Noise models, backend constraints and execution budgets

Choose explicit noise channels and execution budgets, then separate model checks from hardware claims.

Undergraduate 22 min lesson Dr. Charith Jayasekara

By the end, you'll be able to

- Check a density matrix against an analytic channel.
- Distinguish coherent noise, relaxation and readout.
- Record shots, target and execution limits.

Begin with a clear question

A noise model is a mathematical approximation to particular physical processes. Its usefulness comes from the questions it can answer and the assumptions you can test. Choose a channel tied to a mechanism, validate a small exact case and then investigate a larger circuit using that stated model.

Build the reasoning

Amplitude damping models energy relaxation from one to zero with probability gamma. Applied to an initially excited qubit, it produces a mixture with populations gamma and one minus gamma. This differs from taking fewer shots: the underlying state changes. Readout errors act at the measurement interface, and a coherent over-rotation changes a gate systematically. These mechanisms need separate parameters.



Follow time from left to right. Qiskit statevector labels put q0 on the right; diagrams put q0 on the top wire.

Worked example

For damping probability 0.2, an initial one state becomes a density matrix with diagonal entries 0.2 and 0.8. The code checks that exact result, trace one and nonnegative eigenvalues. Notebook 04 extends this to Bloch-vector length, explicit Aer channel placement and asymmetric readout.

$$\rho = \begin{pmatrix} \gamma & 0 \\ 0 & 1-\gamma \end{pmatrix}, \qquad \gamma = 0.2, \qquad \mathrm{Tr}(\rho) = 1.$$

```
import numpy as np
from qiskit.quantum_info import DensityMatrix, Kraus
from qiskit_aer.noise import amplitude_damping_error

channel = Kraus(amplitude_damping_error(0.2).to_quantumchannel())
rho = DensityMatrix.from_label("1").evolve(channel)
assert np.allclose(rho.data, np.diag([0.2, 0.8]))
assert np.isclose(np.trace(rho.data), 1)
assert np.linalg.eigvalsh(rho.data).min() >= -1e-12
print(rho.data)
```

Practical lab

Before execution, record the backend or simulator, target, noise assumptions, shot count, seed and maximum execution setting. A shot limit or time limit is not a monetary price cap. Notebook 08 runs locally by default and leaves hardware submission disabled; a deliberate hardware run requires your own authorised access and current service usage information. Compare several noise strengths and keep a noiseless reference. More shots estimate the noisy distribution more precisely; they do not repair the physical channel.

Run the corresponding notebook from the software practice collection:

- Download Explicit noise: what the model changes and what it assumes
- Download A tested local workflow and an explicit hardware handoff

Keep a short execution record: predicted result, observed result, environment, inputs and the check that would reject a plausible defect. Explain a failure before changing the expected value.

Chapter summary

- Validate an explicit channel on an analytic case.
- Keep relaxation, gate and readout assumptions distinct.
- Record the target, shots, model and execution limits.

Exercises

1. For gamma=1 and input one, give the final density matrix. What does gamma=0 produce?
2. Change one part of the worked program to create a plausible fault. State which assertion should fail and why.

<details>
<summary>Worked answers</summary>

1. Gamma=1 gives $|0\rangle\langle 0|$, the matrix diag(1,0). Gamma=0 preserves $|1\rangle\langle 1|$, diag(0,1). These boundaries are useful channel tests.
2. A useful answer identifies a specific changed gate, parameter, mapping or model and links it to an independent expected behaviour. Keep the intended specification fixed while changing the implementation; save the failing case as a regression test.

</details>

Check your understanding

PAUSE & THINK

What does a larger shot budget do to amplitude damping in this model?

- [ ] Reduces sampling uncertainty while the damped state remains
- [ ] Makes gamma become zero
- [ ] Proves the model matches every real processor

Answer: Reduces sampling uncertainty while the damped state remains. The channel and the sampling budget describe different stages.

Sources

IBM's bit-ordering guide, Statevector API and Operator API describe the SDK conventions and equivalence tools. Qiskit Aer's noise-model guide documents the channel APIs. The notebook bundle records its pinned environment and local execution evidence.