QuanGuru: A Python Package for Numerical Modelling of Quantum Systems

C. Kargi[a], F. Henriques[a], A. Manatuly[a], A. Di Lonardo[a], G. Gemisis[a], J.P. Dehollain[a], and N.K. Langford[a]

[a] *QSI & MaPS, Faculty of Science, University of Technology, NSW 2007, Australia.*

Numerical code for modelling of quantum systems usually involves many common tasks that can be automated/simplified by introducing abstraction in the underlying code library. In this talk, we introduce a new Python library, named QuanGuru, that implements powerful abstractions providing a broad range of useful and highly versatile functionalities, and show QuanGuru examples. The abstractions implemented by QuanGuru provide a uniform API that decouples the modelling of quantum systems in different numerical contexts into a simple five-step recipe.

The abstractions implemented in QuanGuru are designed to be agnostic to the specific context, allowing researchers to write a single script for different numerical tasks relating to the same model/experiment/ protocol/algorithm/etc. Such abstractions also make writing and generalising these scripts extremely easy and clean, thanks to their decoupling from concrete contexts. Our approach also simplifies the design and implementation of context-specific tools by moving the complicated (but most commonly needed) functions to a higher layer. For the example of numerical modelling, our QuantumToolbox contains Python functions (with no other objects) that create and/or process matrices. The functions in QuantumToolbox implement the most fundamental mathematical definitions of numerical tools and techniques, making it very easy to implement, learn, and extend, with only a few basic guiding principles.

While QuanGuru by default uses its own QuantumToolbox for matrix operations (or any other non-trivial computation), but its context-agnostic design makes it straightforward to use/interface other alternative libraries instead, such as QuTiP and scQubits, cuQuantum, to leverage their capabilities. The flexible and generic descriptions of systems and algorithms created using QuanGuru make it easy for a user to rapidly explore numerical results across a range of varied system and algorithm types without replicating complex application-specific-code.

Figure 1. *There are many libraries and/or software for numerical simulations of quantum systems (e.g. QuTiP, scQubits), design of experiments /devices (e.g. Qiskit-metal, PyQIP), electromagnetic simulations (e.g. CST, HFSS), and experimental control (QCodes, QubiC). The vision of QuanGuru is to create a layer of abstraction for the common tasks in these different contexts to provide a uniform API. QuanGuru also provides a concrete tool, QuantumToolbox, for numerical simulations of quantum systems.*