form for each e_j. It uses Guggenheimer's result [GEJ95] on the condition number.

$$\kappa(V) < \frac{2}{|Det(V)|}\left(\frac{\|V\|_F}{\sqrt{d}}\right)^d \tag{6.2.15}$$

Forbenius norm owes

$$\|V\|_F = \sum_{i=1}^{d} ||e_i||_2 \tag{6.2.16}$$

The determinant of V can be estimated as it is equal to the product of the determinant of the diagonal block matrices [MS4]. It is not obvious how to give an upper bound on $\|A\|_F = \sum_{i=1}^{d} ||e_i||_2$. We leave it as a future exercise. □

6.2.2 Outline of Psuedo-spectral method

For the sake of completeness, we briefly summarize the Psuedo-spectral method. The key goal is to solve the homogenous ODE

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = A\mathbf{y}, \qquad \mathbf{y}(0) = \mathbf{y}_{\text{in}} \tag{6.2.17}$$

The goal is to find $\mathbf{y}(t)$ for $t \in [0,\ T]$.

Key idea: Consider a truncated Chebyshev-series approximation for solution $\mathbf{y}(t)$ as

$$\mathbf{y}_i(t) \approx \hat{\mathbf{y}}_i(t) = \sum_{k=0}^{r} c_{i,k} T_k(t), \quad i \in [d]_0 := \{0, 1, \ldots, d-1\} \tag{6.2.18}$$

There are two useful properties of the above consideration.

- Special points exist in the domain $t \in [0,\ T]$ where evaluating the Chebyshev polynomial has marginal cost. These points are called Chebyshev-Gauss-Lobatto quadrature (CGL) nodes, $t_l = \cos\frac{l\pi}{r}$ for $l \in [r+1]_0$. Note, the value of Chebyshev polynomial at CGL nodes $T_k(t_l) = \cos\frac{kl\pi}{r}$.
- For $\forall i \in [d]_0$, the solution components $\mathbf{y}_i(t)$ satisfy the ODE and initial conditions at these CGL nodes $\{t_l\}_{l=0}^{n}$. It would give a system of linear equations solved for the coefficients.

Determining the Chebyshev truncation order: The criteria for selecting the Chebyshev truncation order r is due to the following theorem.

Lemma 13 (Lemma 2 of [CL19]) *Let $\mathbf{y}(t) \in C^{\infty}(-1, 1)$ be the analytic solution of an ordinary differential equation and let the numerical solution be $\hat{\mathbf{y}}(t)$ that satisfies*

the ODE and initial condition for $\{t_l = \cos\frac{l\pi}{n}\}_{l=0}^{n}$. *Then*

$$\max_{t\in[-1,1]} \|\hat{\mathbf{y}}(t) - \mathbf{y}(t)\| \leqslant \sqrt{\frac{2}{\pi}} \max_{t\in[-1,1]} \|\mathbf{y}^{(r+1)}(t)\| \left(\frac{e}{2r}\right)^r. \tag{6.2.19}$$

If the matrix A is diagonalizable, $A = V\Lambda V^{-1}$, then equation 4.21 of [CL19] gives

$$\|\hat{\mathbf{y}}(T) - \mathbf{y}(T)\| \leqslant m\kappa_V \|\mathbf{y}_{\text{in}}\| \left(\frac{e}{2r}\right)^r \tag{6.2.20}$$

where, $m \geq \frac{\|A\|T}{2}$. Thus, expressing the below formula gives an estimate of the condition number in terms of time T and error ε are given.

$$\|\hat{\mathbf{y}}(T) - \mathbf{y}(T)\| \leqslant \left(\frac{\|A\|T}{2}\right) \kappa_V \|\mathbf{y}_{\text{in}}\| \left(\frac{e}{2r}\right)^r \tag{6.2.21}$$

Conclusion: With respect to parameter N, the matrix size d, time T, and error ϵ, the gate complexity of the Psuedo-spectral method is $O(\kappa_V\ N^2\ T\ \text{polylog}(dTN/\varepsilon)\ )$. A definite upper bound on κ_V will make the actual scaling more explicit.

Chapter 7

Chebyshev simulation result

7.1 Runge-Kutta 45 method for the quadratic ODE

We are solving the quadratic ODE using the inbuilt solver in MATLAB (matlab:ode45) gives the following 3D graph for the solution $\mathbf{u}(x,t)$.

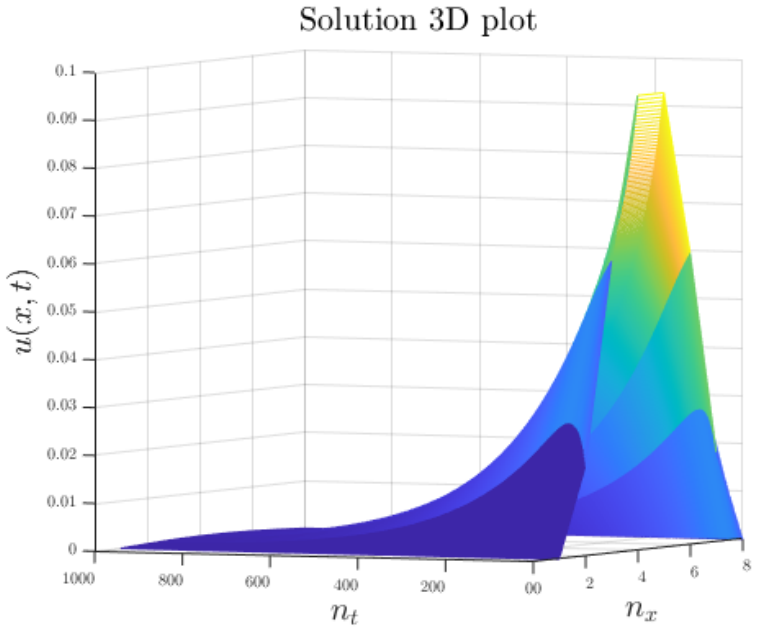

Figure 7.1: (Runge-kutta 45 method) Solution for reaction diffusion equation for $D = 0.2$, $a = 0.4$ *and* $b = -1$. The initial distribution $u(x, t = 0) = 0.1sin^2(\pi x)$. Position domain $x \in [0, 1]$ is discretized into 8 points while time $t \in [0, 3]$ is discretized into 1000 points

.

7.2 Solving the quadratic ODE: Carleman linearization + matrix exponentiation (method-I)

Solving the Fisher-KPP using the Carlman linearization and Chebyshev series gives the following 3D graph for the solution $\mathbf{u}(x,t)$.

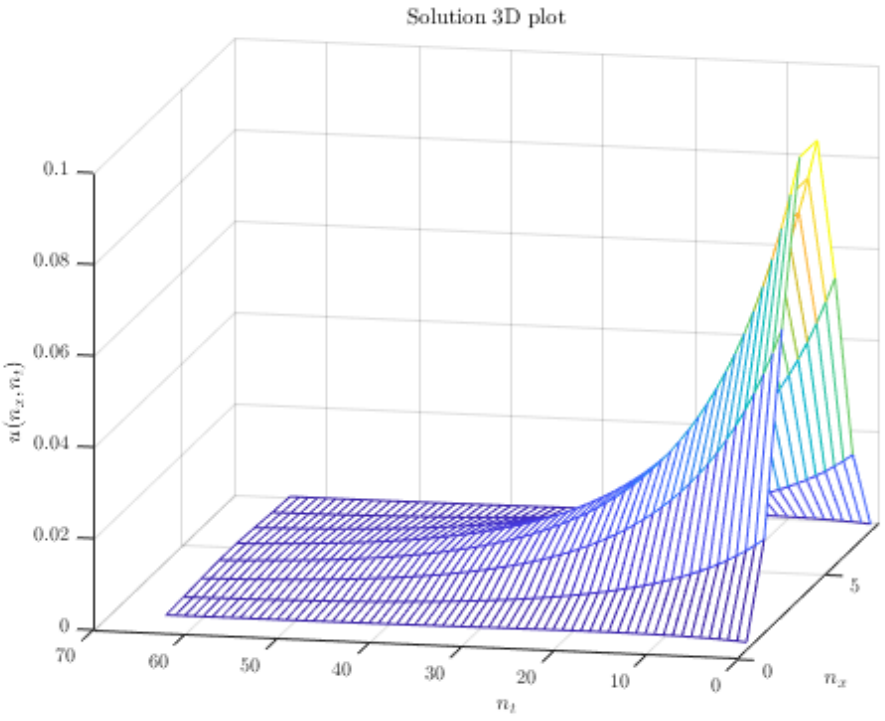

Figure 7.2: (Chebyshev-series method) Solution for reaction diffusion equation for $D = 0.2$, $a = 0.4$ *and* $b = -1$. The initial distribution $u(x, t = 0) = 0.1sin^2(\pi x)$. Position domain $x \in [0, 1]$ is discretized into 8 points. For simulation, the function is approximated at 64 points (to get the resolution) in time domain $t \in [0, 3]$. [See our **Github code**: Here]

7.3 Absolute error: method-I

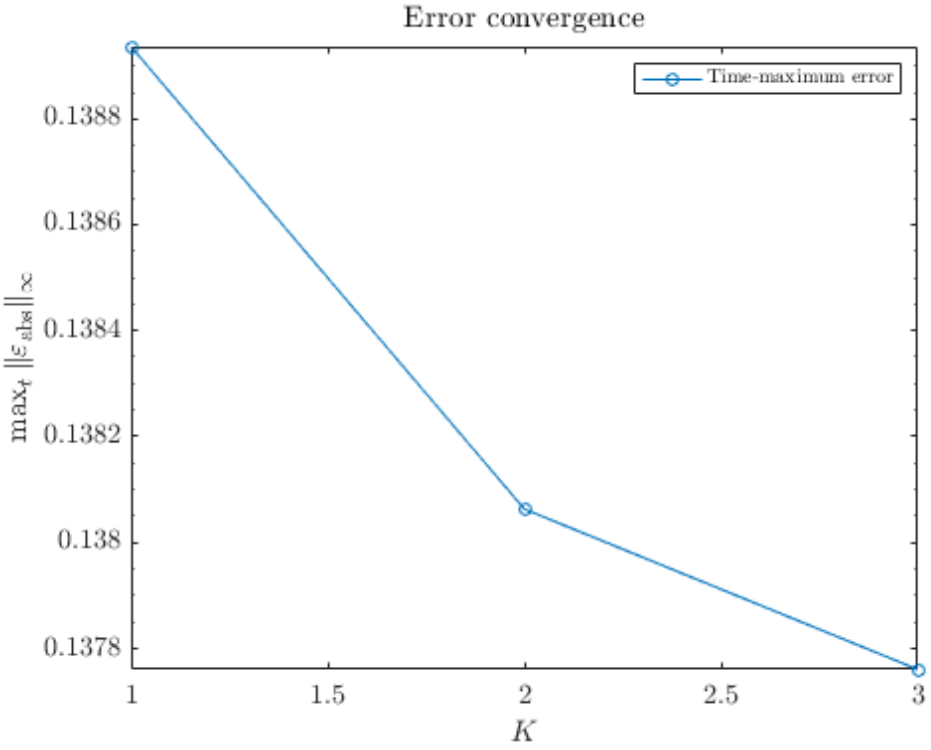

Figure 7.3: (Absolute Error vs Chevyshev truncation order K

The Carleman truncation is fixed at $N = 3$ while the Chebyshev truncation order is varied from $K = 1$ to 3.

Remark: As clear from the algorithm pipeline (Fig. 1.1), several approximation techniques have been employed to obtain the final solution. It includes the error due to finite difference discretization, Carleman truncation, and the Chebyshev method used to solve the ODE. An absolute error graph (Fig. 7.3) wouldn't differentiate between them. A separate controlled error analysis is required to distinguish between individual errors.

Chapter 8

Conclusion

8.1 Hardness of the problem: tractable and intractable regime

Parameter R: We have seen an efficient quantum algorithm in simulation time T exists if parameter $R < 1$. Recently, It has been proven that for $R \geq 1$, hard instances exist where the worst-case time complexity scales exponentially in T.

Theorem 20 *([LEN23]) Assume $R > 1$. Then there is an instance of the quantum quadratic ODE problem such that any quantum algorithm for producing a quantum state approximating the solution $|u(T)\rangle$ with bounded error must have worst-case time complexity exponential in T.* □

Thus, $R < 1$ is considered a tractable regime for the problem.

The lower bound in time T: Now within the tractable regime. The dependency of the complexity can't be made sublinear in T. It is due to the No-fast forwarding theorem for Hamiltonian simulation that puts a lower bound of $\Omega(T)$ [BCO17]. □

Lower bound in time ε: In the context of Hamiltonian simulation, the query complexity has a lower bound in precision ε ([BCC14]) as

$$\Omega\left(\frac{log(1/\varepsilon)}{log(log(1/\varepsilon))}\right) \tag{8.1.1}$$

Lower bound in condition number κ: For a quantum linear system problem ([HHL09]),

it has been noted that any quantum algorithm with query complexity sublinear in κ would imply $BQP = PSPACE$. Thus, $\Omega(\kappa)$ is conjectured to be the lower bound. □

The hardness of the quadraticODE problem: The quadratic ODE problem is BQP-hard.

Proof: In quadartic ODE problem

$$\frac{\mathrm{d}u}{\mathrm{d}t} = F_2 u^{\otimes 2} + F_1 u, \qquad u(0) = u_{\text{in}}. \tag{8.1.2}$$

Let's take the special case of matrix $F_2 = \mathbf{0}$, and F_1 being anti-hermitian; then, it implies Schrodinger's equation. Producing the quantum state $|u(T)\rangle$, given $|u(0)\rangle$, is as hard as a quantum Hamiltonian simulation. Thus, any efficient classical algorithm for this problem would imply $BQP = P$. □

8.2 Comparison of gate complexity of the four algorithms

We consider three (crucial) parameters to compare the gate complexity of the algorithms mentioned in the paper

Algorithm	Time (T)	Error (ε)	Carleman order (N)
Euler [Liu23]	$T^2\text{polylog}(T)$	$\frac{1}{\varepsilon}\text{polylog}(\frac{1}{\varepsilon})$	$N^3\alpha^N\text{polylog}(N)$
Taylor[CSM23]	$T\text{polylog}(T)$	$\text{polylog}(\frac{1}{\varepsilon})$	$N^2\text{polylog}(N)$
Method-I (Matrix Exponentiation)	$T\text{polylog}(T)$	$\text{polylog}(\frac{1}{\varepsilon})$	$\text{N}\beta^N\text{polylog(N)}$ [‡]
Method-II (Psuedo-spectral)	$T\text{polylog}(T)$	$\text{polylog}(\frac{1}{\varepsilon})$	$N^2\kappa_V\text{polylog}(N\beta^N)$ [†]

Remarks:

‡ Matrix exponentiation: We presented it as a hybrid classical-quantum algorithm. For a given d-sized Carleman matrix, the complexity is $O(d\log(d))$. It corresponds to the most expensive step: the (dense) triangular matrix-vector product. The Carleman matrix size depends on the number of discretization points (n) and truncation order N as $d = O(n^N)$. It implies $O(d\log(d)) = O(n^N N\ \log(n))$. (Note: $\beta = n$ in the table.)

† For the pseudo-spectral method, the upper bound on the condition number of V is unclear. It could add to the complexity dependency on N. The Carleman matrix size is substituted as $d = O(n^N)$.

• symbol $\alpha = \|u_{\text{in}}\|$.

8.3 Open problem related to two new algorithms

There are four major open problems related to this work. I list them in the descending order of my priority.

- What is an upper bound on the condition number of matrix V? **Remark**: How to give this upper bound is unclear. I have tried in a way. But computing an upper bound on the Forbenius norm seems challenging as of now.
- To mathematically prove or disprove Hypothesis 1 on No-resonance for the Carleman matrix. **Remark**: The No-resonance condition seems to hold for several combinations of the parameters. We cannot find a counter-example or a case when the condition fails. Here is an ongoing discussion on StackExchange on it [Man24].
- Is there some (local) tensor product structure in the matrix V? **Remark**: If we have such a useful feature, then the matrix-vector product can be done efficiently using a quantum computer.
- To extend the analysis to mixed non-linear ODE. **Remark**: We dealt with quadratic ODE in this paper. Our analysis on diagonalization can readily be generalized to a higher degree non-linear ODE, like

$$\frac{\mathrm{d}u}{\mathrm{d}t} = F_M u^{\otimes M} + F_1 u, \qquad u(0) = u_{\text{in}}. \tag{8.3.1}$$

Because the Carleman matrix has the form

$$A = \begin{pmatrix} A_1^1 & 0 & \cdots & 0 & A_M^1 & & \\ & A_2^2 & \ddots & & \ddots & \ddots & \\ & & \ddots & \ddots & & \ddots & A_N^{N-M+1} \\ & & & \ddots & \ddots & & 0 \\ & & & & \ddots & \ddots & \vdots \\ & & & & & A_{N-1}^{N-1} & 0 \\ & & & & & & A_N^N \end{pmatrix} \tag{8.3.2}$$

Due to Theorem 12, its eigenvalue entirely depends on the eigenvalues of the block matrices along the main diagonal. Thus, changing the degree of polynomial has no impact on the eigenvalues of the new Carleman matrix. It does impact the similarity transformation matrix V. Again, the iterative procedure for diagonalization can be given because the matrix is still a bi-diagonal block matrix.

The problem of diagonalization requires more careful consideration if there are mixed terms in the non-linear ODE, like

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \sum_{k=2}^{M} F_k u^{\otimes k} + F_1 u, \qquad u(0) = u_{\text{in}}. \tag{8.3.3}$$

Now, the Carleman matrix is no longer a block bi-diagonal matrix. It appears the iterative procedure for diagonalization needs to be changed substantially. It can be seen as an open problem related to this work. □

References

[Fey82] R. P. Feynmann. Simulating Physics with Computers. International Journal of Theoretical Physics, Volume 21, Issue 6-7, pp. 467-488 https://link.springer.com/article/10.1007/BF02650179

[Lyo96] S. Lloyd. Universal quantum simulators. Science Vol 273, Issue 5278 pp. 1073-1078 https://www.science.org/doi/10.1126/science.273.5278.1073

[Liu21] Jin-Peng Liu, Herman Øie Kolden, Hari K. Krovi, Nuno F. Loureiro, Konstantina Trivisa, and Andrew M. Childs. Efficient quantum algorithm for dissipative non-linear differential equations. PNAS2021 Vol. 118 No. 35. https://www.pnas.org/doi/full/10.1073/pnas.2026805118

[Liu23] Jin-Peng Liu, Dong An, Di Fang, Jiasu Wang, Guang Hao Low and Stephen Jordan. Efficient Quantum Algorithm for Non-linear Reaction-Diffusion Equations and Energy Estimation. *Commun. Math. Phys. 404, 963–1020 (2023).* https://doi.org/10.1007/s00220-023-04857-9

[PP17] Apoorva Patel and Anjani Priyadarsini. Efficient quantum algorithms for state measurement and linear algebra applications. *International Journal of Quantum Information 2018 16:06* https://doi.org/10.1142/S021974991850048X

[CL19] Childs, A.M., Liu, JP. Quantum Spectral Methods for Differential Equations. Commun. Math. Phys. 375, 1427–1457 (2020). https://doi.org/10.1007/s00220-020-03699-z

[CSM23] Pedro C. S. Costa, Philipp Schleich, Mauro E. S. Morales, Dominic W. Berry.

Further improving quantum algorithms for non-linear differential equations via higher-order methods and rescaling. https://arxiv.org/abs/2312.09518v1 (2023)

[WWX24] Hsuan-Cheng Wu, Jingyao Wang, Xiantao Li. Quantum Algorithms for Nonlinear Dynamics: Revisiting Carleman Linearization with No Dissipative Conditions. https://arxiv.org/abs/2405.12714v1 (2024)

[KS91] K Kowalski and W-H Steeb. Non-linear Dynamical Systems and Carleman Linearization (1991). ISBN: 978-981-4506-34-2

[BCO17] Dominic W. Berry, Andrew M. Childs, Aaron Ostrander, Guoming Wang. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. Communications in Mathematical Physics 356, 1057-1081 (2017) https://arxiv.org/abs/1701.03684v2

[Har22] Hari Krovi. Improved quantum algorithms for linear and non-linear differential equations. Quantum 7, 913 (2023). https://arxiv.org/abs/2202.01054v4 (2024)

[DLW22] Dong An, Jin-Peng Liu, Daochen Wang, Qi Zhao. A theory of quantum differential equation solvers: limitations and fast-forwarding. https://arxiv.org/abs/2211.05246v2

[Kre98] Raimer Kress, *Numerical analysis*, Springer (1998).

[MS11] Moya-Cessa, H.; Soto-Eguibar, F. (Chapter 3) (2011). Differential Equations: An Operational Approach. New Jersey: Rinton Press. ISBN 978-1-58949-060-4.

[Pow81] Powell, M. J. D. (1981). "7: The theory of minimax approximation". Approximation Theory and Methods. Cambridge University Press. ISBN 0521295149.

[GRM07] Christian Grossmann; Hans-G. Roos; Martin Stynes (2007). Numerical Treatment of Partial Differential Equations. Springer Science & Business Media. p. 23. ISBN 978-3-540-71584-9.

[Gri96] Peter Grindrod. The theory and applications of reaction-diffusion equations: Patterns and waves. The Clarendon Press Oxford University Press, New York, second edition, (1996)

[CKS15] Andrew M. Childs, Robin Kothari, Rolando D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision https://arxiv.org/abs/1511.02306v2 (v2; 2017)

[BC22] Dominic W. Berry and Pedro C. S. Costa, Quantum algorithm for time-dependent differential equations using Dyson series https://arxiv.org/abs/2212.03544v2

[CAS22] Pedro C.S. Costa, Dong An, Yuval R. Sanders, Yuan Su, Ryan Babbush, and Dominic W. Berry. Optimal scaling quantum linear-systems solver via discrete adiabatic theorem. PRX Quantum, 3:040303, Oct 2022. DOI: 10.1103/PRXQuantum.3.040303.

[TL89] Christos A Tsiligiannis and Gerasimos Lyberatos. Normal forms, resonance, and bifurcation analysis via the Carleman linearization. Journal of mathematical analysis and applications, 139(1):123–138, 1989

[MS1] user:krm2233 (Mathematics Stack Exchange) https://math.stackexchange.com/a/4870629/474528

[MS2] user:leonbloy (Mathematics Stack Exchange) https://math.stackexchange.com/a/21461/474528

[MS3] user:egreg (Mathematics Stack Exchange) https://math.stackexchange.com/a/2316569/474528

[MW71] Jon Mathews, Robert L. Walker - Mathematical Methods of Physics-W.A. Benjamin (1971)

[GEJ95] Guggenheimer, Edelman, and Johnson (1995) https://nhigham.com/2021/06/08/bounds-for-the-matrix-condition-number/

[MS4] user:okie (Mathematics Stack Exchange) https://math.stackexchange.com/q/1184825/474528

[LEN23] Dylan Lewis, Stephan Eidenbenz, Balasubramanya Nadiga, Yiğit Subaşı Limitations for Quantum Algorithms to Solve Turbulent and Chaotic System https://arxiv.org/pdf/2307.09593

[BCC14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, Rolando D. Somma. Exponential improvement in precision for simulating sparse Hamiltonians https://arxiv.org/abs/1312.1414v2

[HHL09] Aram W. Harrow, Avinatan Hassidim, Seth Lloyd. Quantum algorithm for solving linear systems of equations https://arxiv.org/abs/0811.3171

[Man24] user: manish kumar https://math.stackexchange.com/q/4924878/474528

[MS5] user:Calle (Mathematics Stack Exchange) https://math.stackexchange.com/a/19468/474528