LDE Quantum Algorithm Note

Euan Mendoza

October 21, 2025

Contents

Let $\mathbb{F}$ be a field (either $\mathbb{R}$ or $\mathbb{C}$), and $\mathbb{F}^n$ is a n-dimensional vector space over $\mathbb{F}$.

1 Summary and Key Ideas

- Quantum computers can numerically solve LDE's in $O(\log n)$ time where n is the dimension of a vector space, which takes $O(n^3)$ time classically.

- This provides an exponential speed up which is due to quantum computers being able to efficiently solve systems of linear equations.

- This speedup has some caveats, such as whether an exponential speedup for solving linear differential equations outweighs the costs of using quantum computers in the first place. Is it even that hard to solve LDE's classically or is the performance manageable (is waiting a couple hours for a computer to run ok).

- How much time does it take to solve an LDE on a classical CAS like SageMath?

- The cost of decomposing a non-unitary matrix as a product of unitaries. Computational cost of Gram-Schmidt?

2 The Algorithm

Our goal is to solve the Harmonic oscilator equation using a quantum algorithm,

$$f''(x) + \omega^2 f(x) = 0 \tag{1}$$

with initial conditions $f(x) = 1, f''(x) = 1$. This can be rewritten as follows,

Let $\alpha = f(x)$ and $\beta = f'(x)$. By rewriting Equation 1 as $f''(x) = \omega^2 f(x)$ we obtain the following, $\frac{d}{dx}\beta = f''(x) = -\omega f(x)$. For a vector $v = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, we obtain the following differential equation.

$$\frac{d}{dx}v = Av \tag{2}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \tag{3}$$

This equation can be solved symbolically to give us the following equation,

$$v(x) = e^{Ax}v(0) \tag{4}$$

where $v(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ as per our initial equations.

Notice that when the Harmonic frequency $\omega = 1$, our matrix A is unitary, in this case we denote the matrix $A = U$.

We can numerically approximate $v(x)$ to an approximation order k by truncating it's Taylor series as follows,

$$v(x) \approx \sum_{m=0}^{k} \frac{(Ut)^m}{m!} v(0) \tag{5}$$

2.1 The Quantum Circuit

2.1.1 Goal

Let $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, and $u = f(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Then we want to solve the equation,

$$f(t) \approx \sum_{m=0}^{k} \frac{(At)^m}{m!} f(0) \tag{6}$$

This equation can be described by quantum states as follows. Let $|u\rangle = \sum_{j\in\{0,1\}} \frac{u_j}{||u||} |j\rangle = |+\rangle$, and since A is unitary $||A|| = 1$, which we can denote $U = A$.

$$|f(t)\rangle \approx \sum_{m=0}^{k} \frac{||u||(Ut)^m}{m!} |u\rangle \tag{7}$$

Then if we set a constant $c_{m=||u||\frac{t^m}{m!}}$, we can obtain the following equation which we claim can be simulated efficiently on a quantum computer.

$$|f(t)\rangle \approx \frac{1}{c^4} \sum_{m=0}^{k} c_m U^m |u\rangle \tag{8}$$

where $c = \sqrt{\sum c_m}$.

2.1.2 TODO The Circuit

Let U_u be a unitary matrix that turns the state $|\varphi\rangle$ into $|u\rangle$ and V is the following $k+1 \times k+1$ matrix,

$$V = \frac{1}{c^2} \begin{bmatrix} \sqrt{c_0} & Q & Q & Q & Q \\ \sqrt{c_1} & Q & Q & Q & Q \\ \sqrt{c_2} & Q & Q & Q & Q \\ \vdots & Q & Q & Q & Q \\ \sqrt{c_k} & Q & Q & Q & Q \end{bmatrix} \tag{9}$$

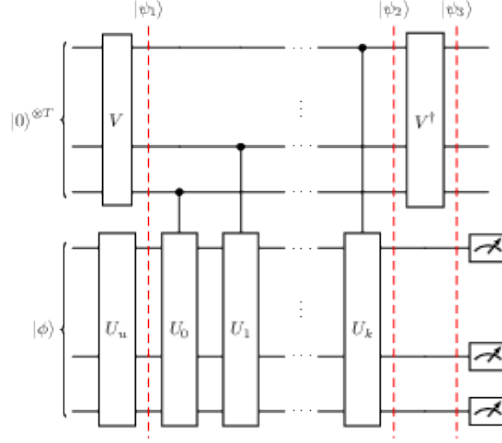where Q pads V to make it into a unitary.

Figure 1: The quantum circuit for solving the harmonic oscillator LDE

1. Encoding: The circuit shown in Figure 1 has $T = \log k$ ancilla qubits and a state $|\varphi\rangle$, with state $|\psi_{in}\rangle = |0\rangle^{\otimes T} \otimes |\varphi\rangle$. U_u turns $|\varphi\rangle$ into $|u\rangle$, and V leaves the system in the following state,

$$|\psi_1\rangle = \frac{1}{c^2} \sum_{m=0}^{k} \sqrt{c_m}|m\rangle \otimes |u\rangle \tag{10}$$

3 Linear Differential Equations Background

Recall that a differential equation that is written in terms of it's derivative. For example Schrodinger's equation is a differential equation,

$$i\hbar \frac{\partial}{\partial t}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle \tag{11}$$

A linear differential equation (LDE) is an equation where each derivative

$$c_0 f(x) + c_1 \frac{df(x)}{dx} + c_2 \frac{df^2(x)}{dx^2} \cdots c_{k-1} \frac{df^{k-1}(x)}{dx^{k-1}} + c_k = 0 \tag{12}$$

Solving LDE's is incredibly important in material science, physics and economics.

Since solving differential equations plays such an important role in many fields of mathematics, science and economics, there are several tools that are classically well suited to solving differential equations.

4 Differential Equations and Computation

Differential equations form the backbone of many real world tasks in material science and quantitative finance. Therefore in computational algebra, there has been many different methods studied to efficiently solve differential equations.

One of the first problems is understanding what it means to be *efficient* in solving linear differential equations. There are many different approaches to solving LDE's where efficiency differs depending on the use case. They can be solved symbolically/algebraically or numerically for example.

There are several computer algebra systems that are quite good at solving differential equations, such as sagemaths, USyd Magma, Wolfram Mathematica and Matlab for example.

4.1 Quantum Advantage

- Mathematica use in material science