Introduction to Quantum Computing: Exam Revision Notes

By Petros Wallden, Raul Garcia-Patron and Joschka Roffe

Exam Format

- Date & Time: Wednesday 09 December 2024. Time 13:00-15:00
- Two questions with equal weight.
- Each question divided into two parts with each one multiple sub questions.
- *NOTES AND CALCULATORS PERMITTED examination: candidates may consult up to THREE A4 pages (6 sides) of notes. CALCULATORS MAY BE USED IN THIS EXAMINATION*
- *Note: In the Exam we are evaluating the non-programming parts of the course (i.e. material covered by Raul, Petros and Joschka)*

General Advice

- **Read all questions before choosing and be "strategic"**
- **Questions** are divided into different parts:
  - Each part of a question is divided into small subquestions (weight is clearly stated).
  - In general subquestions hardness increase gradually, but not always.
  - Balance distribution of easy, medium and hard subquestions.
- **Do good time management!**
- **Exam with notes (6 sides) and calculator**
- **Past papers**: Only 2020-2023 IQC exams were open book(-ish).
  2015-2019 had more theory questions and some material that is no-longer covered in IQC.
- **Study Tutorials & train with past years exams** (check assignments)!
- We encourage you to use Piazza as a mutual help support tool, but we will not monitor Piazza ourselves next weeks.

The Basics of QC and QM (Raul' part)

Disclaimer: These are general indicative guidelines (all material unless explicitly mentioned is examinable)

- Lecture 2,3,4,5+10 (Postulates): Material essential for next chapters. Need to know (for example): basic linear algebra and complex numbers, quantum states, operations and measurements (projective measurement, global and partial: see also lecture 10), tensor products, inner-products.
- Lecture 6 (Circuit model): crucial to recognize most important gates and be able to compute the quantum state along the different layers of a quantum circuit and a final measurement outcome probability.

Quantum Algorithms I (Raul’ part)

- Lecture 7 (Phase Kickback and Deutsch-Jozsa): Deutsch-Jozsa, parallelization, phase kickback, Walsch-Hadamard transform, outcome probability resulting from quantum interference. Quantum/classical oracle of a Boolean function and phase kickback.
- Lecture 9 (Bernstein-Vazirani): Classical reversible circuit and its quantum analogue. Existence of quantum oracle from classical one. Building a phase kickback unitary from the quantum oracle. Bernstein-Vazirani algorithm.
- Lecture 12 (Grover): representation on 2dim space, Grover iterations, phase kickback reflection and reflection over initial state. Estimation of number of iteration needed to reach solution.
- Lecture 13 (Simon’s Algorithm): Projective measurement. Measurement of a subsystem. Understand the classical post-processing. Understand why the outcome of lower register is not necessary for solving the problem (just a trick to simplify the proof).

Quantum Algorithms II (Raul' part)

- Lecture 16 (Hadamard Test): Parity check of two qubits (crucial concept in quantum error-correction). Two-qubit SWAP Test. Multi-qubit SWAP Test. Removing the control qubit and use of Bell measurement when we just need the measurement outcome and not the updated state after measurement. General Hadamard Test. Use of projectors to separate subspaces and define measurements. Non-demolition measurement of a qubit.
- Lecture 18 (Quantum Fourier Transform): Binary representation of integers and binary fractions. Fourier transform over Z_{2N}. Iterative construction of QFT circuit. Counting of gates.
- Lecture 19 (Quantum Phase Estimation): Action of an exponentiation of U on an eigenvalue of U. Parallelization of the measurement register and control of different exponentiations of U. Connect the output with QFT to understand that the outcome of measurement is the eigenvalue of U. Action on a general input state. Use of phase-estimation for order finding (reduction of factoring to order finding not examinable).

Near-term QC (Petros' part)

- Lecture 21 (VQA I)

(i) Understand what VQA/VQE is. Motivation, mathematical prob., relevance and the four steps.

(ii) Understand Max-Cut to Ising Spin mapping. Mapping solution (cut) to spin configurations and think how this generalises.

- Lecture 22 (VQA II)

(i) Energy estimation. Including decomposing Hamiltonian to Pauli.

(ii) Ansatz states. Understand the concept of parametrised circuits/states. Be able to produce output state in simple examples. Compute expectation values of observables on output states.

(iii) Classical optimisation. Concept and gradient decent (see example in tutorial 6)

(See Tutorial 6)

MBQC (Petros' part)

- Lectures 23,24,26: Need to know

(i) basic MBQC patterns for single-qubit, two-qubit gates and how they are combined.

(ii) how to read an MBQC pattern to obtain the corresponding unitary implemented

(iii) how corrections work on angles (when measurement outcomes are not 0)

(iv) how to correct the output state (when measurement outcomes are not 0)

(v) how UBQC works (protocol and intuition). No details of the proof are required.

(See Tutorial 7)

QML (Petros' part)

- Lectures 27: Need to know

(i) General info on Supervised Learning (what it is, and how, inlc percetron and SVM in general, but not the details of SVMs)

(ii) The potential advantages (and obstacles) the QNN can have

(iii) QNN: encoding (be able to encode a bit string input to a quantum state, with the examples given in the lecture); training what cost function (cf VQAs for gradient decent); what is the output

(iv) Quantum Kernels: How and why to use Quantum Kernels. Be able to compute the Kernel (inner product) using quantum encoding given.

QEC Basics

- Lecture 25: Need to Know

i. Challenges of QEC: No-Cloning theorem; Multiple error types; Wavefunction collapse.

ii. Logical states: Understand subspace encoding and how this allows redundancy to be added to the system without cloning.

iii. Error detection via (Hadamard test) stabiliser measurement. Understand why the measurement of the auxiliary qubit in the stabiliser extraction circuit is deterministic and how this resolves the problem of wavefunction collapse.

iv. The digitisation of the error. Understand why it is sufficient to design quantum error correction codes that correct bit- (X) and phase (Z)-flip errors.

v. Pauli commutation rules. You should be able to determine whether two Pauli operators commute or anti-commute by counting the number of non-trivial intersections.

vi. Understand how the syndrome measurement depends on whether the error commutes or anti-commutes with the stabiliser being measured.

vii. You should be able to construct syndrome tables for a QEC code given its stabilisers.

viii. Logical operators: You should understand the difference between logical operators and stabilisers.

QEC: Stabiliser Formalism

- Lecture 28: Need to Know

i. Stabiliser formalism. Understand the definition of the stabiliser group and how it acts on the logical state. You should be able to explain why the stabiliser group must be abelian. You should understand how errors are detected by measuring a generating set of the stabiliser group.

ii. Logical qubit count: You should be able to compute the number of logical qubits encoded by a QEC code.

iii. Logical operators: What are the properties of logical operators? How do logical operators differ from stabilisers? Understand how to find logical operators based on their action on the logical basis states. Understand what happens when you combine a logical operator with a stabiliser.

iv. Code distance: You should understand the definition of the code distance and be able to determine the code distance from the logical operators. You should understand how the weight of logical operators can (sometimes) be reduced by multiplying by stabilisers. You should understand the difference between detection codes and correction codes.

v. Calculating code properties: Given a list of stabiliser generators, calculate the [[n,k,d]] parameters of the code. E.g. understand why the Steane code has parameters [[7,1,3]]

vi. The Shor code: Understand its construction via code concatenation.

vii. Determining success/failure of the QEC protocol: you should understand how the success/failure of the QEC process can be determined by checking whether the residual error is in the stabiliser group.

viii. Preparing the $|0\rangle_L$ basis state: understand how the $|0\rangle_L$ basis state can be prepared by measuring all of the stabilisers on the blank state.

QEC: Surface Codes

- Lecture 29: Need to Know

i. Calderbank-Shor-Steane (CSS) Code: What is a CSS code?

ii. Quantum Tanner graphs. Understand how the Tanner graph can represent a quantum error correction circuit. Given a Tanner graph, you should be able to write all the stabiliser generators of the code.

iii. Surface codes: understand how the construction guarantees that all the stabiliser generators commute.

iv. Understand the behaviour of X- and Z-type error chains in the surface code.

v. Logical operators: You should be able to identify logical operators in the surface code and understand why the surface code distance is equal to the width/height of the lattice.

Tutorials & Coursework Specifics

- Tutorials: all.
- Assignment/CW: No coding question in the exam. Still could be useful to see the theory of the CW.

Quantum Informatics CDT

THE UNIVERSITY of EDINBURGH

HERIOT WATT UNIVERSITY

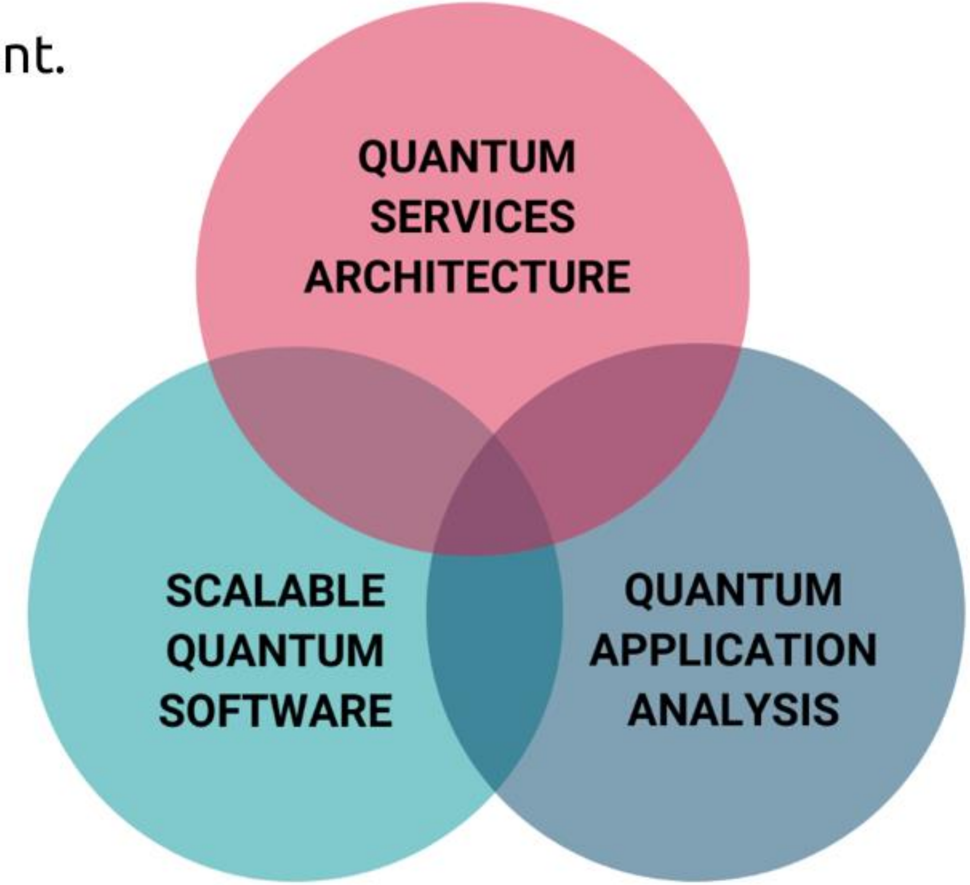
UCL

University of Strathclyde Glasgow

UNIVERSITY OF OXFORD

- 80 PhD studentships in 5 cohorts
- Fully funded 4 year programme
- Year 1 at Quantum Software Lab
- Years 2-4 at five leading Universities
- 70+ Academic experts
- 30+ Industry partners
- 10+ International universities
- Entrepreneurship programme
- Residential symposium
- Spring school

- Bridges computer science, mathematics, physics, and engineering.
- Make quantum technology interoperable and impactful.
- Interaction of quantum hardware, software, and applications.
- Social and ethical implications of innovation in quantum informatics.
- Unique collaboration of academia, industry, and government.



https://quantuminformatics-cdt.ac.uk