From Symmetric Toeplitz Hamiltonians to Quantum Circuits

Rayan Trabelsi, supervised by Benoit Valiron

Université Paris Saclay, LMF, Gif-sur-Yvette, France.

Abstract

This work introduces a quantum circuit synthesis methodology for simulating the unitary evolution under a subclass of symmetric Toeplitz Hamiltonians by decomposing them into specific diagonal matrices M_k. These M_k are then classified, to achieve significant simplification, into power of two bands ($k = 2^m$) and congruence classes where the matrices coefficients are equal. Finally, we construct the explicit simulation circuit for the 1D discrete Poisson equation.

Contents

1 Introduction

Solving large systems of linear equations of the form $Hx = b$ is a fundamental problem across many fields of science. Quantum computing offers a potential speedup for these problems through algorithms like the Harrow-Hassidim-Lloyd (HHL) algorithm [4]. The core of the HHL algorithm relies on simulating the time evolution of a quantum system described by the Hamiltonian H which is achieved by implementing the unitary operator e^{-iHt}. This approach is effective because the eigenvectors of the matrix H are the same as those of e^{iHt}, and their eigenvalues are directly related. A significant class of these problems, particularly those arising from the discretization of differential equations, can be described by Hamiltonians represented as symmetric Toeplitz matrices.

In this work, we present a systematic methodology for the decomposition of specific symmetric Toeplitz matrices and the synthesis of their quantum circuits to simulate their corresponding time evolution. We address the challenge by first decomposing the target matrix into a sum of simpler M_k matrices, each corresponding to a specific diagonal. We then develop circuit construction techniques for two key cases: M_k where k is a power of two, and congruence classes where the coefficients of M_k are equal. The latter case reveals a remarkable simplification, where the sum of M_k matrices reduces to a simple tensor product of Pauli operators. The overall simulation is then constructed by approximating the total evolution using a Trotter-Suzuki decomposition. Finally, the practical utility of this approach is demonstrated by applying it to a canonical problem in physics: the simulation of the one dimensional Poisson equation.

2 Preliminaries

In this section, we first recall the mathematical tools needed, then present the structure and notation of quantum circuits.
We begin by defining Hilbert spaces and quantum states (Section 2.1), then introduce basic operators, gates and their properties (Section 2.2.1), and conclude by illustrating simple quantum circuit constructions (Section 3.3.1).

2.1 Hilbert Spaces and States

A Hilbert space is a complete vector space with respect to the norm $x \to \langle x, x\rangle^{\frac{1}{2}}$, in the sense that any Cauchy sequence converges to a defined limit within the space, its inner product is defined by :

- $\overline{\langle v, w\rangle} = \langle w, v\rangle$, this implies that $\langle w, w\rangle$ is a real number.
- $\langle \lambda v_1 + v_2, \beta w\rangle = \bar{\lambda}\beta\langle v_1, w\rangle + \beta\langle v_2, w\rangle$ with $\lambda, \beta \in \mathbb{C}$
- $\langle x, x\rangle \geq 0$

In quantum information theory, we work with finite-dimensional Hilbert spaces $\mathbb{H}^n$ ($n \in \mathbb{N}$), since each qubit resides in its own Hilbert space and we have a finite number of qubits. We denote the canonical computational basis of $\mathbb{H}^2$ by

$$\{|0\rangle\,,\,|1\rangle\}$$

In Dirac notation, a "ket" $|\psi\rangle$ represents a column vector. Its conjugate transpose $\langle\psi| = (|\psi\rangle)^\dagger$, is called a "bra".

A general qubit pure state is written

$$|\psi\rangle = a\ |0\rangle + b\ |1\rangle$$

where $a, b \in \mathbb{C}$ and the normalization condition $|a|^2 + |b|^2 = 1$ ensures $\langle\psi|\psi\rangle = 1$.

When the system is in the state $|\psi\rangle$, the outcome of a measurement is either 0 with $|a|^2$ probability or 1 with $|b|^2$ probability.

2.2 Operators and Quantum circuits

2.2.1 Operators

An operator in quantum mechanics is a linear map from a Hilbert space to itself, it describes the evolution (or other transformations) of a given system. For instance, if an initial state is $|\psi\rangle$, a unitary operator U evolves it into

$$|\phi\rangle \ = \ U\ |\psi\rangle$$

An operator U is called unitary if it preserves inner products. This implies that, for any two states $|\psi\rangle\,,|\phi\rangle$

$$\langle U\psi \mid U\phi\rangle \ = \ \langle\psi \mid \phi\rangle$$

Equivalently if

$$U\,U^\dagger \ = \ U^\dagger\,U \ = \ I$$

In the case of time evolution under a time independent Hamiltonian H (which is Hermitian, $H = H^\dagger$), the Schrodinger equation

$$i\,\frac{d}{dt}\,|\psi(t)\rangle \ = \ H\ |\psi(t)\rangle$$

has the solution

$$|\psi(t)\rangle \ = \ U(t)\ |\psi(0)\rangle$$

with

$$U(t) \ = \ e^{-\,i\,H\,t}$$

Thus any unitary operator can be considered as the exponential of a hermitian operator:

$$U = e^{-\,i\,H\,t}, \quad H = H^\dagger$$

To introduce the elementary operators, consider a spin $\frac{1}{2}$ system. By definition [2], a measurement of its angular momentum along any axis gives only the two eigenvalues $\pm\frac{\hbar}{2}$. We denote the corresponding operators by $\hat{S}_x$, $\hat{S}_y$, and $\hat{S}_z$. These are related to the Pauli matrices $\sigma_x, \sigma_y, \sigma_z$ via

$$\hat{S}_{x,y,z} \;=\; \frac{\hbar}{2}\,\sigma_{x,y,z}$$

where the Pauli matrices are

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

These satisfy the commutation and anticommutation relations

$$\begin{aligned} [\sigma_i, \sigma_j] \;&=\; 2\,i\,\varepsilon_{ijk}\,\sigma_k \\ \{\sigma_i, \sigma_j\} \;&=\; 2\,\delta_{ij}\,I \end{aligned}$$

where ε_{ijk} is the Levi Civita symbol and δ_{ij} the Kronecker delta.

Pauli matrices and the identity form a basis of $\mathbb{M}_{2\times 2}(\mathbb{C})$, and any $2^n \times 2^n$ matrix can be expressed using a linear combination of their tensor products. For $A \in \mathbb{M}_{2^n\times 2^n}(\mathbb{C})$, we have :

$$A = \sum_{i_1,\ldots,i_n} c_{i_1\ldots i_n} \bigotimes_{k=1}^{n} \sigma_{i_k}, \quad \sigma_{i_k} \in \{I, \sigma_x, \sigma_y, \sigma_z\},\ c_{i_1\ldots i_n} \in \mathbb{C}$$

From a computer science point of view, the Pauli matrices act as elementary quantum gates. In particular, σ_x is the analogue of the classical NOT gate:

$$\begin{aligned} \sigma_x\ |0\rangle &= |1\rangle \\ \sigma_x\ |1\rangle &= |0\rangle \end{aligned}$$

The phase flip gate σ_z acts by

$$\sigma_z\ |x\rangle \;=\; (-1)^x\ |x\rangle$$

for $x \in \{0, 1\}$.

We also introduce the Hadamard gate, denoted by H. Its matrix representation is

$$H \;=\; \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

and it transforms the computational basis states as

$$\begin{aligned} H\ |0\rangle &= \tfrac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) \\ H\ |1\rangle &= \tfrac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \end{aligned}$$

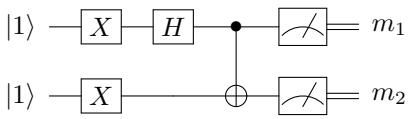
2.2.2 Quantum Circuits

Quantum memories cannot execute arbitrary unitaries at once. Instead, it is restricted to a finite set of basic unitary operations, called gates. These gates act on one or multiple qubits at a time and are represented graphically in circuit diagrams, where each gate symbol specifies the corresponding unitary transformation.

Each horizontal line (or "wire") represents the state of one qubit, and time flows from left to right. Gates are placed on these wires to indicate operations:

- A box labeled X applies a NOT to the qubit.
- A box labeled H applies a Hadamard gate to the qubit.
- A dot connected to a box (or another dot) implements controlled operations (e.g. CNOT).
- A meter symbol denotes measurement in the computational basis.

For example, the following two qubit circuit prepares a Bell state and then measures both qubits:



Here:

- The top wire starts in $|1\rangle$, receives a NOT (as if it started in $|0\rangle$), receives a Hadamard (H), then acts as control for a CNOT.
- The bottom wire is the target of that CNOT.
- Finally, both qubits are measured giving classical bits m_1 and m_2.

3 Symmetric Toeplitz and Band Matrix Decomposition and Circuit Synthesis

In this section, we study a special class of hermitian matrices : symmetric Toeplitz matrices, and we develop quantum circuits to synthesize its unitary evolution. We begin by formally defining symmetric Toeplitz and band matrices (Section 3.1). We then decompose these matrices into sums of simpler components M_k (Section 3.2), leveraging their tensor product structure. Finally, we derive explicit quantum circuits (Section 3.3) to implement $e^{-iM_k t}$.

3.1 Definitions

Symmetric Toeplitz and Band Matrices

A Toeplitz matrix [1] is a $n \times n$ matrix A whose entries are constant along each diagonal:

$$A_{i,j} = A_{i+1,j+1} = A_{i-1,j-1} \quad \text{for all valid } i, j$$

A symmetric Toeplitz matrix satisfies $A_{i,j} = A_{j,i}$, so its entries depend only on $|i-j|$:

$$A_{i,j} = a_{|i-j|}$$

$$A = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ a_1 & a_0 & a_1 & \ddots & \vdots \\ a_2 & a_1 & a_0 & \ddots & a_2 \\ \vdots & \ddots & \ddots & \ddots & a_1 \\ a_{n-1} & \cdots & a_2 & a_1 & a_0 \end{pmatrix}$$

A band matrix has nonzeros confined to a diagonal band: if its lower bandwidth is p and upper bandwidth q, then

$$A_{i,j} = 0 \quad \text{whenever } j < i - p \text{ or } j > i + q.$$

Combining both gives a symmetric Toeplitz band matrix with bandwidth m ($p = q = m$):

$$A_{i,j} = \begin{cases} a_{|i-j|}, & |i-j| \le m \\ 0, & |i-j| > m \end{cases}$$

and in full form

$$A = \begin{pmatrix} a_0 & a_1 & \cdots & a_m & 0 & \cdots & 0 \\ a_1 & a_0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 & a_2 & \ddots & 0 \\ a_m & \ddots & a_1 & a_0 & a_1 & \ddots & \vdots \\ 0 & \ddots & a_2 & a_1 & a_0 & \ddots & a_m \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & a_1 \\ 0 & \cdots & 0 & \cdots & a_m & a_1 & a_0 \end{pmatrix}$$

A particular case is the tridiagonal one ($m = 1$) :

$$A = \begin{pmatrix} a_0 & a_1 & 0 & \cdots & 0 \\ a_1 & a_0 & a_1 & \ddots & \vdots \\ 0 & a_1 & a_0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & a_1 \\ 0 & \cdots & 0 & a_1 & a_0 \end{pmatrix}$$

3.2 Decomposition

Let A be an $N \times N$ symmetric Toeplitz matrix with $N = 2^n$. A is also Hermitian ($A^\dagger = A$):

$$A = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{N-1} \\ a_1 & a_0 & a_1 & \ddots & \vdots \\ a_2 & a_1 & a_0 & \ddots & a_2 \\ \vdots & \ddots & \ddots & \ddots & a_1 \\ a_{N-1} & \cdots & a_2 & a_1 & a_0 \end{pmatrix}$$

To implement the unitary e^{-iAt}, we first decompose A into a sum of "band" matrices,

$$A = \sum_{k=0}^{N-1} M_k$$

where each

$$M_k = a_k \sum_{i=0}^{N-1-k} \left(|i\rangle \langle i+k| + |i+k\rangle \langle i| \right)$$

$$M_0 = \begin{pmatrix} a_0 & 0 & 0 & \cdots & 0 \\ 0 & a_0 & 0 & \ddots & \vdots \\ 0 & 0 & a_0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & a_0 \end{pmatrix}, M_1 = \begin{pmatrix} 0 & a_1 & 0 & \cdots & 0 \\ a_1 & 0 & a_1 & \ddots & \vdots \\ 0 & a_1 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & a_1 \\ 0 & \cdots & 0 & a_1 & 0 \end{pmatrix}, M_2 = \begin{pmatrix} 0 & 0 & a_2 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ a_2 & 0 & 0 & \ddots & a_2 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & a_2 & 0 & 0 \end{pmatrix} \quad \cdots$$

Since $e^{-iM_0 t}$ is diagonal and trivial to implement, we focus on M_k for $k \geq 1$.
Let's start by an example which will make it easier to generalize later.
Take $N = 8$:

$$M_1 = a_1 \left(\begin{array}{cccc|cccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right).$$

We find that :

$$M_1 \;=\; a_1 \left(\underline{\; I \otimes I \otimes X \;} \;+\; P\, R_1\, P^{-1} \right)$$

where

$$P = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad R_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} X & 0_2 & 0_2 & 0_2 \\ 0_2 & X & 0_2 & 0_2 \\ 0_2 & 0_2 & X & 0_2 \\ 0_2 & 0_2 & 0_2 & 0_2 \end{pmatrix}$$

Since R_1 and PR_1P^{-1} describe the same endomorphism, they share the same characteristic polynomial proving the existence of such matrix P (see figure 1). Moreover, $P^{-1} = P^T$.
In addition, R_1 is $I \otimes I \otimes X$ but with its last non-null 2×2 block being null. When structured into diagonal blocks, R_1 becomes straightforward to implement.

$$R_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \rightarrow PR_1P^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1}$$

Applying the same logic to the other bands that are power of 2, considering appropriate block size:

- For $k = 2$, using 4×4 blocks: $M_2 \;=\; a_2 \,(\, I \otimes X \otimes I \;+\; P^2\, R_2\, (P^{-1})^2 \,)$
- For $k = 4$, using 8×8 blocks: $M_4 \;=\; a_4 \,(\, X \otimes I \otimes I \,)$

This pattern leads to a general conclusion for bands that are powers of 2 for $2^n \times 2^n$ symmetric Toeplitz matrices:

$$\text{For } k = 2^m \begin{cases} \text{If } m < n-1, & M_k \;=\; a_k \,(\, I^{\otimes(n-m-1)} \otimes X \otimes I^{\otimes m} \;+\; P^k\, R_k\, (P^{-1})^k \,) \\ \text{If } m = n-1, & M_{2^{n-1}} \;=\; a_{2^{n-1}} \,(\, X \otimes I^{\otimes(n-1)} \,) \end{cases}$$

In each case, the notation $I^{\otimes m} \otimes X \otimes I^{\otimes(n-m-1)}$ means an n-fold tensor product with X at the m-th position (counting from 0)

We now consider the remaining bands for $k > 0$ (not necessarily powers of 2) and distinguish cases based on the parity of k. Each M_k can be expressed as a sum of tensor products involving the Pauli X operator and permutation matrices. For clarity, we list a few examples in an $N = 2^n$ dimensional space. We use the same decrementer permutation matrix P and block matrix R_k as before.

- $M_1 = a_1\big(I^{\otimes(n-1)} \otimes X + \; PR_1P^{-1} \;\big)$
- $M_2 = a_2\big(I^{\otimes(n-2)} \otimes X \otimes I + P^2R_2P^{-2}\big)$
- $M_3 = a_3\big[I^{\otimes(n-2)} \otimes X \otimes X \;-\; PR_1P^{-1} \;+ P^2R_3P^{-2} \;+\; (PR_3P^{-1} + P^3R_3P^{-3} - R_1^*)\;\big]$.
  Here R_1^* denotes the matrix R_1 with its first 2×2 block set to zero (as in our notation above)
- $M_4 = a_4\big(I^{\otimes(n-3)} \otimes X \otimes I \otimes I + P^4R_4P^{-4}\big)$
- $M_5 = a_5\big[I^{\otimes(n-3)} \otimes X \otimes I \otimes X + P^4R_5P^{-4} \;-\; (PR_3P^{-1} + P^3R_3P^{-3} - R_1^*) \;+ (PR_5P^{-1} + P^5R_5P^{-5} - R_3^* + R_1^*)\big]$, where R_3^* is R_3 with its first 2×2 block set to zero, etc.

Each term above shows the decomposition of M_k into a "base" tensor product operator (with X's in certain positions) plus additional terms involving the permutation P and R_j matrices. The exact pattern of these extra terms depends on the bitwise structure of k and ensures the correct structure of M_k. In practice, one can verify these formulas for $n > 3$ by comparing with the definition of M_k.

An important observation arises when groups of coefficients a_k are equal, the P dependent terms cancel in the sum, leaving only the simple tensor product terms determined by the binary patterns. For example, write each index k in binary and interpret each bit "1" as applying X on that qubit and each "0" as applying the identity I

In the 3 qubit case ($n = 3$, $N = 8$), if $a_1 = a_3 = a_5 = a_7 = c$, the odd indices $1, 3, 5, 7$ have binary forms 001, 011, 101, 111. Replacing $1 \to X$ and $0 \to I$ gives the operators $I \otimes I \otimes X$, $I \otimes X \otimes X$, $X \otimes I \otimes X$, and $X \otimes X \otimes X$. One checks that

$$M_1 + M_3 + M_5 + M_7 = c(I \otimes I \otimes X + I \otimes X \otimes X + X \otimes I \otimes X + X \otimes X \otimes X)$$

All P dependent terms vanish in the sum. Similarly, if we let $a_2 = a_6$ in the $N = 8$ case (indices $2 = 010$ and $6 = 110$), then

$$M_2 + M_6 = a_2(I \otimes X \otimes I + X \otimes X \otimes I)$$

More generally, this pattern holds for other congruence classes of indices. For instance, if $a_2 = a_6 = a_{10} = a_{14} = \cdots$ all coincide, one sums over indices whose binary form ends in "10" (in the two least significant bits) or indices congruent to 2 mod 4. Replacing $1 \to X$ and $0 \to I$ in those bitstrings again gives the expected result. Likewise, if $a_4 = a_{12} = a_{20} = \cdots$ (indices congruent to 4 mod 8), then the sum reduces to the tensor products determined by binary endings "100".

For any fixed $1 \leq j < n$, consider the congruence class

$$C_j \;=\; \{\, k : 1 \leq k < 2^n,\; k \equiv 2^{\,j-1} (\mathrm{mod}\, 2^j)\}$$

Writing each k in its n-bit binary expansion

$$k = (b_n b_{n-1} \ldots b_1)_2$$

and setting

$$X^{b_i(k)} = \begin{cases} X, & b_i(k) = 1 \\ I, & b_i(k) = 0 \end{cases}$$

one obtains the following sums whenever all a_k with $k \in C_j$ coincide:

(i) $j = 1$ (odd indices):

$$\sum_{\substack{1 \leq k < 2^n \\ k \equiv 1 \,(\mathrm{mod}\, 2)}} M_k \;=\; a_1 \sum_{\substack{1 \leq k < 2^n \\ k \equiv 1 \,(\mathrm{mod}\, 2)}} \bigotimes_{i=1}^{n} X^{b_i(k)}$$

(ii) $j = 2$ (indices $\equiv 2 \pmod 4$):

$$\sum_{\substack{1 \leq k < 2^n \\ k \equiv 2 \,(\mathrm{mod}\, 4)}} M_k \;=\; a_2 \sum_{\substack{1 \leq k < 2^n \\ k \equiv 2 \,(\mathrm{mod}\, 4)}} \bigotimes_{i=1}^{n} X^{b_i(k)}$$

(iii) $j = 3$ (indices $\equiv 4 \pmod 8$):

$$\sum_{\substack{1 \leq k < 2^n \\ k \equiv 4 \,(\mathrm{mod}\, 8)}} M_k \;=\; a_4 \sum_{\substack{1 \leq k < 2^n \\ k \equiv 4 \,(\mathrm{mod}\, 8)}} \bigotimes_{i=1}^{n} X^{b_i(k)}$$

For $N = 8$:

$$\textbf{(i):} \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}, \quad \textbf{(ii):} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

In general, for $1 \leq j < n$,

$$\sum_{\substack{1 \leq k < 2^n \\ k \equiv 2^{\,j-1} \,(\mathrm{mod}\, 2^j)}} M_k \;=\; a_{2^{\,j-1}} \sum_{\substack{1 \leq k < 2^n \\ k \equiv 2^{\,j-1} \,(\mathrm{mod}\, 2^j)}} \bigotimes_{i=1}^{n} X^{b_i(k)}$$

3.3 Circuit Synthesis

At this stage, we decomposed A into, power of 2 matrices, and sums of M_k that depend on their congruence class. A or variants of A can expressed as a general tensor product of Pauli matrices and P dependent terms. Once exponentiated, the unitary evolution e^{-iAt} reduces to a sequence of straightforward rotation gates interleaved with permutation operations. When the exponential terms do not commute, we use Trotterization [3] to approximate their combined exponential:

$$e^{P+Q} \;\approx\; \left(e^{P/v}\, e^{Q/v}\right)^v$$

The larger the u is, the better approximation we get.

3.3.1 Power of 2

We've seen that if $k = 2^m$ with $m < n-1$, then

$$M_k \;=\; a_k \left(I^{\otimes(n-m-1)} \otimes X \otimes I^{\otimes m} \;+\; P^k\, R_k\, (P^{-1})^k \right)$$

We will denote

$$\pi_1 = I^{\otimes(n-m-1)} \otimes X \otimes I^{\otimes m}$$
$$\pi_2 = P^k\, R_k\, P^{-k}$$

The unitary is then expressed as :

$$U_k(t) \;=\; e^{-it\, M_k} \;=\; e^{-it\, a_k\, (\,\pi_1 \,+\, \pi_2\,)}$$

Since $[\pi_1, \pi_2] \neq 0$

$$U_k(t) \;=\; e^{-it\, a_k\, \pi_1}\; e^{-it\, a_k\, \pi_2} + O([\pi_1, \pi_2])$$

Using Trotterization

$$U_k(t) \;\approx\; (e^{-i\frac{t}{v}\, a_k\, \pi_1}\; e^{-i\frac{t}{v}\, a_k\, \pi_2})^v$$

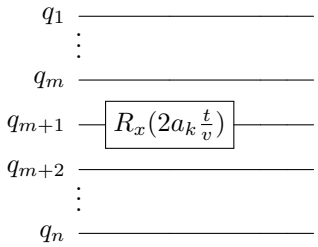
U_k is expressed as the product of two unitaries : $U_{\pi_1} = e^{-i\frac{t}{v}\, a_k\, \pi_1}$ and $U_{\pi_2} = e^{-i\frac{t}{v}\, a_k\, \pi_2}$.

Implementation of U_{π_1}

The circuit of U_{π_1} is given by :

$$U_{\pi_1}(t) = e^{-i\frac{t}{v} a_k\, \pi_1} = I^{\otimes(n-m-1)} \otimes e^{-i\frac{t}{v} a_k X} \otimes I^{\otimes m} = I^{\otimes(n-m-1)} \otimes R_x(2\frac{t}{v} a_k) \otimes I^{\otimes m}$$

The unitary acts as a single qubit rotation on q_{m+1}.

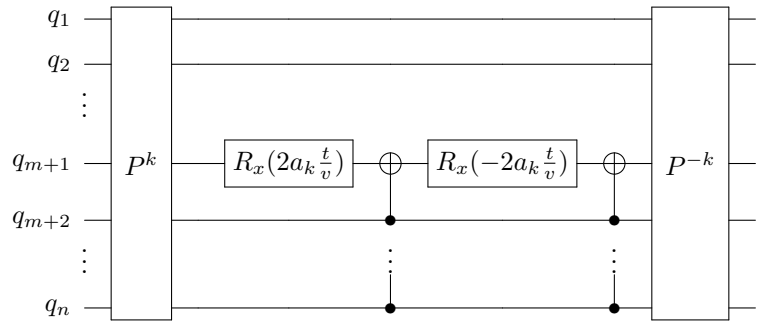


Implementation of U_{π_2}

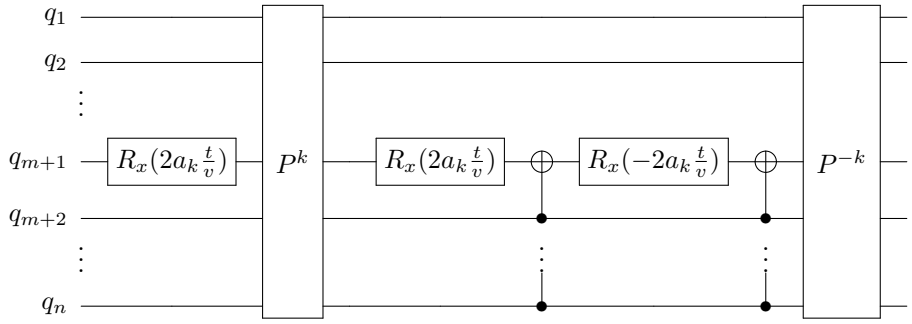
The circuit of U_{π_2} is given by :

$$U_{\pi_2}(t) = e^{-i\frac{t}{v} a_k\, P^k\, R_k\, P^{-k}} = P^k\; e^{-i\frac{t}{v}\, a_k\, R_k}\; P^{-k}$$

Since P is a decrementer, P^k decrements k times and P^{-k1} increments k times



Therefore, the final circuit is obtained by repeating the following subroutine u times:



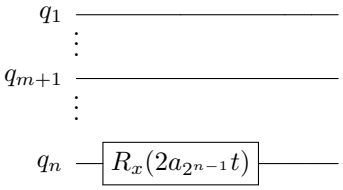
If $k = 2^{n-1}$, then

$$M_{2^{n-1}} \;=\; a_{2^{n-1}}\,(\,X\,\otimes\,I^{\otimes(n-1)}\,)$$

Its unitary is expressed by :

$$U_{2^{n-1}}(t) \;=\; e^{-i\,M_{2^{n-1}}} \;=\; e^{-it\,a_{2^{n-1}}\,(\,X\otimes I^{\otimes(n-1)}\,)} \;=\; e^{-it\,a_{2^{n-1}}\,X}\otimes I^{\otimes(n-1)} \;=\; R_x(2\,a_{2^{n-1}}\,t)\otimes I^{\otimes(n-1)}$$

Thus, its circuit is :



3.3.2 General Congruence Class

For any fixed $1 \le j < n$, consider the congruence class

$$C_j \;=\; \{\,k : 1 \le k < 2^n,\; k \equiv 2^{\,j-1} (\mathrm{mod}\, 2^j)\,\}$$

the sum for all $k \in C_j$ is :

$$\sigma = \sum_{\substack{1\le k<2^n \\ k\equiv 2^{j-1}(\mathrm{mod}\,2^j)}} M_k = a_{2^{j-1}} \sum_{\substack{1\le k<2^n \\ k\equiv 2^{j-1}(\mathrm{mod}\,2^j)}} \bigotimes_{i=1}^{n} X^{b_i(k)}$$

The binary constraint $k \equiv 2^{j-1} \pmod{2^j}$ fixes the j least significant bits to $(b_j, \ldots, b_1) = (1, 0, \ldots, 0)$. Thus, the general term in the sum is:

$$X^{b_n} \otimes X^{b_{n-1}} \otimes \cdots \otimes X^{b_{j+1}} \otimes X \otimes I^{\otimes(j-1)}$$

Consider the sum over all possible combinations of $b_{j+1}, \ldots, b_n$. This is equivalent to summing over all possible tensor products where the last element is $X \otimes I^{\otimes(j-1)}$.
Thus, σ can be represented as :

$$\sigma \;= a_{2^{j-1}} \sum_{b_n\in\{0,1\}} \cdots \sum_{b_{j+1}\in\{0,1\}} \left(X^{b_n} \otimes X^{b_{n-1}} \otimes \cdots \otimes X^{b_{j+1}} \otimes X \otimes I^{\otimes(j-1)} \right)$$

This sum can be factored due to the linearity of the tensor product:

$$\sigma \;= a_{2^{j-1}} \left(\sum_{b_n \in \{0,1\}} X^{b_n} \right) \otimes \left(\sum_{b_{n-1} \in \{0,1\}} X^{b_{n-1}} \right) \otimes \cdots \otimes \left(\sum_{b_{j+1} \in \{0,1\}} X^{b_{j+1}} \right) \otimes X \otimes I^{\otimes (j-1)}$$

Since $\sum_{b \in \{0,1\}} X^b = X^0 + X^1 = I + X$, it simplifies to:

$$\sigma \;=\; a_{2^{j-1}} \underbrace{(I+X) \otimes (I+X) \otimes \cdots \otimes (I+X)}_{n-j \text{ times}} \otimes X \otimes I^{\otimes (j-1)}$$

Therefore, the unitary becomes in this case :

$$U_\sigma(t) \;=\; e^{-it\sigma} \;=\; e^{-it a_{2^{j-1}} \, (I+X)^{\otimes (n-j)} \otimes X \otimes I^{\otimes (j-1)}}$$

One finds using $X = 2\,H\,Z\,H$:

$$I + X \;=\; 2\,H\,|0\rangle\,\langle 0|\,H$$

Hence

$$(I+X)^{\otimes (n-j)} \otimes X = 2^{\,n-j}\, H^{\otimes (n-j+1)}\, (|0\rangle\,\langle 0|^{\otimes (n-j)} \otimes Z)\, H^{\otimes (n-j+1)}$$

so that

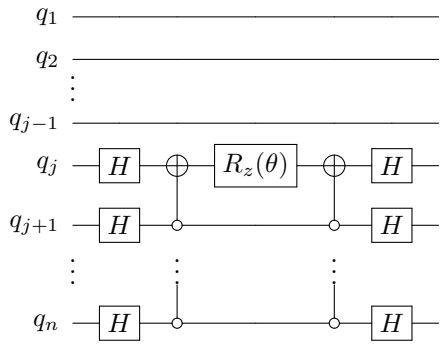
$$\sigma = a_{2^{j-1}}\, 2^{\,n-j}\, (H^{\otimes (n-j+1)} \otimes I^{\otimes (j-1)})\, (|0\rangle\,\langle 0|^{\otimes (n-j)} \otimes Z \otimes I^{\otimes (j-1)})\, (H^{\otimes (n-j+1)} \otimes I^{\otimes (j-1)})$$

Thus

$$U_\sigma(t) \;=\; (H^{\otimes (n-j+1)} \otimes I^{\otimes (j-1)})\, e^{-i\, a_{2^{j-1}}\, 2^{\,n-j}\, t\, (|0\rangle\langle 0|^{\otimes (n-j)} \otimes Z \otimes I^{\otimes (j-1)})}\, (H^{\otimes (n-j+1)} \otimes I^{\otimes (j-1)})$$

The exponential is an $(n-j)$ controlled R_z on qubit q_{n-j+1}, with angle

$$\theta = 2\, a_{2^{j-1}}\, 2^{\,n-j}\, t = a_{2^{j-1}}\, 2^{\,n-j+1}\, t$$



Complexity Analysis

Let $w = n - m$. The circuit of M_k, when k is a power of 2, is dominated by the cost of the P. For a single Trotter step, the cost of implementing a $w-$controlled rotation is $O(w)$ gates and $O(log(w))$ depth using ancilla qubits. Therefore, the circuit's complexity depends on the adder type, and can be summarized in the table below:

Adder Type	$\mathbf{G}_{\text{step}}$	$\mathbf{D}_{\text{step}}$	$\mathbf{G}_{\text{tot}} = \mathbf{u}\,\mathbf{G}_{\text{step}}$	$\mathbf{D}_{\text{tot}} = \mathbf{u}\,\mathbf{D}_{\text{step}}$
Ripple-carry	$O(w)$	$O(w)$	$O(v\,w)$	$O(v\,w)$
Carry-lookahead	$O(w)$	$O(\log w)$	$O(v\,w)$	$O(v \log w)$

Table 1: Circuit gate and depth complexities per Trotter step and in total on an n qubit register.[5]

The carry-lookahead adder provides a depth advantage for each M_k but at the cost of additional ancilla qubits.
When $k = 2^{n-1}$, it becomes a special case where the complexity is $O(1)$ since it is a single rotation.

The gate and depth complexity of the circuit associated to each congruence class C_j is $O(n-j)$ at the cost of $O(n-j)$ ancilla qubits. Without ancilla qubits, it requires $O((n-j)^2)$ gates and depth $O((n-j)^2)$.

4 Application

Symmetric Toeplitz matrices appear in many contexts ranging from the Black-Scholes equation in finance to quantum machine learning and quantum simulation. They commonly arise when discretizing differential equations using finite difference methods. Their structure, particularly their bandwidth, depends on the order of the chosen approximation. To illustrate this, we consider the case of the Poisson equation.

Discrete Poisson Equation

Let's consider the 1D Poisson equation with Dirichlet boundary conditions:

$$\Delta\phi = -\rho(x), \quad x \in [0, L], \quad \phi(0) = \phi(L) = 0$$

We discretize the domain $(0, L)$ into K intervals of width $h = L/(K+1)$, resulting in K interior grid points $x_i = ih$ for $i = 1, ..., K$. The discrete field values are $\phi_i = \phi(x_i)$, with $\phi_0 = 0$ and $\phi_{K+1} = 0$ due to the boundary conditions.

The central difference approximation for the second derivative at grid point i is:

$$\left.\frac{d^2\phi}{dx^2}\right|_{x_i} \approx \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2}$$

Applying this to the Poisson equation at each interior grid point, we get:

$$\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2} = -\rho_i, \quad i = 1, ..., K$$

This system of K linear equations can be written in matrix form $\mathcal{L}\phi = -\rho$, where $\phi = (\phi_1, ..., \phi_K)^T$ and $\rho = (\rho_1, ..., \rho_K)^T$. The discrete Laplacian matrix $\mathcal{L}$ is the coefficient matrix of ϕ and is given by the tridiagonal system: [6]

$$\mathcal{L} = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 \end{pmatrix}$$

which is a symmetric Toeplitz band matrix where the bandwidth $b = 1$.
The unitary evolution $e^{-i\mathcal{L}t}$ is equal to $e^{-i\mathcal{L}_D t}e^{-i\mathcal{L}_1 t}$ since $[\mathcal{L}_D, \mathcal{L}_1] = 0$ with

$$\mathcal{L}_D = \frac{1}{h^2}\begin{pmatrix} -2 & 0 & & \\ 0 & -2 & 0 & \\ & \ddots & \ddots & \ddots \\ & & 0 & -2 \end{pmatrix}, \quad \mathcal{L}_1 = \frac{1}{h^2}\begin{pmatrix} 0 & 1 & & \\ 1 & 0 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & 0 \end{pmatrix}$$

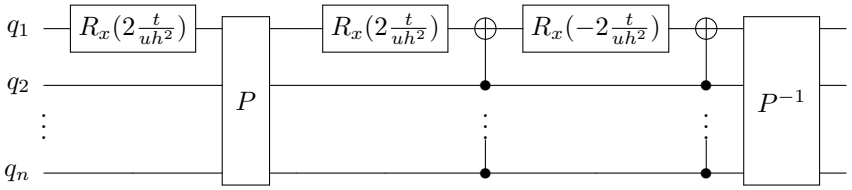
$e^{-i\mathcal{L}_D t}$ corresponds to a global phase, which has no effect on the circuit and can therefore be omitted in the implementation.
$\mathcal{L}_1$ is the M_1 matrix up to a coefficient. Thus, using the result of Section 3.3.1, for $m = 0$

$$\mathcal{L}_1 = \frac{1}{h^2}\left(I^{\otimes(n-1)} \otimes X + P R_1 P^{-1} \right)$$

Its unitary is then given by

$$e^{-i\mathcal{L}_1 t} \approx \left(I^{\otimes(n-1)} \otimes e^{\frac{-it}{uh^2}X} \;\; P\, e^{\frac{-it}{uh^2}R_1}\, P^{-1} \right)^u$$

Therefore, the circuit is obtained by repeating the following subroutine u times where u is the Trotter step:



5 Conclusion

In this report, we have presented a methodology for constructing quantum circuits to simulate the unitary time evolution governed by symmetric Toeplitz Hamiltonians. The core of our approach lies in a strategic decomposition of the matrix into (M_k) matrices, for which we have developed targeted synthesis techniques. We have shown that for those indexed by powers of two ($k = 2^m$), the evolution can be implemented using a combination of qubit rotations and permutation matrices (quantum adder). Furthermore, we established a significant result for cases where coefficients of matrices within the same congruence class are equal, showing that their combined Hamiltonian simplifies to a sum of tensor products of Pauli operators, allowing a circuit implementation via multi-controlled rotation gates.

The application of this method to the discrete 1D Poisson equation underscored its practical relevance, providing a concrete algorithm for simulating a fundamental physical system. The methods presented in this work provide a systematic approach to addressing a wide range of problems.

Future work will focus on extending this framework to general symmetric Toeplitz matrices and performing detailed benchmarking of the algorithm to analyze Trotter errors and resource overhead.

References

[1] Raymond Hon-Fu Chan and Xiao-Qing Jin. *An Introduction to Iterative Toeplitz Solvers*. Society for Industrial and Applied Mathematics, 2007.

[2] Claude Cohen-Tannoudji, Bernard Diu, and Franck Laloë. *Quantum mechanics*. Wiley, New York, NY, 1977. Trans. of : Mécanique quantique. Paris : Hermann, 1973.

[3] Ivan Gavrilyuk, Volodymyr Makarov, and Vitalii Vasylyk. Exponentially convergent algorithms for abstract differential equations, 2011.

[4] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[5] Maxime Remaud and Vivien Vandaele. Ancilla-free quantum adder with sublinear depth. In *International Conference on Reversible Computation*, pages 137–154. Springer, 2025.

[6] Sunheang Ty, Renaud Vilmart, Axel Tahmasebi Moradi, and Chetra Mang. Double-logarithmic depth block-encodings of simple finite difference method's matrices, 2024.

Appendix : Proofs

Pauli matrices

We will prove that any $2^n \times 2^n$ matrix can be expressed using a linear combination of tensor products of Pauli matrices and the identity:

1. For $2^n \times 2^n$ matrices, let's consider all tensor products:

$$\left\{ \bigotimes_{k=1}^{n} \sigma_{i_k} \mid \sigma_{i_k} \in \{I, \sigma_x, \sigma_y, \sigma_z\} \right\}$$

   There are 4^n such products, matching $\dim(\mathbb{M}_{2^n \times 2^n}(\mathbb{C})) = (2^n)^2 = 4^n$.

2. For distinct products $B = \bigotimes_k \sigma_{i_k}$ and $C = \bigotimes_k \sigma_{j_k}$:

$$\mathrm{Tr}(B^\dagger C) = \prod_{k=1}^{n} \mathrm{Tr}(\sigma_{i_k}^\dagger \sigma_{j_k}) = 0$$

   since $\mathrm{Tr}(\sigma_i^\dagger \sigma_j) = 2\delta_{ij}$. This proves linear independence.

3. Thus, the 4^n tensor products form a basis by dimensional matching (4^n elements) and linear independence. QED.

Trotterization

We will prove that

$$\lim_{n\to\infty}\left(e^{iA\frac{t}{n}}\,e^{iB\frac{t}{n}}\right)^n \;=\; e^{i(A+B)t}$$

Using the power series expansion $e^X=\sum_{k=0}^{\infty}\frac{X^k}{k!}$, we have for large n:

$$e^{iA\frac{t}{n}}=I+iA\frac{t}{n}+O\big(n^{-2}\big),\quad e^{iB\frac{t}{n}}=I+iB\frac{t}{n}+O\big(n^{-2}\big)$$

Hence

$$e^{iA\frac{t}{n}}\,e^{iB\frac{t}{n}}=I+i(A+B)\frac{t}{n}+O\big(n^{-2}\big)$$

Raising to the nth power gives

$$(e^{iA\frac{t}{n}}e^{iB\frac{t}{n}})^n=\big[I+\frac{1}{n}i(A+B)t+O(\frac{1}{n^2})\big]^n=\sum_{k=0}^{n}\frac{n!}{k!(n-k)!}\frac{1}{n^k}(i(A+B)t)^k$$

But

$$\lim_{n\to\infty}\frac{n!}{k!(n-k)!}\frac{1}{n^k}=\lim_{n\to\infty}\frac{1}{k!}\frac{n(n-1)(n-2)\dots(n-k+1)}{n^k}=\lim_{n\to\infty}\frac{1}{k!}(1-\frac{1}{n})(1-\frac{2}{n})\dots(1-\frac{k-1}{n})\approx\frac{1}{k!}(1+O(\frac{1}{n}))$$

Therefore

$$\lim_{n\to\infty}\left(e^{iA\frac{t}{n}}\,e^{iB\frac{t}{n}}\right)^n=\sum_{k=0}^{\infty}\frac{\big(i(A+B)t\big)^k}{k!}=e^{\,i(A+B)t}$$