Trotterization in Quantum Computing

1 Introduction

Trotterization is a key numerical technique used in quantum computing for approximating the time evolution operator of a quantum system. It is based on the Suzuki-Trotter expansion, which enables the decomposition of an exponential of sum of non-commuting operators into a sequence of exponentials of individual terms. This technique is particularly useful for simulating quantum many-body systems and Hamiltonian dynamics on quantum computers.

2 Trotter-Suzuki Decomposition

Consider a quantum system with a Hamiltonian that can be decomposed as a sum of non-commuting terms:

$$H = H_A + H_B. \tag{1}$$

The exact time evolution operator over a small time step t is given by:

$$U(t) = e^{-iHt} = e^{-i(H_A+H_B)t}. \tag{2}$$

However, due to the non-commutativity of H_A and H_B, we cannot simply split the exponentiation. Instead, using the first-order Trotter approximation:

$$e^{-i(H_A+H_B)t} \approx e^{-iH_At}e^{-iH_Bt} + \mathcal{O}(t^2). \tag{3}$$

This approximation becomes more accurate when we divide the time interval into smaller steps n:

$$U(t) = \left(e^{-iH_At/n}e^{-iH_Bt/n}\right)^n + \mathcal{O}(t^2/n). \tag{4}$$

In the limit $n \to \infty$, the error vanishes, and the decomposition becomes exact.

3 Why Non-Commuting Observables' Exponential Cannot Be Split

In quantum mechanics, the evolution of a system is often governed by an operator exponential of the form:

$$e^{(A+B)t}. \tag{5}$$

If the operators A and B commute, i.e.,

$$[A, B] = AB - BA = 0, \tag{6}$$

then we can factor the exponential as:

$$e^{(A+B)t} = e^{At}e^{Bt}. \tag{7}$$

However, when A and B do not commute, this factorization is no longer valid. The reason lies in the Baker-Campbell-Hausdorff (BCH) formula, which provides an expansion for the exponential of a sum of non-commuting operators:

$$e^{(A+B)t} = e^{At}e^{Bt}e^{-\frac{t^2}{2}[A,B]}e^{\mathcal{O}(t^3)}. \tag{8}$$

The presence of the additional exponential terms involving the commutator $[A, B]$ and higher-order nested commutators means that the naive splitting of the exponentials introduces errors. Thus, for non-commuting operators, the direct separation:

$$e^{(A+B)t} \neq e^{At}e^{Bt} \tag{9}$$

is incorrect in general.

This property has important implications in quantum mechanics and quantum computing, particularly in time evolution and numerical approximations, where methods like Trotter-Suzuki decomposition are used to approximate such exponentials while managing the error introduced by non-commutativity.

4 Trotterization and Repeated Evolution Steps

In quantum mechanics, time evolution is given by the unitary operator:

$$U(t) = e^{-itH}, \tag{10}$$

where H is the Hamiltonian of the system. If the Hamiltonian consists of multiple non-commuting terms, such as:

$$H = H_X + H_Y + H_Z, \tag{11}$$

then the direct exponentiation of the sum is not equal to the product of exponentials due to non-commutativity:

$$e^{-it(H_X+H_Y+H_Z)} \neq e^{-itH_X} e^{-itH_Y} e^{-itH_Z}. \tag{12}$$

To approximate this evolution, we use Trotterization.

4.1 Small Time Step Approximation

Instead of evolving the system for a large time t, we break it into small steps:

$$t = m \cdot dt, \quad dt = \frac{t}{m}. \tag{13}$$

For a small time step dt, the first-order Trotter approximation is:

$$U(dt) \approx e^{-idtH_X} e^{-idtH_Y} e^{-idtH_Z}. \tag{14}$$

Since evolution is sequential, to achieve the full-time evolution, we apply this small-step evolution m times:

$$U(t) \approx \left(e^{-idtH_X} e^{-idtH_Y} e^{-idtH_Z}\right)^m. \tag{15}$$

This effectively constructs the evolution over t by repeating small-step transformations.

4.2 Why Repeat Instead of Summation?

One might wonder why we do not use a summation instead of repeating the operation. The reason lies in the nature of matrix exponentiation. A naive summation approach would be:

$$U(t) \approx \sum_{k=1}^{m} e^{-idtH}, \tag{16}$$

which is incorrect because exponentials of matrices do not sum linearly. Instead, the correct approach follows from:

$$U(t) = (U(dt))^m = \left(e^{-idtH}\right)^m. \tag{17}$$

This is analogous to compound interest in finance. If an investment grows at rate r, instead of applying the rate once per year, it compounds in small steps:

$$(1 + rdt)^m. \tag{18}$$

As $dt \to 0$, this approaches the continuous exponential growth formula:

$$e^{rt}. \tag{19}$$

Similarly, breaking the quantum evolution into small steps and repeating them ensures the correct accumulation of time evolution effects.

4.3 Final Summary

- Instead of evolving for time t, we break it into small steps: $t = m \cdot dt$. - We approximate $U(dt)$ using first-order Trotterization. - Since evolution is sequential, we repeat $U(dt)$ m times to get $U(t)$. - This method ensures proper time evolution in quantum simulations and quantum computing.