



Documentation reorganization

Status	Draft / Proposed / Accepted / Implemented / Obsolete
Github Issue	https://github.com/unitaryfund/mitiq/projects/6 https://github.com/unitaryfund/mitiq/issues/908
Author(s)	Ryan (ryan@unitary.fund)

Objective

What are we doing and why? What are the goals and non-goals? Keep this short and to the point.

Reorganize and rewrite Mitiq documentation to:

- Update for new Observable support.
- Make it easier to get started with each error mitigation technique in Mitiq.
- Provide clear motivation for using Mitiq by showing improved performance on example problems.
- Develop a stable infrastructure which provides an easy pattern to write new documentation for new techniques.

Documentation infrastructure (e.g., how to host online docs) is outside the scope of this RFC.

Design

Outline a design to implement your proposal in as much detail as possible. Reference specific Mitiq files / code if applicable. Include (pseudo)code if possible. Fill out the specific subsections below.

Organize documentation in two main sections: Guide and examples.

Guide: Contains core concepts in Mitiq and documentation for each error mitigation technique.

What is Mitiq?

Explain what is Mitiq and the basic design principles. E.g.

- toolkit for applying different error mitigation techniques
- thin layer between frontend and backend
- compatible with many frontends
- backend-independent.



Probably use README, or similar content to README.

Core concepts (in no particular order):

- Executor.
- Observable.
- Frontend.
- Backend.
- QuantumResult [floats, density matrices, bitstrings, etc.]
- Composability of multiple error mitigation techniques.

For each error mitigation technique, five (short) files (.myst, .md, .rst, etc.) on the following:

1. How do I use this technique?
 - a. Absolute basics. Just show how to run ``execute_with_technique``.
2. When should I use this technique?
 - a. A simple example where the technique improves performance.
3. What additional options are available?
 - a. E.g., the noise scaling method in ZNE, precision in PEC, or ansatz in CDR.
4. What happens when I use this technique?
 - a. The “lower-level” documentation. E.g., for ZNE, circuits are folded then executed, then results are extrapolated.
5. Who thought of this? Why does it work? And who’s using it?
 - a. A “theory-based” notebook primarily with text and mathematics rather than code.
 - b. References to original literature.
 - c. References to benchmarks and example uses in literature.

Each file uses Cirq with the option to convert to different frontends using `mitiq.conversions.convert_from_mitiq`. Executor uses Cirq simulator with first line as `mitiq.conversions.convert_to_mitiq`.

Examples

Notebooks using Mitiq to reproduce literature results, use a particular front-end, or benchmark a computer with error mitigation.

Organized and searchable by:

- Frontend.
- Backend.
- Technique.



Place to (happily) accept community tutorials / examples.

Guiding principles for documentation:

- The frontend and backend don't matter (in the guide).
- All documentation should ideally be written towards a particular quantum computer, but always default to a noisy simulator so it can be run and tested in CI.
- Keep guides extremely simple and succinct. Fewer words are better.
- Show improved performance to motivate the use of Mitiq.

Notes

Documentation infrastructure is considered outside the scope of this RFC.

Questions and Discussion Topics

Include any open questions you require feedback on for this RFC.

Question 1 about "how do I use this technique"? [AM]

Do we use a generic "draft code" similar to what we did in the paper e.g.:

```
zne_limit = execute_with_zne(  
    circuit: <the input circuit to mitigate>,  
    executor: <the user's defined executor"  
)
```

Or runnable code?

If we use runnable code, what front-end do we use? Always Cirq?

Question 2 about "how do I use this technique"? [AM]

If we want runnable quick-start notebooks instead of picking a single frontend, we could have many notebooks in a subfolder of the examples and link to them from the guide. In this way the user could search (maybe using the "frontend" filter that we plan to add to the examples) for a minimal running notebook that directly works with their frontend.

Question: examples in the guide? [AM]



I have two doubts about section 2 "Why should I use this technique?" that is supposed to be written for each technique :

1) It is a bit strange that we have a few examples in the middle of the guide but later we also have an Examples section. Maybe we could just link one or more examples that we have based on the specific technique.

2) The answer that I would give to "Why should I use this technique?" is similar to all techniques: "to reduce errors in expectation values".

So maybe what the user needs to know is something like: "what technique should I use? What are the main advantages and disadvantages for each method?" This could be unique section or just a simple table placed somewhere in the guide:

```

-----
|technique | advantages| disadv.|
ZNE      | ....    | ..... |
PEC      |         |       |
-----

```