Trotterization and Suzuki Decomposition

Quantum Time Evolution Simulation

Outline

Quantum Hamiltonian Evolution

The time evolution operator for a quantum system is $U(t) = e^{-iHt}$, solving the Schrödinger equation $i, \frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle$. Simulating $U(t)$ is essential in physics and chemistry . Many Hamiltonians are a sum of terms, $H = \sum_j H_j$. If all terms commute, time evolution factorizes exactly: e.g. for $H = H_1 + H_2$ with $[H_1, H_2] = 0$, we have $e^{-i(H_1+H_2)t} = e^{-iH_1 t}e^{-iH_2 t}$. In general H_j *do not* commute, so $e^{-i(H_1+H_2)t} \neq e^{-iH_1 t}e^{-iH_2 t}$. We need to approximate the evolution by alternating the non-commuting pieces in small time slices.

Trotter Product Formula

$\mathrm{e}^{-i(H_1+H_2)t} = \lim_{N\to\infty} \left(e^{-iH_1 \frac{t}{N}} \; e^{-iH_2 \frac{t}{N}} \right)^N .$
This is the basic Trotter-Suzuki decomposition (first-order splitting) . In the infinite step limit, it becomes exact (also known as the Lie product formula or Trotter formula). For finite N, $(e^{-iH_1 t/N} e^{-iH_2 t/N})^N$ approximates $e^{-i(H_1+H_2)t}$ with some error. Using a finite N steps is called Trotterization, and the approximation error can be bounded by a desired ϵ .

Higher-Order Suzuki Decompositions

By symmetrizing the sequence, we can cancel lower-order errors. For example, a second-order formula uses half-step kicks of H_1:$S_2(\Delta) \ = \ e^{-iH_1\Delta/2} \ e^{-iH_2\Delta} \ e^{-iH_1\Delta/2}$, $whichyieldse^{-i(H_1+H_2)\Delta}$ up to $O(\Delta^3)$ error . This symmetric Trotter-Suzuki formula eliminates the $O(\Delta^2)$ term. In general, there are higher even-order formulas (4th, 6th, $\ldots$) that achieve errors $O(\Delta^{p+1})$ for any desired order p . These higher-order decompositions (derived recursively by Suzuki) require more instances of the exponential operators (and sometimes negative-time coefficients) to cancel lower-order commutator errors.

First-Order Trotter Expansion (Derivation)

Using the Baker–Campbell–Hausdorff (BCH) formula, one finds:
$e^A e^B =$
$\exp\Big(A + B + \frac{1}{2}[A, B] + \frac{1}{12}[A, [A, B]] - \frac{1}{12}[B, [A, B]] + \cdots\Big). For$ A=-
iH$_1\Delta, ; B = -iH_2\Delta$:
$e^{-iH_1\Delta} e^{-iH_2\Delta} = \exp\Big(- i(H_1 + H_2)\Delta \ - \ \frac{1}{2}[H_1, H_2]\,\Delta^2 +$
$O(\Delta^3)\Big) . Thus, a single Trotter step incurs a local error term \text{-}1 \frac{}{2[H_1, H_2]\Delta^2}.$
The leading error scales as $O(\Delta^2)$, so after $N = t/\Delta$ steps the total error is $O(t, \Delta)$ (first order in Δ).

Second-Order Trotter Expansion (Insight)

In the symmetric product $S_2(\Delta) = e^{-iH_1\Delta/2}e^{-iH_2\Delta}e^{-iH_1\Delta/2}$, the first-order commutator terms cancel out. Intuitively, the $[H_1, H_2]$ error from the first half-step is negated by the second half-step. The leading error in S_2 involves double commutators like $[H_1, [H_1, H_2]]$ (and $[H_2, [H_1, H_2]]$) , which enter at order $O(\Delta^3)$. Thus the second-order scheme has local error $O(\Delta^3)$ (global error $O(\Delta^2)$), a significant improvement over first order.

Example: Single-Qubit $H = X + Z$

Consider a single qubit with Hamiltonian $H = \sigma_X + \sigma_Z$ (Pauli X and Z). Here $[X, Z] = 2iY \neq 0$, so X and Z do not commute . We cannot implement $e^{-i(X+Z)t}$ as one gate, but must Trotterize. Trotter strategy: alternate short rotations about the X-axis and Z-axis. For small Δt, $e^{-iX\Delta t}$ and $e^{-iZ\Delta t}$ are simpler rotations. Repeating them approximates the full evolution $e^{-i(X+Z)t}$. In this case, $e^{-iX\theta} = R_x(2\theta)$ and $e^{-iZ\theta} = R_z(2\theta)$, standard single-qubit rotations . Thus each Trotter step can be directly realized as two orthogonal axis rotations on the qubit.

Trotterization in Python (First-Order)

```
import numpy as np
from numpy.linalg import norm
from scipy.linalg import expm


Define Pauli matrices


X = np.array([[0, 1],
[1, 0]])
Z = np.array([[1, 0],
[0,-1]])
H = X + Z

t = 1.0
N = 4
dt = t/N


First-order Trotter approximation
```

Results: Trotter Approximation Error

With $N = 4$ time steps, the first-order Trotter approximation gives $|U_{\text{trot}} - U_{\text{exact}}| \approx 2.5 \times 10^{-1}$. Increasing to $N = 16$ steps reduces the error to $\sim 6 \times 10^{-2}$. Doubling N roughly halves the error, consistent with $O(1/N)$ convergence (global error $\sim O(t/N)$ for first order). A second-order Trotter scheme yields far smaller error for the same N. For example, at $N = 4$ steps, the symmetric formula gives error $\sim 2.4 \times 10^{-2}$ (about $10\times$ smaller than first order). This faster convergence (error $\sim O(1/N^2)$) is evident in practice. In general, each $e^{-iH_j \Delta t}$ corresponds to a quantum gate implementing that term. In this 1-qubit example, $e^{-iX\Delta t}$ and $e^{-iZ\Delta t}$ are rotations about X and Z axes. Thus the Trotterized $e^{-i(X+Z)t}$ can be realized as a sequence of short rotations, which becomes exact in the limit of fine steps .

Error Scaling Comparison

Error norm versus number of Trotter steps N for first-order (Lie–Trotter) and second-order (symmetric) decomposition of $H = X + Z$. On a log–log plot, the first-order errors (yellow, circles) decrease linearly (slope -1), while second-order errors (red, squares) decrease with slope -2, confirming the $1/N$ vs $1/N^2$ scaling.

Scaling of Trotter Steps with Accuracy

The number of Trotter steps required grows as a function of the simulation time t and desired accuracy ϵ: First order: global error $\sim O(t^2/N)$, so to achieve error ϵ one needs $N = O(t^2/\epsilon)$ steps (gate operations) . Second order: global error $\sim O(t^3/N^2)$, so one needs $N = O!\big((t^3/\epsilon)^{1/2}\big) = O(t^{3/2}/\sqrt{\epsilon})$ steps for error ϵ. Higher-order Suzuki formulas further reduce the scaling. In practice, there is a trade-off: higher order means more gates per step. One chooses an order that minimizes total error (Trotter error + hardware errors) for a given quantum hardware .

Exercises

1. Use the BCH expansion to show the leading correction term for $U_{\text{trot}}(\Delta) = e^{-iH_1\Delta}e^{-iH_2\Delta}$ is $-\frac{i}{2}[H_1, H_2]\Delta^2$. (Hint: Expand $e^{-iH_1\Delta}e^{-iH_2\Delta}$ to second order in Δ.)
2. Verify that in the second-order formula $S_2(\Delta) = e^{-iH_1\Delta/2}e^{-iH_2\Delta}e^{-iH_1\Delta/2}$, the $[H_1, H_2]$ term cancels out. What commutator(s) govern the leading error term of S_2?
3. Write a Python script (using NumPy) to simulate $U(t) = e^{-iHt}$ for a simple 2×2 Hamiltonian $H = H_1 + H_2$ with and without Trotterization. Compare the norm error $|U_{\text{trot}} - U|$ for different N and for first vs second-order Trotterization.