Precision, condition-number and fault-tolerance costs

Translate numerical accuracy into conditioning, logical reliability and complete physical resource budgets.

Research 22 min lesson Dr. Charith Jayasekara

By the end, you'll be able to

- Calculate sensitivity in an ill-conditioned diagonal system.
- Allocate a simple logical failure budget.
- Compare architectures using space, time and assumptions together.

Precision is a resource choice

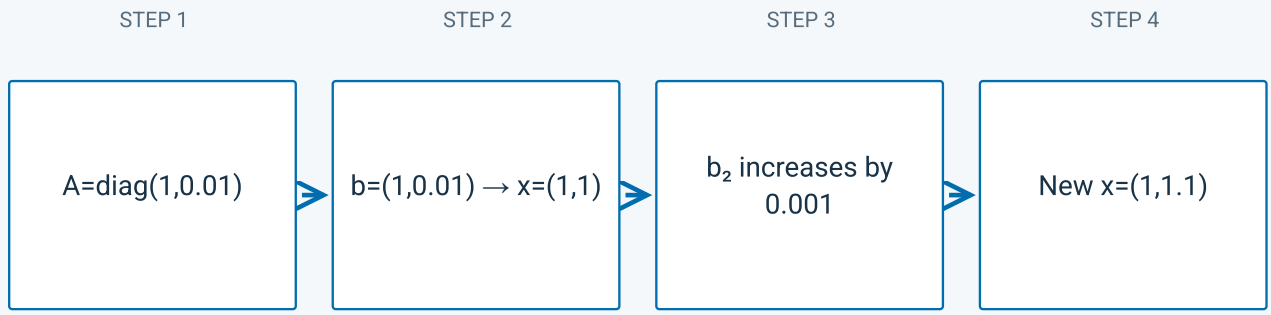
An algorithm is not fully specified by its input size. It also needs an accuracy target, success probability and assumptions about numerical sensitivity. Quantum resource estimates add logical reliability, gate synthesis, scheduling and error-correction overhead. These parameters can dominate even when a formula has a favourable dependence on matrix dimension or candidate count.

The first question is what error the application can tolerate. A normalised-state error, an absolute scalar error, a relative error and a probability of total failure are different quantities. Choosing an unnecessarily strong guarantee can waste resources; choosing a weak one can make the answer useless. Translate the scientific or engineering requirement into the algorithm's actual output metric.

Review normalisation, sampling uncertainty and physical versus logical resources. The useful habit is to keep a budget for each source of error rather than attach one unexplained epsilon to the whole workflow.

WORKED DIAGRAM · AA29

A small input change can matter



The condition number is 100, and the example's second solution component changes by 10%. Separately, 10^8 logical locations with per-location failure $\leq 10^{-10}$ have union-bound failure ≤ 0.01.

Work an ill-conditioned system

Let A be diagonal with entries one and 0.01, and let b be (1,0.01). The solution is (1,1). Now perturb only the second input component to 0.011. The new solution is (1,1.1). A small absolute change of 0.001 in b has produced a ten-percent change in one solution component.

The matrix's condition number is one hundred in the usual spectral norm. That does not mean every perturbation is amplified by exactly one hundred, but it identifies a worst-direction sensitivity scale. A quantum solver cannot remove this mathematical instability. It must resolve the relevant spectral structure and input precision, or solve a regularised problem with a different task description.

If the output is only a normalised state, overall scale is removed, but sensitivity in relative components can remain. If the final observable depends strongly on the amplified direction, input-data uncertainty may dominate. Spending a huge gate budget on a tiny circuit approximation error is then scientifically misplaced.

A small eigenvalue amplifies errors in its direction when the matrix is inverted.

$$A = \operatorname{diag}(1, 0.01), \quad \kappa_2(A) = \frac{1}{0.01} = 100, \qquad \delta b = \begin{pmatrix} 0 \\ 0.001 \end{pmatrix} \implies \delta x = A^{-1}\delta b = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix}.$$

κ_2 is the spectral-norm condition number. The perturbation changes the solution from $(1, 1)$ to $(1, 1.1)$; it does not mean every perturbation is amplified by exactly one hundred.

Separate coherent and statistical accuracy

An approximation to a unitary or block encoding has a coherent error metric. A sampled expectation has statistical uncertainty. Reducing an operator approximation error logarithmically in a synthesis parameter does not automatically make the final classical estimate logarithmic in its desired accuracy. Repeated measurements can still require inverse-error-squared sample scale, or a separate coherent estimation algorithm.

Allocate an overall tolerance across input modelling, numerical approximation, state preparation and measurement. A conservative approach adds bounds where a triangle inequality applies. Statistical confidence should use its own

justified procedure. Do not combine unrelated error bars by an arbitrary formula merely because they share the word “error.”

Work a logical failure budget

Suppose a computation has one hundred million logical operation locations and you allow at most one-percent failure from those locations. A simple union-bound allocation asks for per-location failure probability no greater than one in ten billion. Multiplying the location count by that bound gives 0.01. The union bound does not require independence, although actual error correlations can affect how the per-location guarantees are established.

This is an illustrative budget, not a realistic code-distance prescription. Different operation types can have different risks and costs, and failures may not all cause the final answer to be wrong. A detailed architecture model allocates reliability more carefully. The example explains why a small physical gate error is not by itself enough for an enormous computation.

Translate logical circuits into physical resources

Logical qubits need a code, repeated syndrome extraction, decoding and control. Non-Clifford operations may require specialised resource states or other expensive constructions. Routing and memory can add space and time. The number of physical qubits depends on the code, physical error model, target logical reliability and scheduling, not on a universal multiplier.

Gidney’s 2025 RSA resource study illustrates how arithmetic design, storage choices and resource-state production change a full space–time estimate under explicit hardware assumptions. It is a conditional estimate, not a completed attack demonstration or a hardware delivery date. Read the primary resource study.

Follow a resource-estimation workflow

Start with a precise output and tolerance. Count high-level queries and identify their reversible implementations. Compile them into logical operations with a synthesis error budget. Choose an error-correction architecture and physical assumptions, allocate a logical failure budget and schedule memory and resource-state production. Report total physical qubits, runtime and success target together.

Use sensitivity analysis. Vary physical error rates, cycle times, connectivity, synthesis precision and the number of parallel resource factories. A single headline point can hide a large feasible region or a fragile assumption. Present which parameter dominates and which improvement would actually reduce the bottleneck.

Avoid misleading comparisons

Two estimates for the same task can use different algorithms, codes, timing assumptions or success probabilities. A smaller qubit count may trade for a longer runtime or more demanding connectivity. Compare like-for-like assumptions or explain the differences. Do not rank architectures solely by one number copied from their abstracts.

Likewise, logical qubit count is not interchangeable with physical qubit count. An algorithm may use a modest data register but extensive ancillas or resource factories. A block-encoding theorem with few extra ancillas can still require a large underlying data-access circuit. State the accounting boundary before comparing resource claims.

Tie the budget back to the application

If the original data is uncertain at the percent level, a billionth-level numerical answer may not improve the decision. If a rare-event probability is tiny, a percent-level absolute error can be useless. If an optimisation result must be feasible, a small average cost error may not ensure a valid candidate. Precision should serve the output meaning rather than the appearance of many decimal places.

This perspective is useful to every pathway: practitioners can identify bottlenecks, research readers can interpret asymptotic assumptions, and security readers can distinguish an algorithmic attack from an architecture capable of running it reliably. The same resource discipline prevents both exaggerated optimism and unsupported dismissal.

Practice with answers

Exercise 1. Under the simple union allocation, what per-location bound is needed for ten million locations and total failure budget 0.001?

> Check the worked answer
>
> Divide 0.001 by ten million to obtain one in ten billion, or 10 to the negative ten. This is a sufficient allocation under the per-location guarantees, not a complete physical error model.

Exercise 2. Why does improved dependence on state-preparation precision not automatically imply the same improvement for a measured scalar?

> Check the worked answer
>
> The scalar still needs an estimation procedure with its own variance, confidence and repetition costs. Coherent approximation and statistical estimation are different stages. Count the full measurement or amplitude-estimation workflow.

Check your understanding

PAUSE & THINK

For 10^7 locations and total failure budget 0.001, what sufficient per-location allocation follows?

- [ ] 10^{-7}
- [ ] 10^{-10}
- [ ] 0.001

Answer: 10^{-10}. Divide 0.001 by 10^7 to obtain 10^{-10}. This union allocation is an illustration, not a complete error-correction architecture model.

Sources

- Read the primary resource study