Benefits of Using Quantum Computing Algorithms in the Risk Assessment of Geotechnical Structures

An Interim Report on UG Project

Submitted in Partial Fulfilment of the Requirements for the Degree of

Bachelor of Technology

In Mining Engineering

By

Arunanshu Sinha

Roll No: 21155020 {Pt. III, Sem No.}

Under the guidance of

Dr. Bharadwaj Pandit

Department of Mining Engineering

Indian Institute of Technology (BHU)

Varanasi-221005, India

May 2024

DECLARATION BY THE CANDIDATE

I, **Arunanshu Sinha**, certify that the work embodied in this UG Project Report is my bonafide work and carried out by me under the supervision of **Dr.Bharadwaj Pandit** in the Department of Mining Engineering, Indian Institute of Technology (BHU), Varanasi. The matter embodied in this report has not been submitted for the award of any other degree.

I declare that I have faithfully acknowledged and given credits to the research workers wherever their works have been cited in my work in this report. I further declare that I have not willfully copied any other's work, paragraphs, text, data, results, etc., reported in journals, books, magazines, reports dissertations, theses, etc., or available on websites, and have not included them in this report as my work.

Date: 01/05/2024

Place: Varanasi

Arunanshu Sinha

Roll No. 21155020

B. Tech. Part III, EvenSem

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the above statement made by the student is correct to the best of my knowledge.

Dr. Bharadwaj Pandit
Assistant Professor
Department of Mining Engineering
IIT – BHU. Varanasi

Dr. Suprakash Gupta
The Head
Department of Mining Engineering
IIT – BHU. Varanasi

Table of Contents

List of Figures

List of Tables

List of Abbreviations

MCS	Monte Carlo simulation
QMC	Quantum Monte Carlo
RAC	Risk Assessment Calculations
JCS	Joint wall Compressive Strength
LSA	Local Sensitivity Approaches
OAT	One At a time

Chapter 1. Introduction

Geotechnical structures in mining engineering are essentially engineered elements that ensure the stability and safety of mine excavations. They rely on soil and rock mechanics principles to address challenges specific to the mining environment. Here's a breakdown of their importance:

- **Stability of Excavations:** Both open pit mines (large surface pits) and underground mines rely on geotechnical structures to maintain stable slopes and prevent cave-ins. This protects workers and equipment and optimizes mine design for efficient extraction.
- **Tailings and Stockpile Management:** Mines produce waste rock and processed materials that need to be stored safely. Geotechnical engineers design and analyze tailings storage facilities and stockpiles to ensure they are stable and don't pose environmental risks.
- **Ground Support Systems:** Underground mines often require reinforcement to prevent rockfalls and maintain safe working conditions. Geotechnical engineers design ground support systems like rock bolts, shotcrete (sprayed concrete), and mesh to provide stability.
- **Risk Management:** Geotechnical investigations play a crucial role in assessing potential hazards like landslides or ground subsidence. By understanding the properties of the rock and soil, engineers can design structures and procedures to minimize risks.

In this study, the case of a retaining wall has been considered which is a widely used geo-technical structure in both mining and civil engineering. The applications of retaining walls in mines include.

- **Open Pit Mine Stability:** As miners dig deeper into an open pit, the sides become steeper and require support to prevent collapse. Retaining walls, particularly cantilever or buttress types, help maintain the stability of these slopes, ensuring safe operation and access to the ore body.
- **Stockpile Management:** Mines often stockpile large quantities of excavated material, like waste rock or low-grade ore. Retaining walls are used to create secure boundaries

for these stockpiles, preventing them from spilling over and creating safety hazards or environmental problems.

- **Tailings Dams:** Mining processes generate tailings, a slurry of leftover rock particles. Retaining walls are a key component of tailings dams, which safely store these tailings and prevent them from escaping into the environment.
- **Underground Support:** In some cases, retaining walls can be used within underground mines to reinforce specific areas. For instance, they might be used to create a stable platform for processing equipment or to prevent loose material from entering designated work zones.

The choice of retaining wall type for a mining application depends on factors like the height of the wall, the type of material it needs to retain, and the overall stability of the surrounding ground.

The optimisation and accuracy improvement of RAC of geotechnical structures in mining is critical for several reasons:

Safety: These structures are vital for preventing failures like collapses, landslides, or dam breaches. A proper risk assessment helps identify potential hazards and their likelihood of occurring, allowing engineers to design structures that can withstand these threats and keep workers safe.

Economic Efficiency: Catastrophic failures of geotechnical structures can be incredibly expensive. Risk assessment helps optimize designs by identifying areas where additional measures are truly necessary, avoiding unnecessary costs while still maintaining safety.

Environmental Protection: Mining operations can have a significant environmental impact. Tailings dam failures, for example, can cause major pollution events. Risk assessment helps ensure these structures are designed and built to minimize environmental risks.

The current RAC techniques for geotechnical structures in mining typically involve a combination of qualitative and quantitative methods:

- **Qualitative Assessment:** This involves identifying potential failure modes (how the structure might fail), their causes (events that trigger failure), and their consequences (impact of failure). Experience and expert judgment play a major role in this stage.

- **Quantitative Assessment:** Here, engineers assign probabilities to the identified failure modes and their consequences. This might involve statistical analysis of historical data, computer modeling, or a combination of both.

Common Tools and Approaches:

- **Event Tree Analysis:** This method maps out potential sequences of events that could lead to failure, allowing engineers to identify critical points for intervention.
- **Fault Tree Analysis:** This approach works backward, starting from a potential failure and then identifying all the possible combinations of events that could cause it.
- **Numerical Modeling:** Numerical modelling methods like Monte Carlo simulations are software programs that can be used to simulate the behaviour of geotechnical structures under various loading conditions, helping assess stability and identify potential weaknesses

Table 1.1. Limitations of the current techniques versus how quantum computers can help in optimising risk analysis calculations.

Classical Computing Approach	Quantum Computing Approach
Computational Complexity: Simulating complex geological conditions and interactions between different materials (rock, soil, water) can be computationally expensive with traditional computers. This limits the number of variables that can be considered, potentially leading to inaccurate assessments.	**Faster Simulations:** Quantum computers, with their ability to perform calculations in parallel, could significantly speed up complex simulations of geotechnical structures. This would allow engineers to consider a wider range of variables and scenarios, leading to more comprehensive risk assessments.
Data Dependence: The accuracy of risk assessments heavily relies on the quality and quantity of available data. Limited historical data or a lack of data on specific	**Advanced Data Analysis:** Quantum algorithms have the potential to analyze vast amounts of data from various sources, including sensor data, historical records, and geological surveys. This could help identify

geological features can lead to underestimating risks.	patterns and relationships that might be missed by traditional methods, leading to more informed risk assessments.
Limited Modelling Capabilities: Traditional computer models may struggle to accurately capture the non-linear behavior of geotechnical materials, especially when dealing with large or complex structures.	**Improved Material Modelling:** Quantum computers could be used to develop more sophisticated models that can better capture the non-linear behavior of geotechnical materials. This would lead to more accurate predictions of how these structures will perform under different loading conditions.

During working on this project we might encounter a couple of roadblocks that are mainly due to the current restrictions and associated difficulties of practical usage of quantum computing. 1 of the crucial problems is that quantum hardware have a limited number of qubits. The conduct of risk assessment simulations for geotechnical structures that involve complicated systems with many variables frequently exceed the current limits of processing power represented in the existing quantum processors. Also, the introduction of various errors in the process of quantum computations is a principal issue. Quantum systems by the very nature of them tend to be noisy and prone to decoherence resulting in numerical error. Error correction techniques are the necessary prerequisites for diminishing these issues, though they usually need more qubits and computational assets, making the qubit lack even greater. To break through these barriers, we will be in need of hardware and software fixes, the widening of error correction protocols, and the enlarging of the number of qubits in quantum processors. Collaboration between quantum computing and domain experts such as geotechnical engineers will be be the more crucial in solving these obstacles and achieving the strongest results from the application of quantum computing in risk assessment and mining engineering.

1.1 Objectives

The primary objectives of this report are – i) To study the current risk assessment methods using monte-carlo simulations and identify their limitations ii) To develop a Quantum monte Carlo algorithm to perform the risk assessment simulation on a Quantum computer using IBM Qiskit and Python iii) Compare the results from both the methods and highlight the improvement in efficiency and discuss a future plan of execution in actual mines.

1.2 Organization of the report

Following the brief introduction of the research problem, this report comprises of six chapters (Chapter 1 to Chapter 6). A brief overview of each chapter is given below.

Chapter 1 (This chapter) provides an introduction to the topic of study, does a basic comparison of the current approach with proposed quantum computing approach and why research on this problem statement has significant importance and can have great future applications

Chapter 2 Literature review is conducted on the probabilistic analysis of retaining walls and discusses how a system of safety ratios and monte carlo simulator programs are used presently in for the risk assessment of the geotechnical structure. It also discusses an approach of implementing a simple monte carlo simulation using code and the limitations in the current method.

Chapter 3 In the theoretical study chapter we dive deeper into the quantum algorithm design approach, understand the differences between a classical computer algorithm and a quantum computer algorithm and also discuss the basic building blocks of quantum algorithms.

Chapter 4 We discuss the undertaken case of a retaining wall in detail and explain how computer code in both the classical and quantum approaches.

Chapter 5 Provides results and discussion on the final findings of our study

Chapter 6 Provides the conclusions and future scope of study to further implement the quantum circuit design and eventually compare cost and efficiency of both approaches

Chapter 2. Literature Review

Let's explore the existing method of reliability calculation of a concrete retaining wall, we choose the cantilever retaining wall amongst other types (counterfort wall, buttress wall, basement wall) for the sake of simplifying calculations.

2.1 Probabilistic analysis of the retaining wall

Conventional designs of external stability has been based on deterministic methods and on the concept of Factors of Safety (FS). The stability of the wall is examined as a system consisting of different modes of failure. Typically, these modes are due to the (i) Overturning of the wall about its toe (ii) Due to sliding along its base (iii) due to bearing capacity inadequacy of the foundation soil.

A methodology is presented for the reliability analysis of retaining walls that takes into account the dependency between different failure modes. This dependency is investigated and quantified through their correlation, and its influence on the system reliability is addressed. Computations are performed using a Monte Carlo simulation algorithm for assumed probability distributions of the random variables. Monte Carlo simulation is a stochastic technique that generates a great number of repeated simulation processes (realizations). Each simulation is based on generating a series of values of one or more random variables. The procedure requires the complete definition of the random variable and probability distributions, but through simple computations, it provides empirical outcomes of numerically simulated random realizations of the safety ratios.

The expression for the sliding safety ratio

$$SR_{\mathrm{SL}} = \frac{\sum \mathrm{F_R}}{\sum \mathrm{F_{SL}}} = \frac{P_{\mathrm{P}} + S}{P_{\mathrm{A}}}$$

The expression for the bearing capacity safety ratio

$$SR_{\mathrm{BC}} = \frac{\mathrm{q_{b,L}}}{\mathrm{q_{max}}}$$

The expression for the eccentricity safety ratio

$$SR_e = \frac{B/6}{|e|}$$

The final reliability is given by the probability of all of the failure criteria safety ratios being greater than 1

$$R = P\left[(SR_e \geqslant 1) \cap (SR_{SL} \geqslant 1) \cap (SR_{BC} \geqslant 1)\right]$$

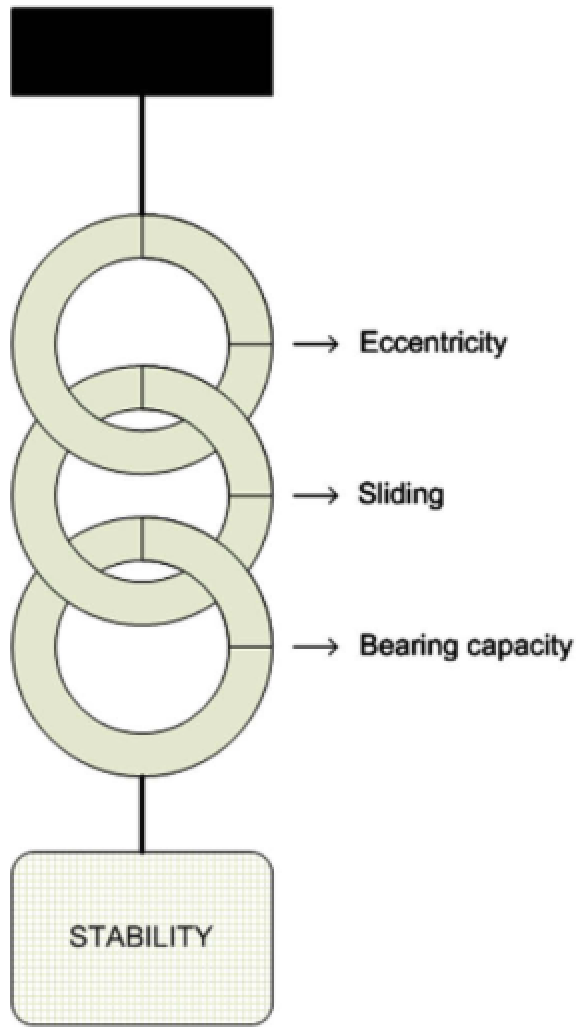

Fig 2.1 An illustration of the series system concept

The global stability of the wall is shown at the end of the chain whose links (components) are the three modes of instability. If any of the links breaks, then the global stability is lost.

2.2 Steps of Implementing a Classical Monte Carlo simulation

Geotechnical structures, like foundations, slopes, and retaining walls, are subject to various uncertainties:

- Soil properties (strength, stiffness, density) may vary spatially.
- Load conditions (dead weight, live loads) might be imprecise.
- Construction processes can introduce variability.

Identifying key parameters: Define the input variables that influence the geotechnical structure's behaviour (e.g., soil strength, unit weight, loads).

Assigning probability distributions: Based on on-site investigation data and engineering judgment, assign probability distributions to each parameter (e.g., normal, lognormal) that reflect the expected range and variability.

Random sampling: The core of MCS. Use appropriate random number generators to sample values for each parameter from their assigned distributions.

Deterministic model: Develop a model (analytical, numerical) that calculates the performance metric of interest (e.g., settlement, bearing capacity) for a given set of input parameters.

Repeated simulations: Run the model a large number of times (thousands or more) using different random samples from the input parameter distributions.

Statistical analysis: Analyze the resulting output data (performance metric values) to understand the probability distribution of the outcome. Calculate statistics like mean, standard deviation, and percentiles (e.g., 5th percentile for critical scenarios).

2.3 Limitations of Classical Monte Carlo simulation

Material Behavior:

- **Non-linearity:** Classical methods often rely on simplified material models. Soil and rock behavior can be non-linear, exhibiting complex stress-strain relationships. This can lead to inaccuracies in predicting failure mechanisms, especially for highly stressed retaining walls.

- **Heterogeneity:** Classical models might struggle to capture the inherent heterogeneity of soil and rock formations. Variations in density, strength, and presence of fractures can significantly impact retaining wall stability.

Uncertainty Quantification:

- **Limited Stochastic Analysis:** While classical methods can handle some uncertainty through probabilistic approaches like Monte Carlo simulations, they might not fully capture the complex interactions of multiple uncertainties.
- **Difficulties with Rare Events:** Classical methods might struggle to accurately predict the probability of rare events like extreme weather or seismic activity that could lead to retaining wall failure.

Computational Cost:

- **Complex Models:** Highly detailed classical models incorporating advanced material behavior and heterogeneity can be computationally expensive, especially for large-scale retaining wall projects.

Quantum computing offers potential advantages in addressing some of these limitations:

- **Simulating Complex Systems:** Quantum algorithms might one day be able to efficiently model complex material behavior and soil heterogeneity, leading to more accurate predictions for retaining wall stability.
- **Uncertainty Quantification:** Quantum algorithms could potentially handle high-dimensional uncertainty spaces more efficiently, providing a more comprehensive risk assessment for retaining walls.

Chapter 3. Theoretical Study

Quantum computing is an emerging technology that uses the laws of quantum physics to perform computations. Quantum computations use phenomena in quantum physics, such as superposition and entanglement to perform computations. In contrast to classical computing where information is stored as binary bits, which can only be in either the 0 or 1 state, quantum computing uses quantum bits, or *qubits*, which can be in the 0 and 1 states at the same time. Because of this fundamental difference, quantum computers have the potential to outperform classical computers when solving certain types of problems, such as optimization or simulation.

3.1 Building block of quantum algorithms - Qubit

The qubit is the basic building block of quantum computing. Qubits store information and can be physically realized by two-state quantum devices. A qubit can be in a linear combination of the $|0\rangle$and $|1\rangle$ states with complex coefficients, referred to as a *superposition*. If the complex coefficients are normalized, then they represent the probability amplitudes of measuring the qubit in the $|0\rangle$or $|1\rangle$ state.

3.2 Building block of quantum algorithms - Quantum gates

A qubit is useless unless you can use it to carry out a quantum calculation. And these quantum calculations are achieved by performing a series of fundamental operations, known as quantum logic gates. There are lots of types of quantum gates. There are single-qubit gates, which can flip a qubit from 0 to 1 as well as allowing superposition states to be created. Then there are also two-qubit gates. These allow the qubits to interact with each other and can be used to create quantum entanglement: a state of two or more qubits that are correlated in a way. Quantum gates represent reversible operations that transform the quantum state according to unitary matrices. While all gate operations are deterministic, measuring in quantum computing is probabilistic, with the probabilities of various measurements depending on the states of the qubits.

An example of a gate is the controlled-NOT (CNOT) gate. It's a two-qubit operation where the first qubit is labelled the control qubit and the second one is the target qubit. If the control qubit is $|1>$ then it will flip the target's qubit state from $|0>$ to $|1>$or vice versa.

(Note: the | and > used in the notation are just to remind us that we're talking about vectors that represent the qubit states labelled 0 and 1)

Similar to traditional computers, quantum algorithms with many qubits can be broken down into a sequence of single and two-qubit quantum gates, forming a universal quantum gate set.

Therefore, if we can get these quantum gates to work in a reliable way that scales to many qubits, we can use them to run all possible algorithms on our quantum computer.

Creation Function	Gate Name	No. of Qubits	Matrix Representation
H `hGate`	Hadamard gate	1	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
ID `idGate`	Identity gate	1	$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$
X `xGate`	Pauli X gate	1	$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$
Y `yGate`	Pauli Y gate	1	$\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$
Z `zGate`	Pauli Z gate	1	$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$

Fig 3.1 The different one target qubit quantum gates

There are several types of quantum gates to enable multiple types of algorithms

- Gates on One Target Qubit
- Rotation Gates
- Gates with One Control Qubit and One Target Qubit
- Gate That Swap States of Two Qubits
- Controlled Rotation Gates
- Ising Coupling Gates
- Composite and Specialized Gates

3.3 Building block of quantum algorithms - Quantum circuit

In computer science, *circuits* are models of computation in which information is carried by wires through a network of *gates*, which represent operations that transform the information carried by the wires. In the quantum circuit model, wires represent qubits and gates represent operations acting on these qubits. We'll focus for now on operations we've encountered so far, namely *unitary operations* and *standard basis measurements*. As we learn about other sorts of quantum operations and measurements, we'll enhance our model accordingly.

Here's a simple example of a quantum circuit:

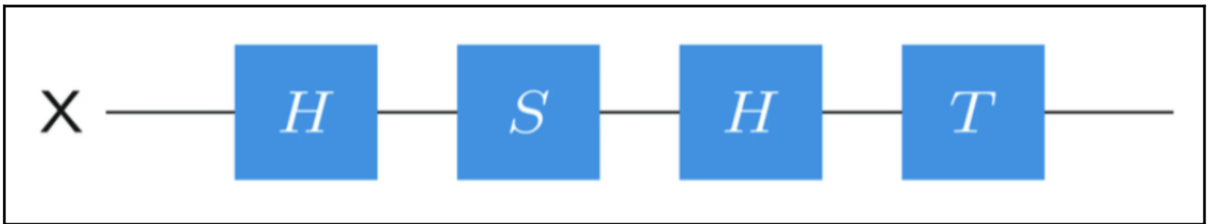

Fig 3.2 An example Quantum Circuit

In this circuit, we have a single qubit named X,, which is represented by the horizontal line, and a sequence of gates representing unitary operations on this qubit. Just like in the examples above, the flow of information goes from left to right

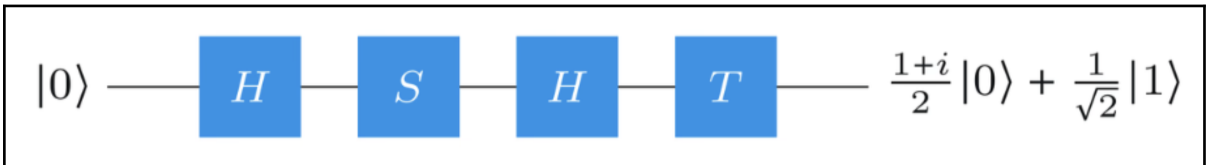

Fig 3.3 Quantum circuit with its output showing the superposition state of the qubit

Designing for a Specific Problem:

- Quantum algorithms, which are analogous to classical algorithms, outline the steps needed to solve a problem using a quantum computer. However, these algorithms still need to be translated into concrete instructions for the specific hardware.

- The arrangement of gates and how they interact with qubits within the circuit determine the specific operations the quantum computer performs. To solve a new problem, a new sequence of gates and manipulations must be designed to exploit the advantages of quantum mechanics.

Adaptability and Challenges:

- Unlike classical computers where the same set of logic gates can be applied to various problems, quantum circuits are highly problem-specific.
- Designing efficient quantum circuits for new problems is an active research area. It requires understanding the problem, the available quantum hardware capabilities (e.g., number of qubits), and expertise in quantum algorithms and circuit design.

Existing vs. New Circuits:

- While some basic quantum circuits might be reusable for specific subroutines within different algorithms, it's unlikely that a complete circuit designed for one problem will directly solve another.
- New problems with unique computational demands necessitate tailoring the circuit to leverage the power of quantum superposition and entanglement effectively.

3.4 The ideal approach to developing a quantum Monte Carlo algorithm to do risk assessment calculations of a geotechnical structure

1. **Problem Formulation**: Define the problem in terms of geotechnical engineering. For a retaining wall, this involves considering factors such as soil properties, wall design, external loads, and potential failure modes (e.g., sliding, overturning, bearing capacity failure).
2. **Quantum Representation**: Map the geotechnical problem onto a quantum system. This could involve representing the soil properties, wall design parameters, and external loads as quantum states or operators in a quantum circuit. The goal is to encode the relevant aspects of the problem into a quantum format that can be manipulated and analyzed using quantum algorithms.

3. **Hamiltonian Encoding**: Construct the Hamiltonian of the geotechnical system. The Hamiltonian represents the total energy of the system and is crucial for performing quantum simulations. It should include terms corresponding to the potential energy of the soil, the energy associated with the wall structure, and any external forces or constraints.
4. **Quantum Circuit Design**: Design quantum circuits that implement the Hamiltonian of the geotechnical system. This may involve decomposing the Hamiltonian into a series of quantum gates and constructing a quantum circuit that prepares the ground state of the system or evolves it over time. Qiskit provides tools for building and manipulating quantum circuits programmatically in Python.
5. **Variational Quantum Monte Carlo (VQMC)**: Utilize VQMC or other quantum algorithms to simulate the behavior of the geotechnical system. VQMC involves optimizing the parameters of a parameterized quantum circuit to minimize the energy of the system, which corresponds to finding the most stable configuration of the retaining wall under different conditions.
6. **Sampling and Analysis**: Use Monte Carlo methods to sample configurations of the quantum system and analyze the results. This could involve running the quantum circuit multiple times to obtain statistical information about the stability of the retaining wall under various scenarios, such as different soil conditions or loading conditions.
7. **Classical Post-Processing**: Perform classical post-processing of the simulation results to extract meaningful insights and assess the risk associated with the retaining wall. This may involve computing probabilities of failure modes, estimating safety factors, or optimizing design parameters to minimize risk.
8. **Error Mitigation and Validation**: Account for errors inherent in quantum computing, such as gate errors and measurement errors, and apply error mitigation techniques to improve the accuracy of the simulation results. Additionally, validate the results of the quantum simulation against classical methods or experimental data to ensure their reliability and usefulness for real-world applications.

Chapter 4. Description of the Case study and computer code

In this report we have chosen the case study of a retaining wall and here we will discuss in detail the basic implementation of an MCS and a QMC.

4.1 Computer code implementation of MCS on a classical computer

Identifying key parameters: Define the input variables that influence the geotechnical structure's behavior (e.g., soil strength, unit weight, loads).

Assigning probability distributions: Based on site investigation data and engineering judgment, assign probability distributions to each parameter (e.g., normal, lognormal) that reflect the expected range and variability.

Random sampling: The core of MCS. Use appropriate random number generators to sample values for each parameter from their assigned distributions.

Deterministic model: Develop a model (analytical, numerical) that calculates the performance metric of interest (e.g., settlement, bearing capacity) for a given set of input parameters.

Repeated simulations: Run the model a large number of times (thousands or more) using different random samples from the input parameter distributions.

Statistical analysis: Analyze the resulting output data (performance metric values) to understand the probability distribution of the outcome. Calculate statistics like mean, standard deviation, percentiles

```python
import numpy as np
from scipy.stats import norm  # Assuming normal distribution for an input parameter

# Define input parameter (e.g., soil strength) and its distribution
soil_strength_mean = 100  # kPa
soil_strength_std = 10  # kPa
soil_strength_dist = norm(loc=soil_strength_mean, scale=soil_strength_std)

# Number of simulations
num_simulations = 10000

# Deterministic model (replace with your actual model)
def calculate_settlement(soil_strength):
    # Simplified model: settlement inversely proportional to soil strength
    return 1 / soil_strength

# Perform simulations
settlement_values = np.zeros(num_simulations)
for i in range(num_simulations):
    sampled_strength = soil_strength_dist.rvs()  # Sample soil strength
    settlement_values[i] = calculate_settlement(sampled_strength)

# Analyze results
mean_settlement = np.mean(settlement_values)
std_settlement = np.std(settlement_values)
percentile_settlement = np.percentile(settlement_values, 5)  # 5th percentile (critical)

print(f"Mean settlement: {mean_settlement:.2f} kPa")
print(f"Standard deviation of settlement: {std_settlement:.2f} kPa")
```

Fig 4.1 Conceptual code of MCS implementation on a classical computer in Python

Deterministic Model: The code defines a simplified model function `calculate_settlement(soil_strength)`. This function calculates the settlement based on a single input parameter, `soil_strength`. In this example, it's a simplified relationship where settlement is inversely proportional to soil strength (stronger soil settles less).

MCS and Settlement Distribution: The MCS performs numerous simulations by sampling values for soil strength from its probability distribution. This simulates the uncertainty in the actual soil strength at the site.

Settlement Values: For each simulation, the `calculate_settlement` function is called with the sampled soil strength, resulting in a corresponding settlement value. This generates a series of settlement values representing possible outcomes due to the variability in soil strength.

Analyzing Settlement Results: Finally, the code calculates statistics like mean settlement, standard deviation, and percentiles (e.g., 5th percentile) of the settlement values. These statistics provide insights into the range of potential settlements and the likelihood of exceeding critical values.

4.2 Computer code implementation of a QMC on a quantum computer

4.2.1 Setting up the quantum computing environment on our local machine

Quantum computing with Qiskit, IBM's open-source quantum computing framework, offers a versatile platform for both simulation and execution on real quantum hardware. Using Qiskit's simulators, users can model quantum circuits and algorithms without the constraints of physical qubits. This allows for rapid prototyping, debugging, and exploration of quantum algorithms' behavior. Simulators provide an accessible entry point for learning and experimentation, facilitating the development of quantum intuition and understanding. Transitioning to actual quantum hardware involves connecting to IBM's Quantum Experience or other quantum processors available through the cloud. Qiskit abstracts away much of the complexity, offering a seamless interface for submitting quantum circuits to real devices. However, users must contend with noise, decoherence, and limited qubit connectivity inherent in current quantum processors. This necessitates techniques such as error mitigation and optimization to achieve meaningful results. Thus, Qiskit empowers users to explore the potential of quantum computing from simulation to real-world implementation, fostering innovation and advancement in the field.

To connect to IBM's quantum computing resources using Qiskit, start by signing up for an IBM Quantum account and obtaining your API key. Once you have your API key, install Qiskit using pip if you haven't already. In your Python script or Jupyter Notebook, import the necessary modules from Qiskit and set up your IBM Quantum account using your API key. After saving your API key, load your IBM Quantum account in Qiskit using the provided command. With your account loaded, you can now access IBM's quantum computing resources, including simulators and real quantum devices, through Qiskit. This allows you to run experiments on simulators or submit jobs to real quantum devices for execution, enabling you to explore quantum algorithms and their behaviour in both simulated and real-world environments seamlessly.

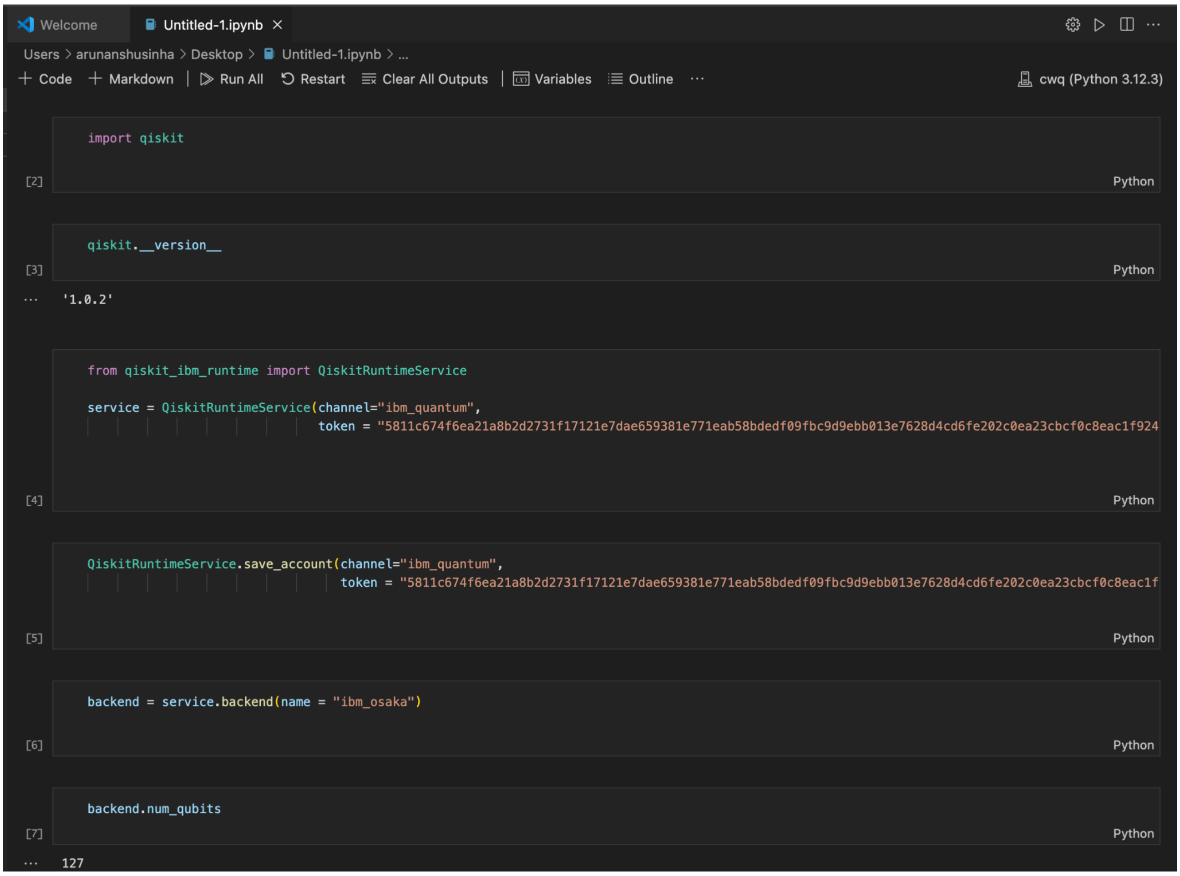

Fig 4.2 Getting access to quantum hardware using Qiskit's API

4.2.2 Implementing the Quantum Monte Carlo algorithm

Users > arunanshusinha > Desktop > Untitled-1.ipynb > ...

+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline

```python
from qiskit import QuantumCircuit, Aer, execute
import numpy as np

# Define the quantum circuit
def build_circuit(params):
    # Create a quantum circuit with the necessary qubits and classical bits
    circuit = QuantumCircuit(2, 2)

    # Apply quantum gates based on the parameters
    circuit.rx(params[0], 0)
    circuit.ry(params[1], 1)

    # Measure the qubits
    circuit.measure([0, 1], [0, 1])

    return circuit

# Simulate the quantum circuit
def run_simulation(circuit, shots=1024):
    # Use the Aer simulator
    simulator = Aer.get_backend('qasm_simulator')

    # Execute the circuit on the simulator for a certain number of shots
    job = execute(circuit, simulator, shots=shots)

    # Get the results
    result = job.result()
    counts = result.get_counts(circuit)

    return counts

# Example parameters
params = np.pi/2 * np.random.rand(2)

# Build and run the circuit
qc = build_circuit(params)
counts = run_simulation(qc)

print("Results:", counts)
```

Fig 4.3 Conceptual code of QMC implementation on a quantum computer in Qiskit

The provided Python code utilizes Qiskit, a library for quantum computing, along with NumPy for numerical operations. It defines functions to build and simulate a basic quantum circuit. The build_circuit function constructs a quantum circuit with two qubits and applies single-qubit rotation gates, rx and ry, with parameters determined by the input list params. After applying the gates, the circuit measures the qubits and stores the results in classical bits. The run_simulation function executes the quantum circuit using Qiskit's Aer simulator, specifying the number of repetitions with the shots parameter. It then retrieves the measurement outcomes and returns the counts of each outcome. An example usage demonstrates generating random parameters, constructing a quantum circuit, simulating it, and printing the measurement results. This code serves as a foundational framework for developing more complex quantum simulations using Qiskit in Python, adaptable to diverse quantum computing applications.

Retaining wall analysis relies on well-established classical mechanics principles. Finite element analysis (FEA) software already tackles these problems efficiently like OpenSees - https://opensees.berkeley.edu/.

Quantum Advantage is yet not Achieved: Simulating a retaining wall with QMC would require a massive number of qubits to represent the soil particles, forces, and material properties. This exceeds current capabilities for achieving a "quantum advantage" over classical methods.

Chapter 5. Results and Discussions

This section presents the proposed approach and preliminary findings towards utilizing quantum computing for risk assessment in geotechnical structures, focusing on retaining walls. While the implementation of the Quantum Monte Carlo (QMC) algorithm using IBM Qiskit and Python is underway, the discussion here revolves around the conceptual framework and anticipated outcomes based on the proposed methodology.

1. Current Risk Assessment Methods and Limitations: A comprehensive study of existing risk assessment techniques employing Monte Carlo simulations revealed several limitations. Classical Monte Carlo methods often face challenges in handling complex systems with high-dimensional parameter spaces. Additionally, the computational resources required for extensive sampling hinder the scalability of these approaches, particularly in scenarios involving large-scale geotechnical structures.

2. Development of Quantum Monte Carlo Algorithm: In response to the identified limitations, the primary objective of this project is to develop a Quantum Monte Carlo algorithm tailored for risk assessment in geotechnical structures. Leveraging the principles of quantum computing, the algorithm aims to exploit quantum parallelism and superposition to efficiently explore the parameter space and simulate the behavior of retaining walls under various conditions. The conceptual design of the QMC algorithm involves mapping the risk assessment problem onto a quantum circuit architecture, utilizing quantum gates to represent system dynamics, and employing measurement outcomes to estimate risk probabilities.

3. Anticipated Benefits of Quantum Computing: The integration of quantum computing in risk assessment offers several potential advantages over classical methods. Quantum algorithms, such as QMC, have the capability to explore exponentially large solution spaces in parallel, enabling more efficient sampling and faster convergence towards accurate risk estimates. By harnessing the computational power of quantum systems, we anticipate significant improvements in the efficiency and accuracy of risk assessment for geotechnical structures. The ability to handle complex systems with higher precision and scalability could revolutionize the field, facilitating better-informed decision-making and enhanced safety measures.

Chapter 6. Conclusions and Future scope of study

To finally conclude, in this interim report, I have discussed the possible advantages of applying quantum computing to geotechnical structural risk assessment, with a case study on retaining walls. After recognizing the current limitations of conventional risk analysis techniques, particularly Monte Carlo simulations, in the project, we have discussed the basic study of developing quantum algorithms and the approach of a making a Quantum Monte Carlo (QMC) algorithm utilizing IBM Qiskit and Python. While the QMC implementation is still in the process, the procedures that have been detailed offer a thorough grasp and lay down the approach of how to use quantum computing for risk assessment jobs.

Once the comparison between classical Monte Carlo simulations and Quantum Monte Carlo simulations. It is expected that we will be able to highlight the efficiency improvements offered by quantum computing. By exploiting the foundational ability of parallelism and computational power of quantum systems, we expect to observe significant enhancements in the speed and accuracy of risk assessment for geotechnical structures. These improvements can potentially add value to the field, enabling more accurate predictions and before-hand mitigation strategies.

Looking ahead, the future scope of this study involves further study of the Quantum Monte Carlo algorithm and its application in real-world scenarios, particularly in the context of mining operations. A big aspect of this would be the generalisation of this study and putting it to use to other applications. Additionally, in the coming semester, deeper exploration into the design of quantum circuits specifically to solve the undertaken geotechnical risk assessment problems will be pursued. As the project progresses, in the coming iterations I will delve into the intricacies of quantum computing, refining methodologies, and exploring new approaches to address challenges in geotechnical risk assessment.

Moving forward, in the next semester I will be dedicatedly continuing research and development, focusing on completing the implementation of the Quantum Monte Carlo simulation and conducting comprehensive analyses of its performance compared to classical methods. The nature of the project is challenging and aspirational but I am sure through collaboration with experts in both quantum computing and geotechnical engineering, this project will be able to contribute to the advancement of risk assessment techniques and ignite a line of thought in the minds of researchers worldwide for the integration of quantum technologies in mining engineering applications.

References

Ioannis E. Zevgolis, Philippe L. Bourdeau, (2010). Probabilistic analysis of retaining walls, Computers and Geotechnics.

N.J. Cerf, S.E. Koonin, (2007). Monte Carlo simulation of quantum computation1Presented at the IMACS Seminar on Monte Carlo Methods. Mathematics and Computers in Simulation

https://docs.quantum.ibm.com/

https://in.mathworks.com/help/matlab/math/quantum-monte-carlo-simulation.html

https://in.mathworks.com/help/matlab/math/introduction-to-quantum-computing.html