Trotter–Suzuki Methods

Distinguish product-formula approximation error from sampling and hardware errors.

Undergraduate | 22 min lesson | Dr. Charith Jayasekara

By the end, you'll be able to

- Apply first- and second-order splitting conventions.
- Compare an X+Z example with exact evolution.
- Explain the cost trade-off from increasing slice count.

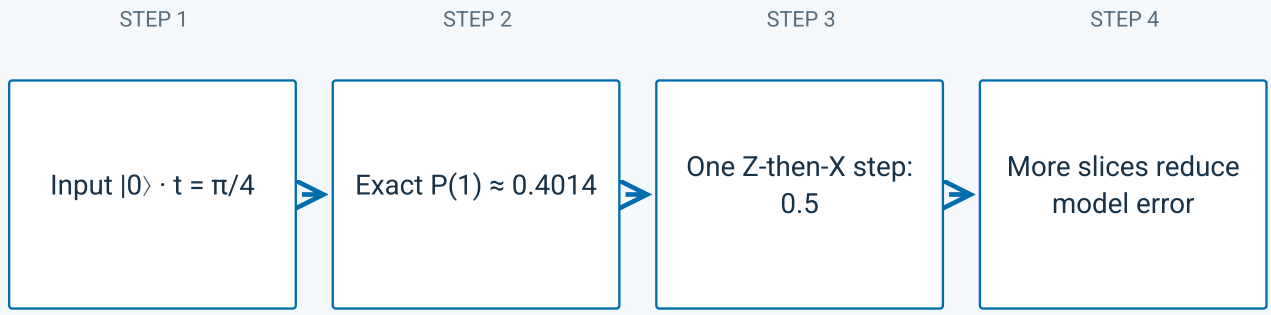
Approximate simultaneous evolution by short pieces

Product-formula simulation replaces evolution under a sum of Hamiltonian terms with a sequence of evolutions under the individual terms. The idea resembles approximating a curved route by many short straight moves. The analogy is useful only if you remember that quantum terms may not commute: changing their order can change the approximation.

Prerequisites: matrix and gate order, phase rotations and Hamiltonian simulation. This lesson uses one-qubit matrices so that an approximation error can be checked independently. Sparse-Hamiltonian simulation work gives a primary algorithmic context for higher-order product formulas. Berry and colleagues.

WORKED DIAGRAM · A23

Split X + Z evolution



The one-step product formula gives P(1) = 0.5 rather than the exact value about 0.4014; more shots do not repair that deterministic difference.

State the formula and its convention

For H equal to A plus B, a first-order approximation over time t uses r repetitions of an A evolution and a B evolution, each for time t divided by r. Written as a matrix product, the rightmost factor acts first. Specify that order when

translating a formula into a left-to-right circuit.

As r grows, the ideal approximation approaches the desired evolution under suitable finite-dimensional conditions. If A and B commute, the composition is exact already at one step. If they do not, commutator terms govern the leading error. The number of slices needed therefore depends on more than the count of Hamiltonian terms.

Write the approximation in the same execution order used by the worked circuit.

$$U_1(t;r) = \left(e^{-iAt/r}e^{-iBt/r}\right)^r \longrightarrow e^{-i(A+B)t} \quad (r \to \infty).$$

r is the positive integer slice count. The rightmost factor applies B first; choosing $A = X, B = Z$ matches the worked example. For commuting A, B, the equality holds at every slice count.

Worked example: X plus Z

Take H equal to X plus Z, start in zero and evolve for time pi over four. Since H squared equals twice the identity, the exact unitary is a cosine times identity minus i times H divided by square root of two times a sine. The exact probability of one is

$$P_{\text{exact}}(1) = \frac{1}{2}\sin^2\left(\frac{\pi}{2\sqrt{2}}\right) \approx 0.4014.$$

Use one first-order step with Z applied first and X second, each for the full requested time. The Z operation adds a common phase to the zero input. The X operation then creates equal-magnitude zero and one components. The resulting amplitudes are one half minus i over two and negative one half minus i over two, giving probability one half for one.

The approximate probability differs by about 0.0986. This discrepancy exists before any shot sampling or hardware noise. Taking more shots would estimate the approximate circuit's half probability more precisely; it would not make that circuit converge to the exact Hamiltonian result. To reduce this error, change the approximation, such as increasing the slice count.

The exact unitary and the one-slice circuit can be compared directly on the same input.

$$e^{-it(X+Z)} = \cos(\sqrt{2}t)I - i\frac{\sin(\sqrt{2}t)}{\sqrt{2}}(X+Z), \qquad e^{-iX\pi/4}e^{-iZ\pi/4}|0\rangle = \frac{1-i}{2}|0\rangle - \frac{1+i}{2}|1\rangle.$$

The exact expression follows from $(X+Z)^2 = 2I$. The approximate column has probability one half on each label at the worked time, which differs from the existing exact probability formula.

Why smaller pieces help

Over a short interval delta, both the exact exponential and the product have the same constant and first-order terms. Their first disagreement involves products of A and B at second order. With roughly t divided by delta intervals, the accumulated first-order-formula error has a scale proportional to t squared divided by r, multiplied by appropriate operator and commutator factors.

That expression is a leading scaling guide, not a universal equality or a sharp bound for every input. Some observables and structured Hamiltonians exhibit smaller error than a worst-case norm bound. Verify a small case numerically and state whether the accuracy claim concerns the full unitary, a prepared state or a particular observable.

Symmetric formulas and higher order

A common second-order step applies half an A interval, a full B interval and another half A interval. Its symmetric arrangement cancels the leading second-order mismatch, giving local error of order the step size cubed. For fixed bounded operators and total time, repeating r slices gives second-order global scaling, proportional to $1/r^2$ under the usual assumptions. Repeating that step gives a better approximation at a given small step size, but also uses more elementary operations per slice.

Higher-order Suzuki constructions combine shorter formulas with carefully chosen coefficients. Some include negative effective time segments. These can be valid when inverse evolutions are implementable, but they are not free on every hardware platform. The best order balances approximation cost, available interactions, compilation and physical error exposure.

Construct and validate a simulation

First identify individually implementable Hamiltonian terms and their norms. Select a time, error target, ordering and formula order. Estimate or bound the slice count. Build a small exact reference using matrix exponentiation or diagonalisation. Compare the approximate unitary or requested observable across increasing slice counts. Only then introduce finite-shot sampling and a hardware-noise model.

Keep the mathematical error budget separate from physical gate errors. More slices reduce product-formula error but increase circuit depth and exposure to imperfect operations. An experiment can therefore have an intermediate best slice count rather than improving indefinitely. Report both trends instead of interpreting a noisy convergence curve as a failure of the product-formula theorem.

Change the input to expose ordering

The one-step X-plus-Z example used zero, which made the first Z evolution contribute only a common phase before X. To test order more fully, use plus or an input with an imaginary component. Noncommuting operations can then produce different final amplitudes even when a selected probability happens to agree.

The commutator of X and Z is nonzero, so reversing the two factors is not a harmless rearrangement. A symmetric second-order step reduces its leading accumulated effect through cancellation, but each half-step must use the correct sign and duration. Doubling a half-step accidentally changes the simulated Hamiltonian rather than merely its approximation accuracy.

A good convergence plot includes an exact reference and multiple slice counts on the same fixed input and time. Keep shots high enough that sampling error does not hide the deterministic trend, or compare exact simulator expectations first. Then add finite-shot and hardware effects as separate studies.

Complexity and honest comparisons

Product formulas are attractive because their gates can match natural local interactions and their implementation is transparent. They are not always asymptotically optimal in precision. LCU and qubitization provide different access models and precision trade-offs, often with more complicated subroutines. A practical comparison must include those construction costs rather than compare only an asymptotic exponent.

The worked one-qubit example is deliberately classically trivial. Its purpose is to separate model error, approximation error, sampling error and hardware error in a calculation that can be inspected completely. That separation is useful in larger chemistry and many-body simulation studies where mistakes are harder to localise.

Exercises and worked answers

Exercise 1. If A and B are both proportional to Z, does first-order splitting have noncommutation error?

> Check the worked answer
>
> No. The terms commute, so their exponentials combine exactly for any slice count. Hardware and synthesis errors may remain, but the mathematical product-formula error is zero.

Exercise 2. The one-step X-plus-Z circuit yields a frequency 0.499 with one million shots. Has the exact target probability been confirmed?

> Check the worked answer
>
> No. It confirms the approximate circuit's near-half distribution. The exact target is about 0.4014. More shots improve statistical precision about the wrong approximation unless the slice count or formula is changed.

Check your understanding

PAUSE & THINK

Can more shots remove the one-step product-formula error?

- [ ] Yes, always
- [ ] No; change the approximation or slice count
- [ ] Only when the seed is zero

Answer: No; change the approximation or slice count. Shots reduce statistical uncertainty about the circuit actually executed. They do not change its approximate unitary.

Sources

- Berry and colleagues