Quantum Computing and Quantum Internet

(Prof. Luciano Lenzini)

Motivations

The power of quantum computing lies in its ability to solve problems that are beyond the reach of classical computers. Tasks such as data encryption, optimization, and complex simulations can be revolutionized. For instance, Shor's algorithm can efficiently factor large numbers, potentially threatening classical encryption methods. This opens many new opportunities for computer engineers and software developers.

As quantum computing becomes more accessible, developers need the appropriate tools to harness its potential. Software Development Kits for working with quantum computers, like Qiskit, have emerged, enabling developers to write code for quantum computers. These SDKs bridge the gap between classical and quantum computing, simplifying exploration into the quantum realm for software developers. Mastering quantum programming languages is essential for leveraging the full capabilities of quantum computers, allowing for the development of innovative applications and solutions.

With the momentum of quantum computing growing, software developers are on the brink of a new frontier. Those who venture into the quantum realm may find solutions to some of the most complex problems of our time. The future is quantum, and it's up to developers to ride the wave and unlock its potential.

In response to the evolving landscape shaped by quantum computing, the computer engineering master's program for the 2024/2025 academic year makes available to its students a groundbreaking course titled "**Quantum Computing and Quantum Internet.**" This course focuses on quantum computers' programmability and interconnectedness within the quantum Internet.

Through this course, students gain a robust understanding of quantum computing theory, quantum Internet architectures and protocols, and quantum programming, through a blend of lectures, classroom exercises, and hands-on software laboratories. The course places a strong emphasis on quantum programming, equipping students with the skills needed to write, test, and debug quantum code. In the lab, Python, Qiskit, and NetSquid are used to translate theoretical concepts from lectures into code, which is then executed on simulators and/or real quantum computers. Students will learn to implement quantum algorithms, and quantum Internet protocols, develop quantum circuits, and utilize quantum error correction techniques, providing a comprehensive foundation in quantum software development.

To put it in a nutshell, this course:

- is at the leading edge of new emerging technology, disciplines, and industries;
- **emphasizes programming as the foundational approach;**
- provides an excellent opportunity to push your boundaries and rethink ***computation*** and ***networking*** from entirely novel perspectives;
- **DOES NOT** require quantum mechanics knowledge;
- **DOES NOT** cover qubits and quantum gates technologies.

Prerequisites

To succeed in this course, the following prerequisites are essential:

- A solid foundation in linear algebra, with deep insight into key concepts such as vector spaces, linear operators, change of bases, and matrices.
- Proficiency in Python programming.

Examination

The course consists of four weekly lessons, each lasting two hours, totaling 90 hours. The examination consists of an oral assessment and the successful completion of a project. Students can choose their projects from options provided by the professor, each requiring the writing of a Python program to implement quantum algorithms or quantum Internet protocols, among other possibilities. These projects will challenge students to apply their quantum programming knowledge to solve real-world problems, fostering innovation and practical skills essential for the future of quantum computing.

Detailed Program

Unit 1 - Single-Qubit System and the Framework of Quantum Mechanics

- Introduction
- Quantum Bit (Qubit) vs Classical Bit
- Physical Interpretation of a Qubit
- The First Postulate of Quantum Mechanics
- Mathematical Interpretation of a Qubit State
- Geometric Interpretation of a Qubit State Using the Bloch Sphere
- The Third Postulate of Quantum Mechanics and its Application to a Single-Qubit Measurement
- Relative Phase and Global Phase of a Qubit
- Single Qubit Gates
- Linear Combination (Basic Version of a Quantum Circuit) of Single Qubit Gates
- The Second Postulate of Quantum Mechanics and the Time Evolution of the State of a Qubit

Unit 2 - Quantum Programming

- OpenQASM: assembly for quantum computing
- IBM Quantum Experience
- Qiskit: a Frameworks for Quantum Software Development
- Qiskit Installation Guide
- Backends in Qiskit
- Qiskit Classical Simulators
- Run a Circuit on an IBM-Q Quantum Computer
- Running Several Configurations of Single Qubit Gates

Unit 3 - Mathematical Background

- Hilbert Spaces and Operators on Them
- The Dirac Notation (Bra-Ket Notation)
- Representation of Operators by Matrices
- The Spectral Theorem
- Functions of Operators
- Tensor Products

Unit 4 - Multi-Qubit System and the Framework of Quantum Mechanics

- The Fourth Postulate of Quantum Mechanics and the State Space of a Multi-Qubit System
- The Third Postulate of Quantum Mechanics and its Application to Multi-Qubit Measurement
- Bipartite and Multipartite Entangled States
- Monogamy of Entanglement
- Qiskit Programs Implementing the Above Quantum Concepts:
  - ✓ Build higher-dimensional Hadamard states
  - ✓ Measurements on Bell entangled states

Unit 5 - A Quantum Model of Computation

- The Quantum Circuit Model
- Controlled Qubit Gates
- Clues to Universal sets of Quantum Gates
- Qubit Copying Quantum Circuit? No-Cloning Theorem
- Implementing Measurements with Quantum Circuits
- Qiskit Programs Implementing the Above Quantum Concepts
  - ✓ Try implementing a copying circuit
  - ✓ Choose a measurement basis

Unit 6 - Quantum Teleportation and Superdense Coding

- Quantum Teleportation
- Superdense Coding
- Qiskit Programs Implementing the Above Quantum Algorithms

Unit 7 – Notions of Complexity

- Complexity
- Asymptotic Notation
- Complexity Classes

Unit 8 - Introductory Quantum Algorithms

- Quantum Parallelism, Quantum Interference, Phase Kick-Back
- The Deutsch Algorithm
- The Deutsch-Jozsa Algorithm
- Qiskit Programs Implementing the Above Quantum Algorithms

Unit 9 - The Density Operator

- Ensemble of *Pure* Quantum States
- The Density Operator as a *Mixed* State
- The Density Operator Formulation of Quantum Mechanics
- General Properties of the Density Operator
- The Reduced Density Operator
- The Geometry of Single-Qubit Mixed States

Unit 10 - Quantum Noise and Quantum Operations

- Classical Noise
- Quantum Operations
    - ✓ Overview
    - ✓ Environments and Quantum Operations
    - ✓ Operator-sum Representation
- Examples of Quantum Noise and Quantum Operations
    - ✓ Trace and Partial Trace
    - ✓ Geometric Picture of Single Qubit Quantum Operations
    - ✓ Bit Flip and Phase Flip Channels
    - ✓ Depolarizing Channel
    - ✓ Amplitude Damping
    - ✓ Phase Damping
- Applications of Quantum Operations
    - ✓ Master Equations
    - ✓ Quantum State Tomography
    - ✓ Limitations of the Quantum Operations Formalism
- Qiskit Programs Implementing the Above Quantum Concepts
    - ✓ Setting noise models in Qiskit
    - ✓ Characterizing noise with tomography

Unit 11 - Distance Measures for Quantum Information

- Distance Measures for Classical Information
- How Close are Two Quantum States?
    - ✓ Fidelity
- How well does a quantum channel preserve information?
- Qiskit Programs Implementing the Above Quantum Concepts
    - ✓ Python: computing fidelity with respect to a reference state
    - ✓ Evaluate fidelity realistically with Qiskit (tomography)

Unit 12 - Quantum Error-Correction: A General Framework

- Introduction
    - ✓ The Three-Qubit Bit Flip Code
    - ✓ Three-Qubit Phase Flip Code
- The Shor Code
- Qiskit Programs Implementing the Above Quantum Concepts
    - ✓ Three-Qubit Bit and Phase Flip Codes in action
    - ✓ Shor Code in action

Unit 13 - Quantum Internet Background

- Quantum Internet Motivations
- Components and Structure of the Quantum Internet
- Quantum Internet Protocol Stack
- The Role of Quantum Repeaters in the Quantum Internet
- Creating Entanglement Using Light
- Entanglement Distribution Between a Line of Quantum Repeaters
- Entanglement Swapping

Unit 14 - Quantum Purification Protocols

- Bennett's Protocol and Deutsch's protocol
- Basic Purification (Bit Flip Errors and Phase Flip Errors)
- Generalization by Incorporating Different Bell Pairs
- Multiple Rounds and Error Redistribution
- Scheduling Purification (Symmetric, Nested Pumping, Greedy, Banded)
- Purification and Entanglement Swapping-Based Repeaters
- Putting it all Together

Unit 15 - The Quantum Internet Simulator: NetSquid

- When should we use NetSquid
- NetSquid in a Nutshell
- Installation Guide
- Qubits, Components and Nodes
- Quantum memory and processor
- Nodes Behavior: Protocols
- Channels and Connections
- Ping Pong Simulation Example
- Collecting Metrics

Unit 16 - Programming Components and Protocols via NetSquid

- Entangled Photon Source Component
- A Simple Quantum Link Layer Protocol
- Re-implementing the Protocol on a Higher Abstraction Level
- Add Some Quantum Noise
- An Entanglement Distribution Protocol
- Collecting Throughput and Fidelity of End-to-End Entanglement

Unit 17 - Algorithms with Superpolinomial Speedup

- Quantum Fourier Transform and Its Inverse
- Quantum Phase/Eigenvalue Estimation
- Periodic States
- The Order-Finding Problem
- Shor's Approach to Order-Finding
- Qiskit Programs Implementing the Above Quantum Algorithms

Unit 18 - Algorithms Based on Amplitude Amplification

- Grover's Quantum Search Algorithm
- Amplitude Amplification
- Quantum Amplitude Estimation and Quantum Counting
- Qiskit Programs Implementing the Above Quantum Algorithms

Unit 19 - Quantum Key Distribution Protocols

- Quantum Key Distribution
  - ✓ Solution I: The BB84 Protocol
  - ✓ Solution II: The B92 Protocol
  - ✓ Solution III: The EPR Protocol
- Qiskit Programs Implementing the Above Protocols