AFRL-AFOSR-VA-TR-2024-0081

Expanding functionality and enhancing documentation of scQubits, an open-source Python package for superconducting qubits

Jens Koch
NORTHWESTERN UNIVERSITY
633 CLARK STREET
EVANSTON, IL, 60208
USA

01/03/2024
Final Technical Report

DISTRIBUTION A: Distribution approved for public release.

Air Force Research Laboratory
Air Force Office of Scientific Research
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	**2. REPORT TYPE**	**3. DATES COVERED**	
20240103	Final	**START DATE** 20200815	**END DATE** 20230814

4. TITLE AND SUBTITLE
Expanding functionality and enhancing documentation of scQubits, an open-source Python package for superconducting qubits

5a. CONTRACT NUMBER	**5b. GRANT NUMBER**	**5c. PROGRAM ELEMENT NUMBER**
	FA9550-20-1-0271	61102F
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Jens Koch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
NORTHWESTERN UNIVERSITY 633 CLARK STREET EVANSTON, IL 60208 USA	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**	**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**
Air Force Office of Scientific Research 875 N. Randolph St. Room 3112 Arlington, VA 22203	AFRL/AFOSR RTB1	AFRL-AFOSR-VA-TR-2024-0081

12. DISTRIBUTION/AVAILABILITY STATEMENT
A Distribution Unlimited: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Research Objectives:
The objective of this project was to expand the functionality of the scqubits package, to make existing components more accessible to researchers entering the field, and to transform scqubits into a widely used tool for research on superconducting qubits.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	5
U	U	U		

19a. NAME OF RESPONSIBLE PERSON	**19b. PHONE NUMBER** *(Include area code)*
GRACE METCALFE	(703) 696-9740

Final Report

Expanding scQubits: an open-source Python package for superconducting qubits

AFOSR Grant No: FA9550-20-1-0271
Principal Investigator: Jens Koch
Institution: Northwestern University

AFOSR Program Manager(s): Dr. Grace Metcalfe

Date: December 15, 2023

SECTION 2: TECHNICAL REPORT

ACCOMPLISHMENTS

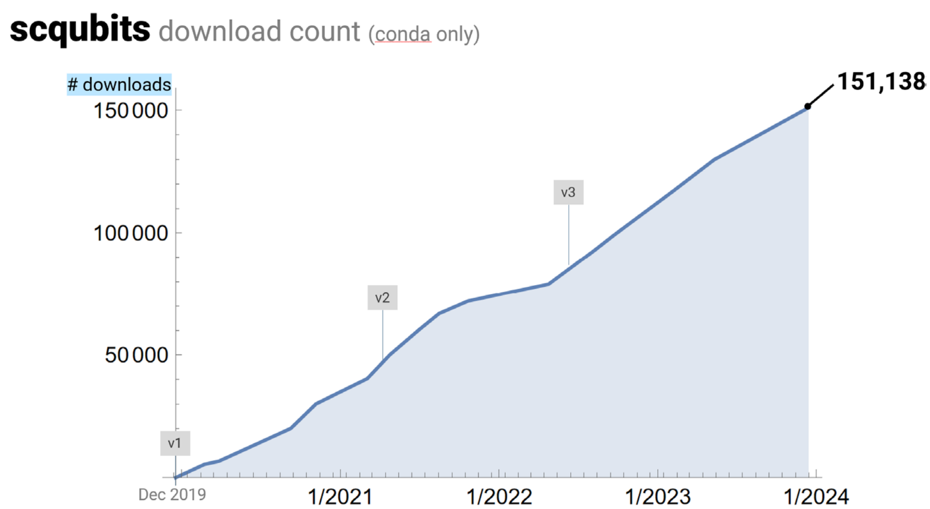
Research Objectives:
The objective of this project was to expand the functionality of the `scqubits` package, to make existing components more accessible to researchers entering the field, and to transform `scqubits` into a widely used tool for research on superconducting qubits.

Detailed goals:
1. Add comprehensive routines for predicting coherence times of the implemented qubit types, based on known relevant noise channels
2. Provide an interface for data extraction and fitting of experimental spectroscopy data
3. Implement new types of superconducting qubit of interest

These goals were pursued and accomplished via platform agnostic code development in Python for numerical solution of computational problems in the context of superconducting qubits and circuit QED. Accessibility and educational value of the package were advanced by creating tutorials, documentation, and graphical user interface components.

Summary of Accomplishments:

- From 03/2020 to 12/2023: published 18 new releases incorporating 1905 commits
- the `scqubits` download count on conda tripled from ~5,000 (March 2020) to ~151,000 (December 2023)



- added module for custom circuit quantization and spectral analysis
- added support for broader set of eigensolvers, including GPU support
- created YouTube video tutorials for scqubits
- advanced "qfit" app to beta release stage (currently undergoing extensive user testing)
- added and extended module for coherence-time predictions
- added graphical user interfaces for qubit initialization, generation of composite Hilbert spaces, and for exploration of single-qubit spectral sweeps
- added new qubit classes:
  `Cos2PhiQubit` [npj Quantum Inf. 6, 8 (2020)], `KerrOscillator`
- introduced support for performing multi-dimensional parameter sweeps

Dissemination of Results:

Publications:
1. Sai Pavan Chitta, Tianpu Zhao, Ziwen Huang, Ian Mondragon-Shem, Jens Koch
Computer-aided quantization and numerical analysis of superconducting circuits
New J. Phys. 24 103020 (2022)

2. Peter Groszkowski, Jens Koch, *Scqubits: a Python package for superconducting qubits*, Quantum 5, 583 (2021)

Presentations:
1. Jens Koch, *Computer-aided quantization and modeling of superconducting qubits, Bolder Boulder Quantum Workshop*, Invited Talk (Boulder, June 2023)
2. Jens Koch, *Introduction to Superconducting Qubits – From Fundamentals to Practical Modeling*, Invited, Lectures at the Spring School on Superconducting Qubit Technology, Centro di Ciencias de Benasque Pedro Pascual, Spain (April 2023)
3. Jens Koch, *Computer-aided quantization and modeling of superconducting qubits*, MIT, iQuISE Seminar (March 2023)
4. Jens Koch, *Computer-aided quantization and numerical modeling of superconducting circuits with "scqubits"*, IEEE QCE 2022, invited workshop talk and panel discussion (September 2022)

5. Jens Koch, *Computer-aided quantization and numerical modeling of superconducting circuits with "scqubits"*, Quantum Africa 2022, invited conference talk (via Zoom, September 2022)
6. Jens Koch, *Computer-aided quantization and numerical modeling of superconducting circuits with "scqubits"*, Rutgers University, seminar talk (Rutgers University, August 2022)
7. Jens Koch, *Computer-aided quantization and numerical modeling of superconducting circuits with "scqubits"*, IMSI Mathematical Methods for Quantum Hardware Workshop, invited talk (UChicago, July 2022)
8. Jens Koch, *scQubits – a Python package for superconducting qubits*, Invited Talk and Panel, IEEE Quantum Week 2021 (via Zoom, October 2021)
9. Jens Koch, *Intrinsic error protection in superconducting qubits*
Invited talk at the Bhaba Atomic Research Center Web Conference (virtual, May 2021)
10. Jens Koch, *Modeling spectra and coherence properties of superconducting qubits with scQubits*, Invited Talk, APS March Meeting 2021 (virtual, March 2021)

YouTube videos

Jun Sung (undergraduate researcher funded by this grant), *Scqubits Tutorial 1: GUI* (5:58)
Jun Sung (undergraduate researcher funded by this grant), *Scqubits Tutorial 2: GUI advanced features* (6:10)

IMPACT

The increasing download count (see above), invitations to give talks and teach lectures at summer schools specifically on scqubits, as well as numerous discussions with colleagues, postdocs and students indicate that the scqubits package has been widely adopted for research purposes, for training of entering graduate students, and for education in the classroom.

AFOSR funding for scqubits successfully boosted the involvement of undergraduate students in research and scientific coding. The created opportunities for undergraduate students to receive compensation for their research was particularly appreciated during COVID times, when many other lab-based opportunities of this kind were temporarily lost. Most of the Northwestern undergraduate students who benefitted from the AFOSR funding provided for this project decided for a continued career in Master's and PhD programs in physics in the US.

CHANGES

We did not encounter fundamental obstacles that would have required adjustments to the scope of plans originally proposed. Spending proceeded at a slower rate than anticipated due to bottlenecks in undergraduate researcher recruitment over COVID. We thank the program manager for authorizing the 1-year No-Cost Extension that has now concluded.

TECHNICAL UPDATES

1. **Beta release of qfit**

README   BSD-3-Clause license

QFit: Interactive Parameter Fitting for Superconducting Circuits

Notice: This package is currently in beta testing. Bugs and issues are expected. We greatly appreciate your feedback and bug reports to help us improve.

Tianpu Zhao, Danyang Chen, Jens Koch

Overview

QFit is your go-to Python application for extracting parameters of superconducting circuits from measured spectroscopy data. Following the four-step workflow, you can get your circuit parameters in no time:

1. **Calibration**: QFit helps to establish the mapping from voltage (your experimental tunable input) to circuit parameters (your simulation ingredients).
2. **Point Extraction**: With just a click, you can locate the peak of the spectrum sweep data with computer-assistance. The extracted data can be simply grouped as a transition and labeled. QFit even provides filters and coloring options for enhancing data visualization.
3. **Interactive Pre-fit**: See your numerical model result and the data on the same plot for intuitive comparison. Adjust the numerical simulator with simple sliders to improve your fit.
4. **Automated Fitting**: With one click, let the numerical optimizers do the work. You can easily configure your fitting: adjust which paramters are fixed or free, set their range, and more.

QFit supports a wide variety of circuit quantum electrodynamic systems, thanks to the powerful Python library `scqubits` as its backend simulator. Once you've extracted your parameters, you can pass them directly to scQubits for any further numerical simulations you need to do.

2. Scqubits updates

- The custom diagonalization module now also supports the jax library.
- Custom diagonalization is now possible for the FullZeroPi qubit.
- New diagonalization module: customize your eigensolver by determining which library, what solvers, and what solver options to use to calculate the eigenvalues and eigenvectors of the various problems scqubits addresses (single qubit or composite system; support for custom Circuits and FullZeroPi to be added). Both sparse and dense diagonalization procedures from libraries such as scipy, primme, and cupy (which offers GPU support), are exposed, and can be easily selected with different sets of predefined options and parameters. In principle, completely arbitrary diagonalization functions can be set by the user.

- Added support for 1/f flux noise dephasing time calculation to the FluxQubit class.
- Overhaul of the graphical user interfaces, including scqubits.GUI, scqubits.Explorer, and scqubits.HilbertSpace.create
- Enhanced support for interfacing with QuTiP/mesolve: for simulation of time evolution it is often preferable to work with matrices in the dressed eigenenergy basis (in the absence of a time-dependent drive).
  To simplify this, all qubits (i.e., QuantumSystem children) now offer an energy_esys keyword argument, and introduce HilbertSpace.op_in_dressed_eigenbasis.
- Added fit method Transmon.find_EJ_EC that extracts EJ, EC of a transmon, based on given E01 and anharmonicity.
- Add E01, anharmonicity as attributes to all qubits inheriting from QubitBaseClass.
- GUI now includes functionality to plot coherence time estimates for various qubits
- Speedup for diagonalization of Transmon and TunableTransmon by recognizing the Hamiltonian matrix as tridiagonal.
- Circuit now implements multiprocessing in routines like get_evals_vs_paramvals when specifying num_cpus=2 or higher as optional keyword argument.
- The class Circuit is now “frozen” to prevent accidental creation of new instance attributes. Doing <circuit instance>.non_existing_attribute = 3 will now raise an error message instead of creating a new attribute.
- New threshold setting scqubits.settings.SYM_INVERSION_MAX_NODES (default: =3) decides whether the capacitance matrix is inverted symbolically (number of nodes ≤ threshold) or numerically (number of nodes > threshold). This avoids apparent hang-ups due to generation of massive symbolic expressions for matrix inverses.
- Added circuit and symbolic_circuit modules, introducing the Circuit class for symbolic and numerical analysis of custom circuits
- Added official support for Python 3.9 and 3.10
- Improved GUI for single qubits (incl., e.g., a Dropdown menu with parameter choices from papers)
- Improved and extended Explorer class
- Additional options for specifying initial and final states in transitions and plot_transitions inside ParameterSweep
- additional helper functions in ParameterSweep: get_subsys(index), subsys_by_id_str(id_str), subsys_evals_count(index), dressed_evals_count
- ParameterSweep offers a new option ignore_low_overlap
- improved status information output when using parallel processing of ParameterSweep data