Translational Research Program

# QECSIM – PYTHON QUANTUM ERROR CORRECTION SIMULATOR

## Key Benefits:

- Evaluates key metrics of quantum error correction codes and decoders
- Numerous codes, error models and decoders as used in published results
- Modular design for ease of extension including CLI integration
- Optimized and fully documented

## Background

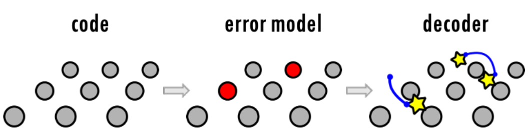
Quantum computers have the potential to solve certain problems that are intractable using classical computers, with the promise of significant societal benefits in areas such as medicine, energy-efficient processes and machine learning.

Quantum error correction (QEC) is expected to play a fundamental role in protecting fragile quantum states and enabling quantum computers to realise their full potential.

Numerical simulation is an essential tool in researching QEC codes and decoders.

## Technology

QECSIM enables researchers to evaluate key metrics of QEC codes and decoders in the presence of experimentally realistic error models.



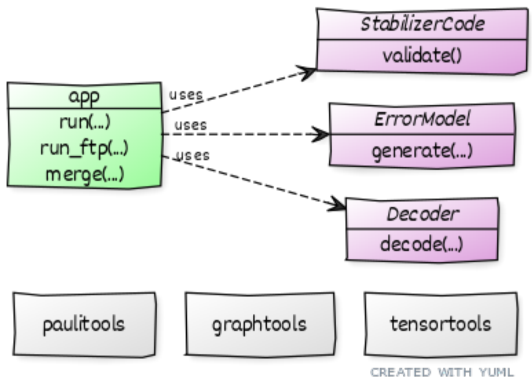

A fully documented CLI and API facilitates running simulations on HPC clusters and prototyping in notebooks.

```
# run [OPTIONS] CODE ERROR_MODEL DECODER ERROR_PROBABILITY...
qecsim run -r50 "toric(3,3)" "generic.bit_flip" "toric.mwpm" \
   0.05 0.1 0.15
```

```
>>> from qecsim import app
>>> app.run(code, error_model, decoder, 0.05, max_runs=50)
```

```
{"code": "Toric 3x3", "error_model": "Bit-flip",
 "decoder": "Toric MWPM", "error_probability": 0.05,
 ..., "logical_failure_rate": 0.03, ...}
```

The design is modular and extensible. Codes, error models and decoders are defined using a rich Python object model. Services use C++ and optimized libraries.




Many leading codes and decoders are included, as well as a detailed extension example with CLI integration.

## Translation project

QECSIM was initially developed as a research tool by a professional software engineer turned PhD student in QEC.

The benefit of using QECSIM as a tool for QEC research has already been established through several high-profile publications:

- DOI:10.1103/PhysRevLett.120.050505
- DOI:10.1103/PhysRevX.9.041031
- DOI:10.1103/PhysRevLett.124.130501

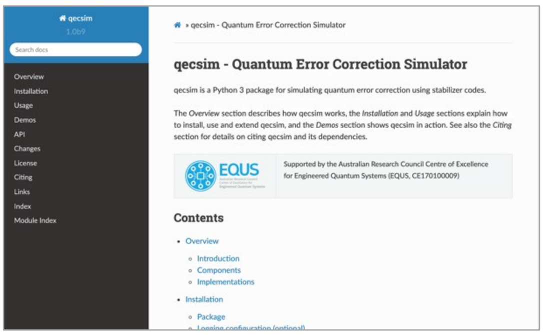
The translation project makes QECSIM freely available to academic and industrial research communities, and packages it for ease of use and extension. The aim is to contribute to the realisation of large-scale reliable quantum computation.

## Outcomes

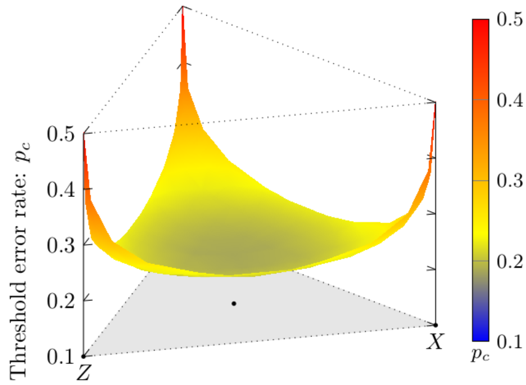
As a result of the translation project, QECSIM has been publicly released under an open-source licence.

The following key objectives were achieved:

- Python Package Index deployment and standard installation:
  **pip install qecsim**
- Open-source code release:
  https://github.com/qecsim/qecsim
- Refactoring to improve separation, flexibility and extension
- Extension example including CLI integration without modification of the core package:
  https://github.com/qecsim/qecsimext
- Full suite of unit tests
- Full documentation:
  https://qecsim.github.io/




In parallel to the translation project, QECSIM has continued to be used for research.




*The first evidence that a topological code can match the performance of random codes for all single-qubit Pauli noise channels [arXiv:2009.07851].*

Researchers both within EQUS and beyond are using QECSIM for new projects.

## Future opportunities

The following future opportunities are anticipated:

- Further research output
- Tutorials and webinars
- New simulation workflows
- Education and outreach material
- Best practice model for related software developments

Forking the codebase and community contributions are also encouraged.

## Team

QECSIM is developed by David K. Tuckett in the Quantum Computation and Information Theory group led by Prof. Stephen Bartlett at the University of Sydney. David is a postdoctoral researcher in quantum physics with more than 13 years of experience as a professional software engineer.