Optimal Time Discretization in Quantum Simulation: The Trade-off Between Trotter Error and Finite Precision

Manav Ramprasad
Intro to Quantum Algorithms

Abstract

Digital quantum simulation requires approximating the continuous time evolution operator e^{-iHt} via product formulas (Trotterization). While increasing the number of Trotter steps (n) reduces algorithmic error, it simultaneously increases the computational depth, leading to an accumulation of errors due to finite numerical precision. In this project, we simulate a 1D Heisenberg spin chain to characterize the total simulation error as a function of n. We demonstrate a convex "U-shaped" error profile, identifying an optimal operating regime where algorithmic and precision errors are balanced. Furthermore, we compare first-order and second-order product formulas, showing that while higher-order methods reduce Trotter error, they remain bounded by the same precision floor. Our results provide a classical analogy for the trade-off between Trotter depth and fault-tolerant rotation synthesis precision.

1 Introduction

Simulating the dynamics of quantum systems is one of the most promising applications of quantum computers [1]. The core challenge lies in implementing the time-evolution operator $U(t) = e^{-iHt}$, where H is the system Hamiltonian. Since H is often a sum of non-commuting terms (e.g., $H = A + B$), we cannot simply implement $e^{-iAt}e^{-iBt}$. Therefore, when simulating the time evolution of a quantum system, it is usually not possible to directly compute the matrix exponential of the Hamiltonian H. In class, we have learned about using Linear Combination of Unitaries (LCU) as a strategy for approximation of this target unitary. Another approach, where instead of discretizing on the exponential, we discretize on time, is called Trotterization [2]. Trotterization involves splitting H into a sum of parts that are easier to exponentiate, followed by applying a Trotter product formula:

$$e^{-i(A+B)t} = \lim_{n\to\infty} \left(e^{-iAt/n}e^{-iBt/n}\right)^n \tag{1}$$

These equations fall under the Trotter-Suzuki product formulas, which we can then use to approximate the evolution.

However, choosing the number of Trotter steps n presents a fundamental optimization problem.

- **Low n:** The Trotter approximation is coarse, leading to significant algorithmic error (Trotter error).
- **High n:** The circuit depth increases. On classical hardware, this accumulates floating-point round-off error. On quantum hardware, this accumulates gate errors and requires higher precision rotation synthesis.

This project investigates this trade-off by simulating a quantum system under artificial precision constraints. We aim to verify the hypothesis that the total error follows a convex curve and determine how higher-order formulas and varying precision levels shift this curve.

2 Theoretical Background

2.1 The Hamiltonian

We consider a 1D Heisenberg spin chain with $N = 3$ qubits. The Hamiltonian is defined as:

$$H = \sum_{j=1}^{N-1} (X_j X_{j+1} + Z_j Z_{j+1}) \tag{2}$$

where X and Z are the Pauli matrices. We decompose this into two non-commuting parts, $H = H_A + H_B$, where H_A contains the X terms and H_B contains the Z terms.

2.2 Trotter Error

According to the theory developed by Childs et al. [3], the error in a product formula depends on the commutators of the Hamiltonian terms.

- **First Order (Lie-Trotter):** $S_1(t) = e^{-iH_A t} e^{-iH_B t}$. The error scales as $O(t^2/n)$.
- **Second Order (Strang Splitting):** $S_2(t) = e^{-iH_B t/2} e^{-iH_A t} e^{-iH_B t/2}$. The error scales as $O(t^3/n^2)$.

As $n \to \infty$, Trotter error vanishes.

2.3 Precision Error

In any finite-precision computation (classical or quantum), each matrix multiplication introduces an error ϵ. If the simulation requires n steps, these errors accumulate. A worst-case analysis suggests the error scales linearly as $O(n \cdot \epsilon)$.

2.4 Total Error Model

We propose that the total error E_{total} is the sum of these two components:

$$E_{total}(n) \approx \frac{C_{alg}}{n^p} + C_{prec} \cdot n \cdot \epsilon \tag{3}$$

This function is convex, implying the existence of a global minimum n_{opt}.

3 Methodology

To visualize the trade-off within a feasible computation time, we developed a Python simulation that explicitly "injects" precision noise.

- **System:** 3 Qubits, Total time $T = 1.0$.
- **Exact Evolution:** Computed using 'scipy.linalg.expm' with full 64-bit precision.
- **Noisy Simulation:** The Trotter evolution was computed by iteratively multiplying unitary matrices. After each multiplication, the matrix elements were rounded to d decimal places (varying $d \in \{3, 4, 5\}$). While our simulation uses decimal truncation, this is mathematically analogous to the unitary approximation error ϵ in grid synthesis, where a target gate U is approximated by a sequence V such that $||U - V|| \leq \epsilon$.
- **Metric:** We measured error using the Spectral Norm difference: $||U_{exact} - U_{approx}||_2$.

4 Results

4.1 Impact of Trotter Order

We first compared the performance of 1st Order (Lie-Trotter) and 2nd Order (Strang) formulas under fixed precision ($d = 4$).

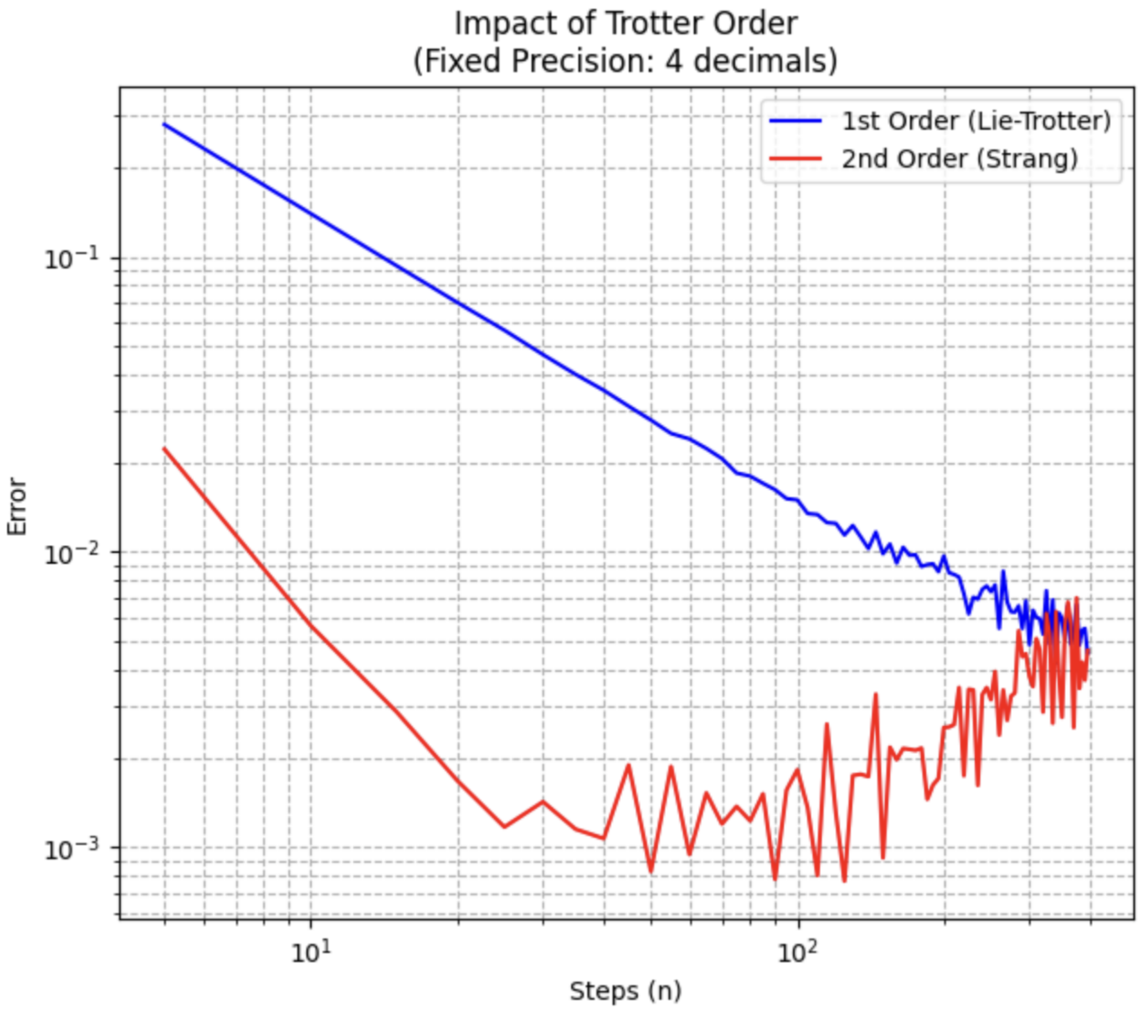

Figure 1: Comparison of First and Second Order Trotter errors (Fixed Precision: 4 decimals). The Second Order formula (Red) reduces algorithmic error more rapidly ($O(1/n^2)$) than the First Order formula ($O(1/n)$) in the left-hand regime, but both eventually reach the same noise floor.

In our Trotter order figure, the red line (2nd order) drops much faster on the left side than the blue line (1st order). This steeper slope indicates a much more rapid drop in error, and the minimum that the red line hits is significantly below the minimum of the blue line. However, both lines are actually bounded by the same precision "floor", as shown by the convergence on the right side of the graph. This behavior happens because the matrix multiplications for each is roughly the same, and proves that better algorithms may save time but do not fix hardware noise.

4.2 Impact of Finite Precision

Next, we analyzed the 1st Order formula while varying the simulated hardware precision.

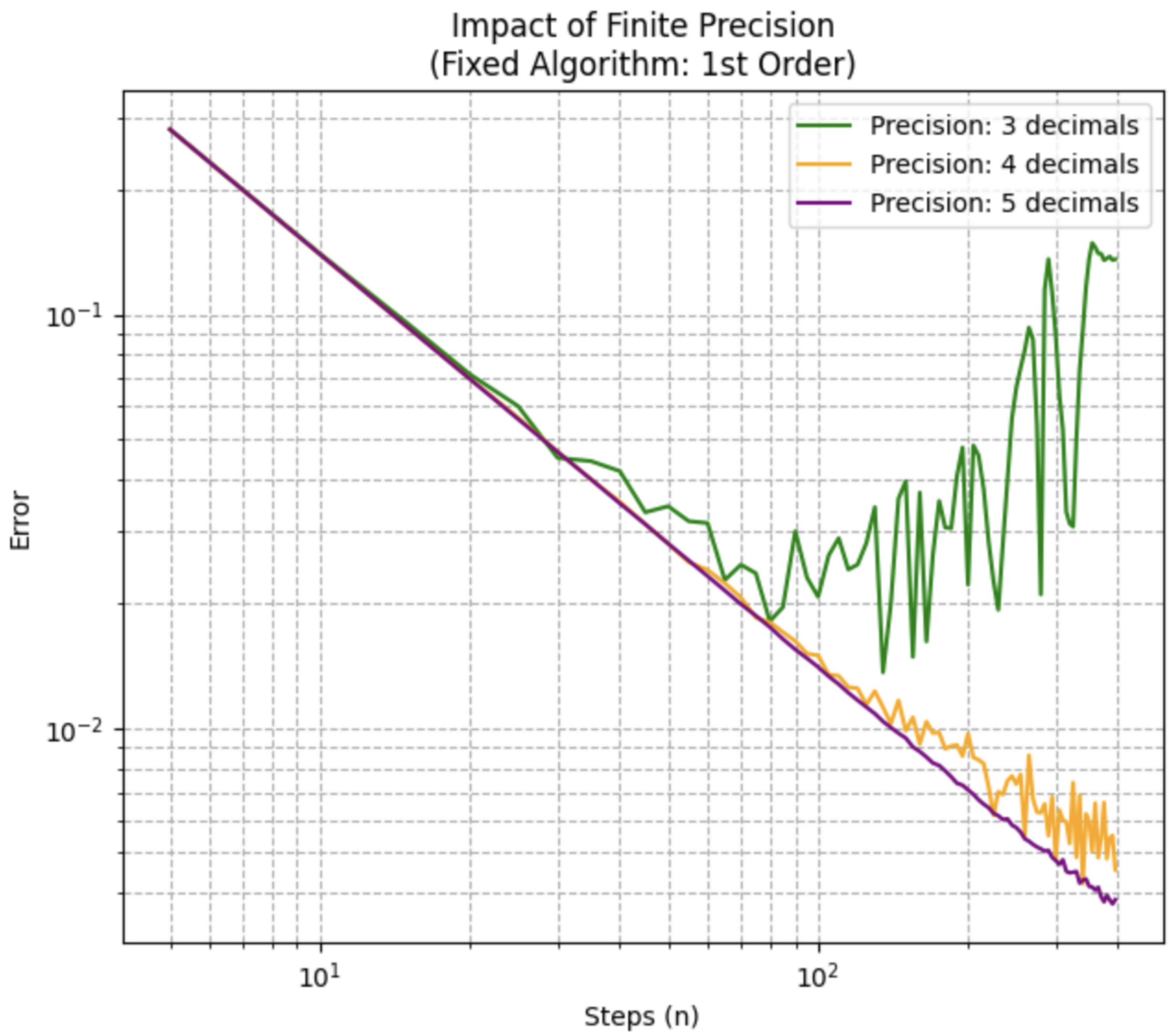

Figure 2: Total simulation error for varying levels of precision (10^{-3}, 10^{-4}, 10^{-5}). As precision improves, the noise floor drops and the "U-curve" minimum shifts to the right, allowing for a larger optimal number of Trotter steps (n_{opt}).

In our finite precision figure, the green line (low precision), reaches is minimum much sooner than the yellow line (medium precision) and the purple line (high precision). As expected, the higher the precision, the lower the minimum error, and the less we see the "U-Curve".

The code used to generate the figures is available via Google Colab at: [Link].

5 Discussion

5.1 The Jagged Nature of Precision Error

A notable feature in the right side of the plots is the jagged, non-monotonic behavior of the error. Unlike Trotter error, which is deterministic and smooth, rounding error is effectively stochastic (technically deterministic in a deterministic classical simulation). At each step, values are rounded up or down, leading to a "random walk" of error accumulation. This confirms that the dominant error source in this regime is indeed numerical noise rather than algorithmic deficiency.

5.2 Analogy to Quantum Rotation Synthesis

While this simulation used classical floating-point rounding, the results map directly to quantum fault tolerance. In quantum computing, arbitrary unitaries are synthesized using a discrete gate set (e.g., Clifford+T). Approximating a rotation to a finer precision ϵ requires a longer gate sequence, meaning a higher T-count [1].

Our results in Fig. 2 show that for a given hardware/synthesis precision, there is a maximum useful circuit depth. Attempting to reduce Trotter error by increasing n beyond this point is counter-productive, as the cost of implementing the gates (or the noise within them) outweighs the algorithmic benefit.

6 Conclusion

In this project, I wanted to demonstrate the convex trade-off between Trotterization error and finite precision error. We showed that second-order product formulas provide a significant advantage in the algorithmic-limited regime but offer no benefit once the precision floor is reached. These findings highlight the importance of balancing algorithm design with hardware capabilities; specifically, the number of Trotter steps must be carefully tuned to match the available precision (or gate fidelity) of the quantum processor.

7 Future Work

While this project establishes the fundamental trade-off between algorithmic and precision errors, there are many related problems that I wasn't able to tackle in this project:

1. Higher-Order Suzuki-Trotter Formulas: We limited our analysis to first-order and second-order symmetric formulas. Future work could investigate higher-order Suzuki product formulas. While these formulas reduce the asymptotic Trotter error to $O(1/n^4)$, they require significantly more matrix exponentials per time step (exponentially increasing circuit depth).

2. Commutator-Based Adaptive Trotterization: Our current simulation uses a fixed step size Δt. However, the theory developed by Childs et al. [3] shows that error scales with the norm of the commutators $[H_A, H_B]$. For systems where the Hamiltonian terms vary in magnitude or local interaction strength, an adaptive Trotter scheme—where the step size changes dynamically based on the instantaneous commutator norm—could potentially smooth out the error curve and allow for lower total gate counts.

3. Open System Dynamics: In this project, our model assumes unitary evolution (closed system) where noise is purely coherent (control/precision error). Real quantum hardware suffers from incoherent noise, such as amplitude damping (T_1) and dephasing (T_2). Somehow including this could be interesting.

References

[1] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information.* Cambridge university press, 2010.

[2] H. F. Trotter, "On the product of semi-groups of operators," *Proceedings of the American Mathematical Society*, vol. 10, no. 4, pp. 545–551, 1959.

[3] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, "A theory of trotter error," *Physical Review X*, vol. 11, no. 1, p. 011 020, 2021.