Paper Presentations

Functional Circuits:

- Quantum Theory in Practice, 2025, Poster, First Author (submission)

- Quantum Physics and Logic, 2025, Poster, First Author (submission)

Reduction to a Subsystem Always Internalises:

- Quantum TUT Workshop, 2023, Workshop, First Author

- Quantum Physics and Logic, 2023, Poster, First Author

Entanglement-efficient bipartite-distributed quantum computing:

- Japan-French Quantum Information Workshop, 2023, Workshop, Co-author

- Quantum Physics and Logic, 2023, Poster, Co-author

- Japan-Singapore Joint Seminar, 2023, Poster, Co-author

- International Symposium on Trans-Scale Quantum Science, 2022, Poster, Co-author

Entanglement-assisted LOCC implementation of bipartite non-local unitary operations:

- Quantum Information Processing, 2022, Poster, First Author

Research Theme Overview

Keywords: Distributed quantum computing, quantum programming

My research is split into two topics: Distributed quantum computing, and quantum programming

=== Distributed quantum computing ===

- Research Background:

Quantum computers have the theoretical ability to efficiently perform computations that classical (i.e. everyday) computers cannot. However due to the technical challenges in building large quantum computers, they currently are not powerful enough to perform computations that are of great use.

- Research Objectives:

Investigate how multiple quantum computers might be combined together to perform calculations that individually they could not, in analogy to classical distributed computing.

- Research Content:

We take advantage of a quantum phenomenon known as 'entanglement' to split a program between multiple computers connected in an arbitrary way. We aim to do this splitting in an efficient way, minimising the amount of entanglement needed to perform the computation. We use graph methods in order to find an efficient implementation, and all of the requisite calculations can be performed automatically by a Python library that we produced (https://github.com/CQCL/pytket-dqc).

- Implementation language:

Python

========================

=== Quantum programming ===

- Research background:

In classical programming, complicated functions are built up from more simple sub-functions.

This is not so easy for quantum computers because they are extremely restricted in the class of functions that can be implemented. For example, an unknown variable (often represented as a complex vector) cannot be duplicated. This makes it difficult to design more complicated functions from simpler sub-functions. However, these restrictions can be avoided if there is some information provided about what the variable is—for example what basis the vector belongs to.

- Research objectives:

Design a framework that allows quantum functions to be combined together, and provide an intuitive way for doing so through understanding how the requisite information needed for each sub-function combines together.

- Research content:

We take an approach from a branch of maths called category theory. By making a 'category' we are able to then precisely describe combinations of different quantum sub-functions, as well as the information needed to implement them.

- Implementation language:

At present, this framework is entirely theoretical, but it is hoped to implement some version of it in Haskell.

========================

Work Experience

iOS App Developer:

(1) UK, Durham University, Department of Physics

(2) Temporary

(3) 2019 Jul - Aug (8 weeks)

(4) Full-time

(5)

Task: Produce an iOS application to assist new Physics students with everyday life.

Role: I was the sole developer working on this project.

Development language and technology used: Swift, Git, Xcode

Repository available at: https://github.com/tim-forrer/DurPhys

Other Development Experience

I completely rewrote our research group website (https://www.eve.phys.s.u-tokyo.ac.jp/), which was originally written in 2004 and had not been redesigned since.

The issues I identified with it were:

1) Everything was written purely in HTML, which made updating the website more cumbersome than necessary, particularly since it had to be updated in both English and Japanese separately.

2) The design was outdated (for example, the view width was a fixed size for all devices) and inconsistent across pages. This made it unpleasant and unintuitive to use.

3) There was no version control in case of changes that broke the website.

I rectified each issue in my rewrite by:

1) Using PHP so that web pages could be built dynamically by reading .json files. This allows members to easily update the page contents by adding entries to the .json files, which would then update both the English and Japanese versions of the site together. It also massively cut down on the amount of duplicate code (for example, in the headers).

2) Introducing CSS stylesheets to both update the website and create a consistent look between pages.

3) Setting up a central Git repository hosted on GitHub that contained all the code and allowed for simple reversion of site-breaking changes.