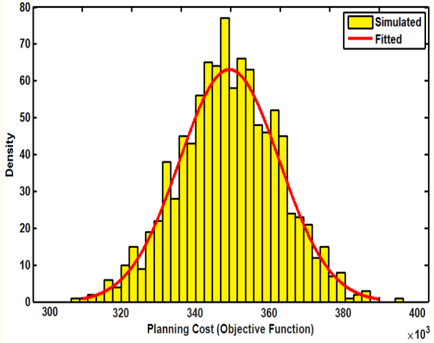
Risk Analysis of Federal Bonds Using Quantum Computing



SE20UECM045-Matta Rithik Reddy

Under the Guidance of :Prof D. Jayasri

Problem Statement

This project aims to compare the efficacy and efficiency of Monte Carlo simulations and Quantum Computing Amplitude Estimation (QAE) in assessing the risk associated with Federal Treasury bonds (T-bonds). By comparing accuracy, efficiency, scalability, and practicality, we aim to reveal the strengths and limitations of both methods for risk analysis.

This approach addresses the need to integrate quantum computing into traditional risk assessment, offering crucial insights for investors and policymakers in bond market navigation.

Table of contents

01 What are Federal Treasury Bonds ?

Basic Info, Types of Bonds, Risks Involved

02 What is Risk Analysis ?

Basic Definition, Types and Measurement variables

03 Monte Carlo Method Simulation

Definition, Process, Dataset, Implementation, Observations and Drawbacks

04 Quantum Amplitude Estimation Method

Definition Process, Dataset and Implementation, Observations and Drawbacks

01

Federal Bonds

The second most safest investment in the world

Definition of Treasury Bond

Treasury bonds, also known as T-bonds or government bonds, are long-term debt securities issued by a country's government to finance public spending and manage its debt obligations. Treasury bonds are issued with maturities of 20 or 30 years. They are issued with a minimum denomination of $100, and coupon payments on the bonds are paid semi-annually.

The bonds are initially sold through an auction; the maximum purchase amount is $5 million if the bid is non-competitive (or 35% of the offering if the bid is competitive).

Different Types of Treasury Bonds

Nominal Treasury Bonds:

(Coupon Bonds)

- Nominal bonds, also known as coupon bonds, are issued with a specified face value (the amount repaid at maturity) and pay periodic interest payments (coupons) to bondholders. These interest payments are typically made semi-annually.
- The interest rate paid on nominal bonds is called the coupon rate, and it's determined at the time of issuance. The coupon rate is applied to the face value to determine the amount of each interest payment.
- Investors who hold nominal bonds receive regular interest payments until the bond matures, at which point they receive the face value of the bond.

Different Types of Treasury Bonds

Zero-Coupon Treasury Bonds:

- Zero-coupon bonds are issued at a discount to their face value. Unlike nominal bonds, they do not make periodic interest payments (hence the term "zero-coupon"). Instead, investors purchase these bonds at a discounted price and receive the full face value at maturity.

- Because zero-coupon bonds do not make regular interest payments, they tend to be more sensitive to changes in interest rates. Their prices fluctuate more dramatically in response to changes in the market interest rate.

- The return on investment for zero-coupon bonds comes from the difference between the purchase price and the face value when the bond matures. This return is effectively the interest earned on the investment over the bond's term.

~$26.9T

Total Outstanding Amount
(April 2024)

Interest Rate (Repo Rate) Increases Bond Costs Fall

~$892.8B

Monthly Trading in (April 2024)

02

Risk Analysis

Checking all the possibilities of Disaster

Text Book Definition of Risk Analysis

Risk analysis is the process of identifying, assessing, and prioritizing potential risks or uncertainties that could adversely affect the achievement of objectives or the success of a project, investment, or organization.

In this scenario, It’s about determining the maximum potential loss associated with investing in Federal bonds across various market interest rate conditions.

Different Types of Risk by Investing In any Financial Instruments

Market Risk

Arises from fluctuations in the overall market, affecting the value of all investments.

Credit Risk

The risk of loss due to a borrower failing to repay a loan or meet contractual obligations.

Liquidity Risk

The risk of not being able to sell an investment quickly at a fair price.

Operational Risk

Arises from internal processes, systems, or human error within an organization.

Political Regulatory Risk

Arises from changes in government policies, regulations, or political instability.

Event Risk

Arises from specific events such as natural disasters, terrorist attacks, or corporate scandals.

Measurement Metrics of Risk

Value at Risk (VaR)

VaR quantifies the maximum potential loss of an investment or portfolio within a specified time frame and with a certain level of confidence.

Conditional Value-at-Risk (CVaR)

Also known as expected shortfall, CVaR measures the expected loss of an investment portfolio in the worst-case scenarios beyond a certain confidence level.

VaR

It provides the maximum loss that will not be exceeded with a specified confidence level. In simpler terms, VaR tells you how much you could potentially lose with a certain probability.

The VaR at a 95% confidence level indicates that there is a 5% chance that the portfolio could lose more than the VaR value in the specified time period.

CVaR

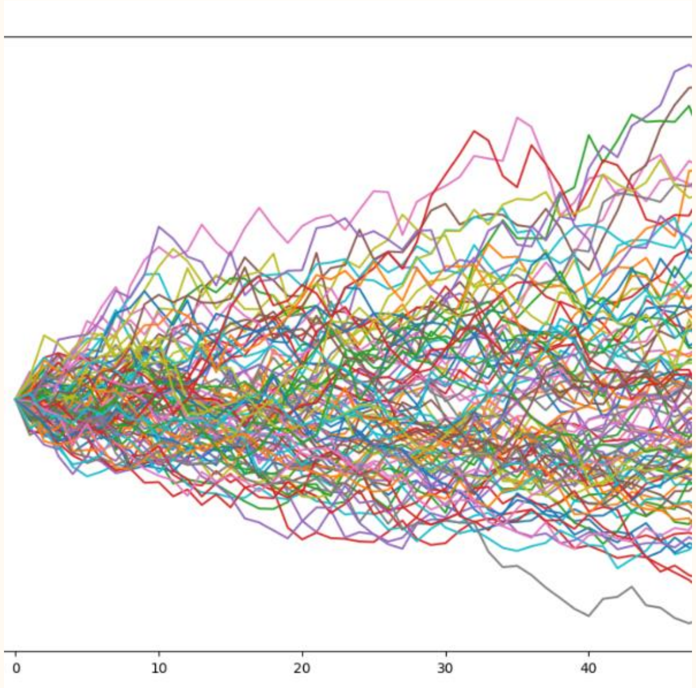
CVaR gives a more comprehensive view of risk by considering the tail end of the distribution of potential losses.

CVaR is particularly useful because it takes into account not just the threshold loss, but also the average of losses that occur beyond this threshold, providing a better understanding of extreme losses.

Confidence Level

The confidence level is the probability that the value of a portfolio will not exceed a specified level of loss over a defined period. It reflects the degree of certainty in the risk assessment. For example, a 95% confidence level indicates there is a 95% chance that the actual loss will not exceed the VaR.

Fact : The Var Models Failed to forsee the 2007-2008 crisis as the models can't capture the Extreme black swan moments which led to bankruptcy of Lehman Brothers and led the whole world to Recession.



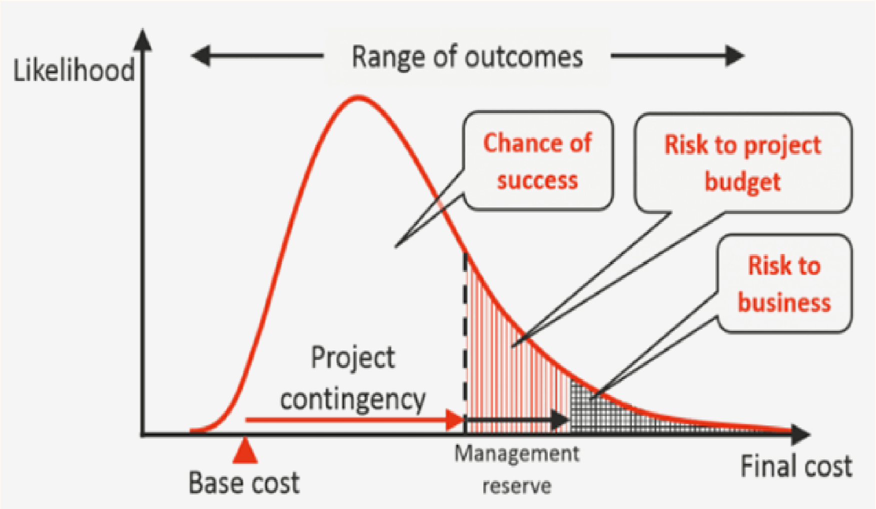
03

Monte Carlo

Prediction Using Random Sampling

Definition of Monte Carlo

Monte Carlo methods are computational algorithms that employ random sampling to simulate and analyse complex systems or processes, providing numerical solutions or estimates to problems that may be otherwise intractable or computationally expensive to solve using deterministic methods. These techniques are widely used across various fields to tackle uncertainty, optimize decision-making, and model real-world phenomena.



Datasets Taken

- Monthly Average Interest Dataset From 2001-2024 https://fiscaldata.treasury.gov/datasets/electronic-securities-transactions/sales

- Daily Interest Dataset From 1963-2024 from The Day the Got Introduced till The Day (Taking all the Extreme Events into Consideration) https://www.federalreserve.gov/datadownload/default.htm

- Daily Interest Dataset From 1981-2000 (Taking only Two Extreme Events) https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_bill_rates&field_tdr_date_value_month=202405

Basic Steps of Monte Carlo Simulation

1. **Define the Problem:** Identify the uncertain parameters and the outcome to be estimated.
2. **Generate Random Inputs:** Use random sampling to create a range of possible inputs based on historical data or assumed probability distributions.
3. **Run Simulations:** Perform a large number of simulations to model the process or system.
4. **Analyse Results**: Aggregate and analyse the outcomes to understand the distribution of possible results.

Process Explanation With Code

```
import time
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

num_bonds=1000
bond_face_value = 100
bond_years_to_maturity = 30
num_simulations = 10000
confidence_level = 0.95
```

Importing the necessary libraries

Declaration of Variables and some Information of Bonds

Process Explanation With Code

```
data_path = "DGS1962_2024.csv"
data = pd.read_csv(data_path)

# Remove rows with "." in the interest rate column
data = data[data['DGS10'] != "."]
interest_rates = data['DGS10'].astype(float)

#Setting bond coupon rate
mean_interest_rate = interest_rates.mean()
bond_coupon_rate = mean_interest_rate/100
```

- Importing data
- Cleaning of Data for some inconsistencies
- Calculating mean interest rate and for Real Life Calculation coupon rate is set to mean interest rate

Process Explanation With Code

```
# Calculate interest rate changes
interest_rate_changes = interest_rates.diff().dropna()


# Calculate mean and standard deviation of interest rate
changes
mean_rate_change = interest_rate_changes.mean()
std_dev_rate_change = interest_rate_changes.std()
```

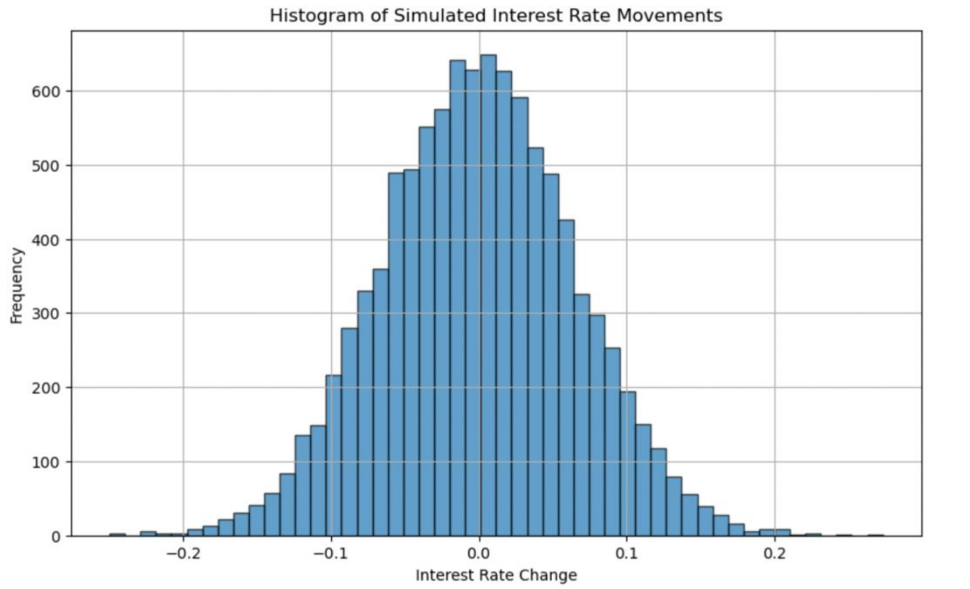
- Calculating the interest rate change from one day to the other
- Calcuating mean & standard deviation of the interest rate changes day by day

Process Explanation With Code

```
# Simulate future interest rate movements
interest_rate_simulations = np.random.normal(mean_rate_change, std_dev_rate_change, num_simulations)
```

The probability density for the Gaussian distribution is

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$



Process Explanation With Code

```
# Calculate bond price for each simulation
def calculate_bond_price(rate):
    discount_factor = 1 / (1 + rate) ** bond_years_to_maturity
    coupon_payment = bond_coupon_rate * bond_face_value
    bond_price = (coupon_payment * ((1 - discount_factor) / rate)) +
(bond_face_value * discount_factor)
    return bond_price
```

$$discount\ factor = \frac{1}{\left((1+rate)^{years}\right)}$$

$$bond\,price = \left(coupon\,payment \times \left(\frac{(1 - discount\ factor)}{rate}\right)\right) + (bond\ face\,value \times discount\ factor)$$

Process Explanation With Code

```
# Simulate investment values for each scenario
investment_values = []
for rate_change in interest_rate_simulations:
    rate = 1 + rate_change
    bond_price = calculate_bond_price(rate)
    investment_value = bond_price * num_bonds
    investment_values.append(investment_value)
```

For every interest rate we simulated we calculated. The bond price and multiplied with number of bonds to get the investment value and added it to the array of the investment values

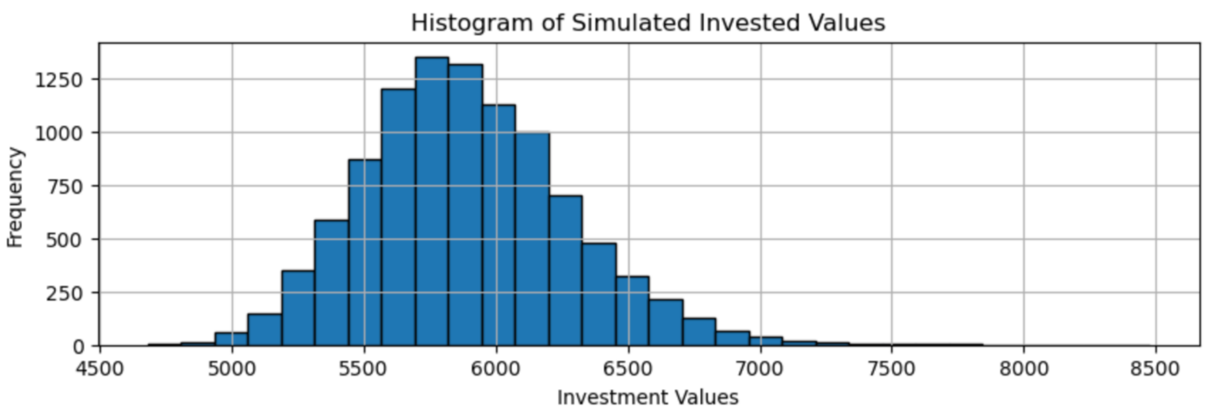
Process Explanation With Code

```
# Sort portfolio returns
sorted_returns = sorted([(v - (num_bonds*bond_face_value)) /
(num_bonds*bond_face_value) for v in investment_values])

# Calculate VaR
var_index = int(num_simulations * (1 - confidence_level))
var = -sorted_returns[var_index]
# Calculate CVaR
cvar = -np.mean(sorted_returns[:var_index])
```

Bond Maturity : 20 Years. 1962-2024

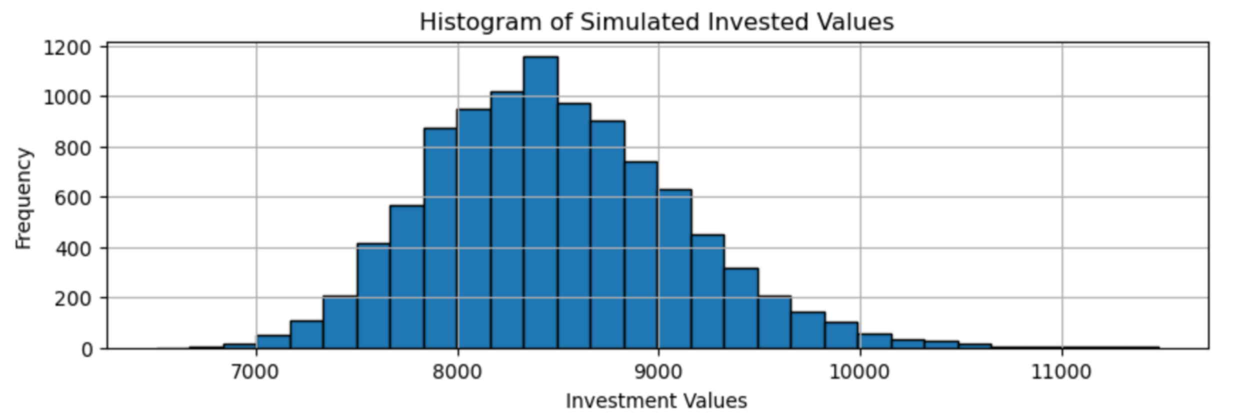
Bond Face Value = $100
Number of Bonds = 1000
Confidence Level = 95%
Simulations =1000




```
For a 95.0% Confidence Interval : VaR is 94.70%
For a 95.0% Confidence Interval : CVaR is 94.83%
```

Bond Maturity : 20 Years. 1981-2000

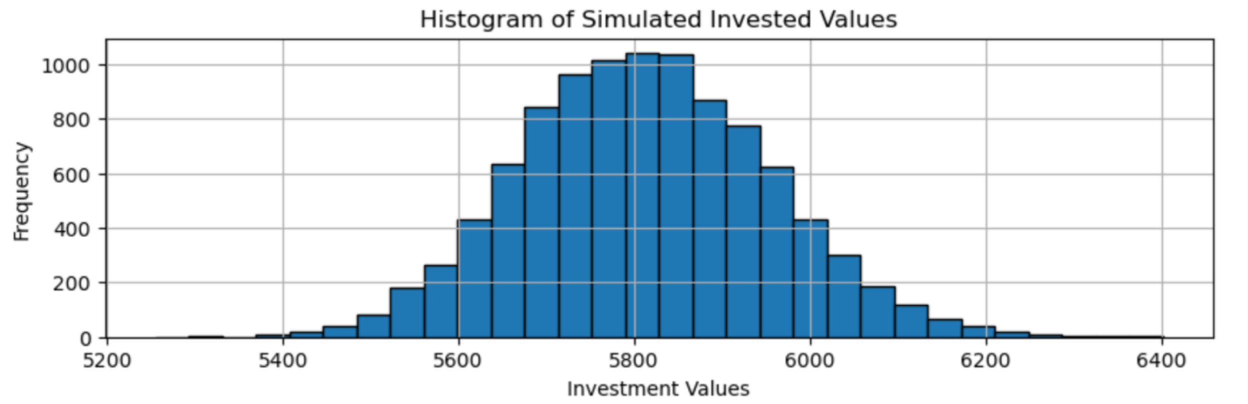
Bond Face Value = $100
Number of Bonds = 1000
Confidence Level = 95%
Simulations =1000




```
For a 95.0% Confidence Interval : VaR is 92.45%
For a 95.0% Confidence Interval : CVaR is 92.64%
```

Bond Maturity : 20 Years. MA-2000-2024

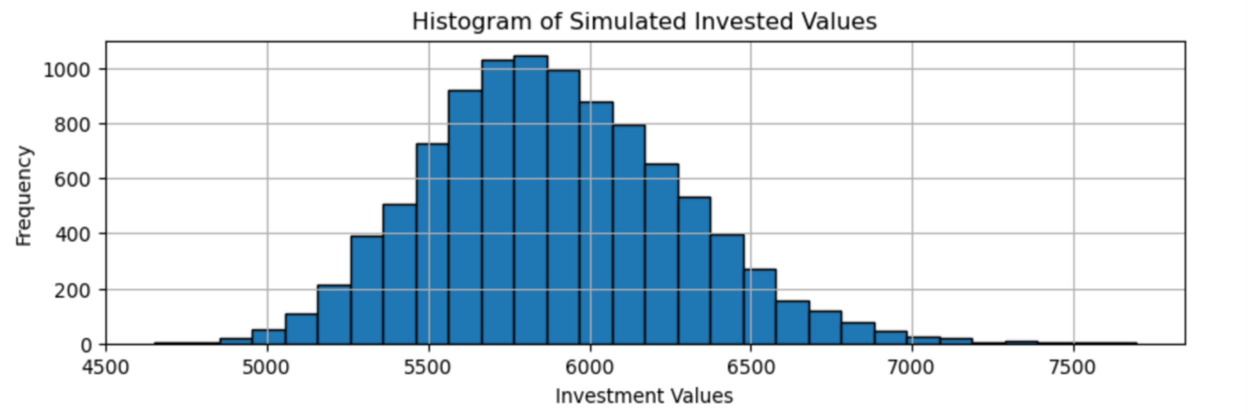
Bond Face Value = $100
Number of Bonds = 1000
Confidence Level = 95%
Simulations =1000




```
For a 95.0% Confidence Interval : VaR is 94.41%
For a 95.0% Confidence Interval : CVaR is 94.47%
```

Bond Maturity : 30 Years. 1962-2024

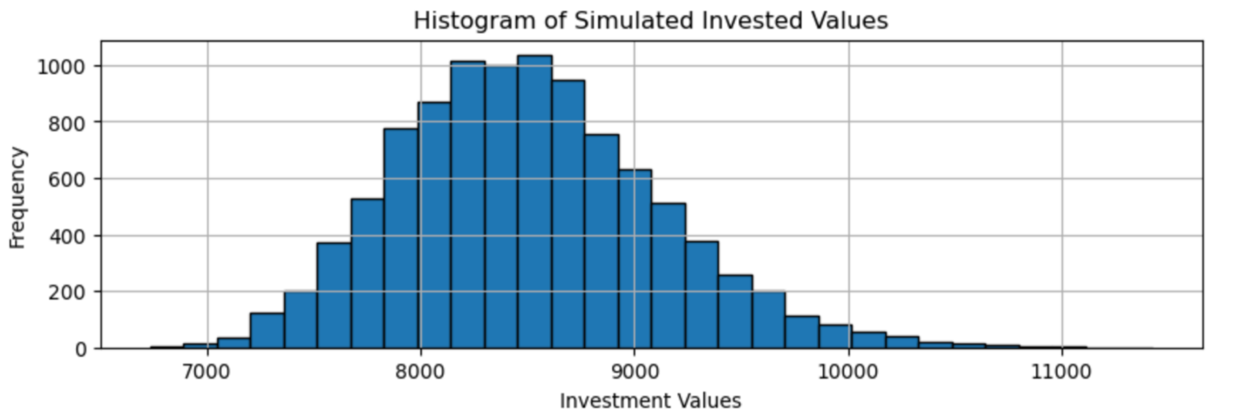
Bond Face Value = $100
Number of Bonds = 1000
Confidence Level = 95%
Simulations =1000




```
For a 95.0% Confidence Interval : VaR is 94.71%
For a 95.0% Confidence Interval : CVaR is 94.84%
```

Bond Maturity : 30 Years. 1981-2000

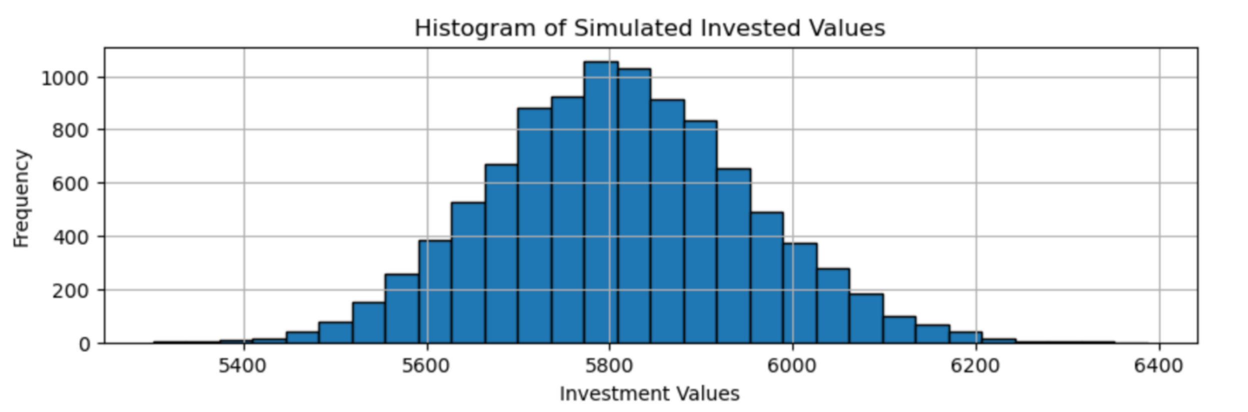
Bond Face Value = $100
Number of Bonds = 1000
Confidence Level = 95%
Simulations =1000




```
For a 95.0% Confidence Interval : VaR is 92.43%
For a 95.0% Confidence Interval : CVaR is 92.61%
```

Bond Maturity : 30 Years. MA-2000-2024

Bond Face Value = $100
Number of Bonds = 1000
Confidence Level = 95%
Simulations =1000




```
For a 95.0% Confidence Interval : VaR is 94.42%
For a 95.0% Confidence Interval : CVaR is 94.47%
```

Observations

Dataset	N	VaR	CVaR	Time
MA2001-2024	20	94.33%	94.39%	0.29
MA2001-2024	30	94.41%	94.47%	0.28
1962-2024	20	94.66%	94.79%	0.29
1962-2024	30	94.71%	94.83%	0.28
1981-2000	20	92.38%	92.58%	0.29
1981-2000	30	92.43%	92.63%	0.29

Drawbacks of this Method

1. Computational Cost:

- **Monte Carlo Simulation:** Requires a large number of simulations (often in the thousands or millions) to achieve accurate results. This leads to high computational costs and long processing times, especially when dealing with complex models or large datasets.

- **Quantum Amplitude Estimation:** QAE leverages quantum computing principles to achieve a quadratic speedup over classical Monte Carlo methods. This means it can reach the same level of accuracy with significantly fewer simulations, reducing computational costs and time.

2. Accuracy and Convergence:

- **Monte Carlo Simulation:** The accuracy of the results depends on the number of simulations. For rare events or extreme quantiles, a very high number of simulations may be needed to get reliable estimates, which can be impractical.

- **Quantum Amplitude Estimation:** Provides more precise estimates with fewer iterations, enhancing the accuracy of the simulation outcomes, particularly for rare events.

Drawbacks of this Method

3. Random Sampling Variability:

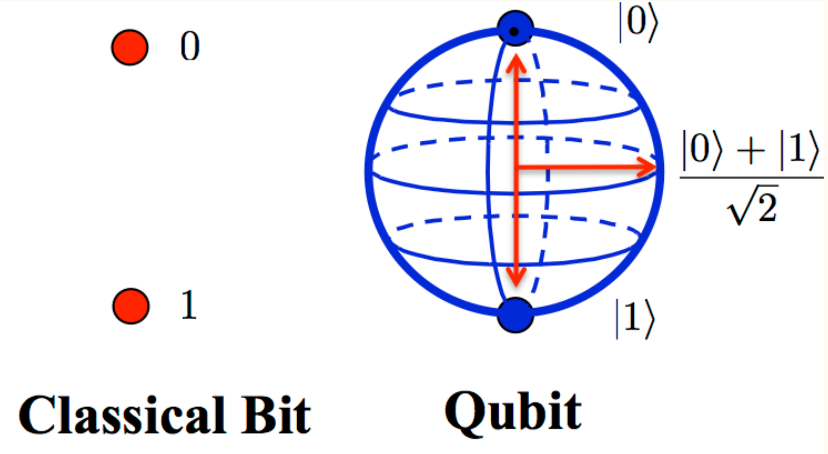
- **Monte Carlo Simulation:** Relies on random sampling, which introduces variability and potential biases in the results. Multiple runs can yield different outcomes due to this inherent randomness.

- **Quantum Amplitude Estimation:** While QAE also involves probabilistic elements, the quantum approach can reduce the variance in the estimates, providing more stable and reliable results.

4. Scalability Issues:

- **Monte Carlo Simulation:** As the complexity of the model increases, the number of simulations required grows exponentially, making it difficult to scale.

- **Quantum Amplitude Estimation:** The quantum approach can handle increased complexity more efficiently, offering better scalability for large and complex financial models.



03

Quantum Amplitude Estimation

Definition of Quantum Amplitude Estimation

Quantum Amplitude Estimation (QAE) is a quantum algorithm designed to estimate the probability amplitude of a specific quantum state. It leverages quantum principles such as superposition and interference to provide a significant speedup compared to classical algorithms for certain types of estimation problems.

Process

1. **Data Preprocessing:** Describe loading interest rate data from CSV file. Explain normalization of bond price changes and conversion of interest rates to bond prices.
2. **Quantum Circuit Construction:** Overview of constructing quantum circuits for A and Q operators.
3. **Estimation Problem Setup:** Explanation of setting up the estimation problem, including state preparation and Grover operator. Mention choice of parameters like target accuracy and confidence interval width.
4. **Iterative Amplitude Estimation (IAE):** Introduction to IAE as the chosen QAE variant. Description of parameters used for IAE and the role of the sampler.
5. **Quantum Risk Analysis Results:**

   Presentation of results:

   - VaR (Value at Risk) estimation.
   - CVaR (Conditional Value at Risk) estimation.

   Mention confidence level used for calculations (e.g., 95%).

Dataset

- Monthly dataset From 2001-2024
  https://fiscaldata.treasury.gov/datasets/electronic-securities-transactions/sales

- Daily Dataset From 1963-2024 from The Day the Got Introduced till The Day (Taking all the Extreme Events into Consideration)
  https://www.federalreserve.gov/datadownload/default.htm

- Daily Dataset From 1981-2000 (Taking only Two Extreme Events)
  https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_bill_rates&field_tdr_date_value_month=202405

Basic Explanation of Amplitude Estimation

1. **State Preparation (A):**
   1. This is the initial quantum state preparation, which sets up the quantum system.
   2. Typically involves an operator A that transforms the initial state $|0\rangle$ into a superposition state.
2. **Grover Operator (Q):**
   1. Used for amplitude amplification, it enhances the probability of measuring the desired state.
   2. The Grover operator Q is applied iteratively to increase the amplitude of the target state.
3. **Measurement:**
   1. The objective is to measure the qubits and identify the probability of the "good" state (often denoted as $|\Psi 1\rangle$)

Construction of Quantum Circuit

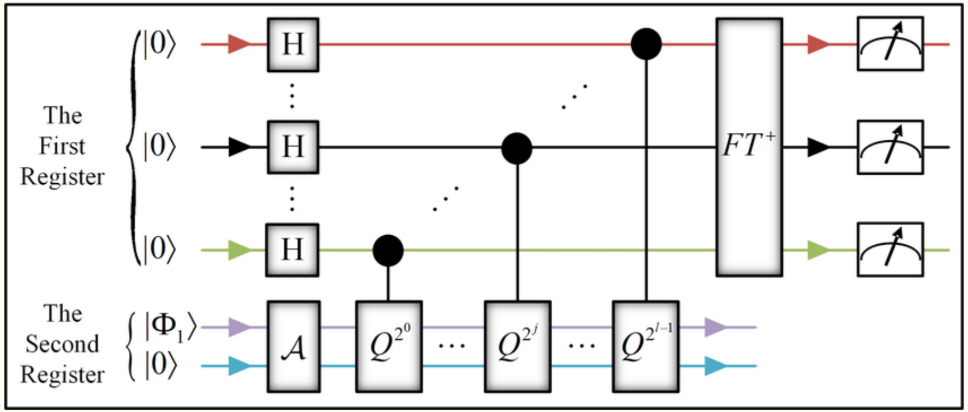
A Operator(**State Preparation**)

$$\mathcal{A}|0\rangle = \sqrt{1-p}|0\rangle + \sqrt{p}|1\rangle.$$

$$\mathcal{A} = R_Y(\theta_p), \theta_p = 2\sin^{-1}(\sqrt{p}).$$

Q Operator (**Grover Operator**)

$$\mathcal{Q} = R_Y(2\theta_p),$$



Estimation Problem

An **Estimation Problem** in the context of Amplitude Estimation (AE) is a structured way to define the quantum system and the parameters needed for estimating the amplitude of a desired state.

Benefits of Using Estimation Problems:

- **Modularity:** Allows clear separation of the problem definition and the algorithm.
- **Flexibility:** Can be adapted to different quantum algorithms and objectives.
- **Reusability:** The same problem instance can be used with various AE algorithms.

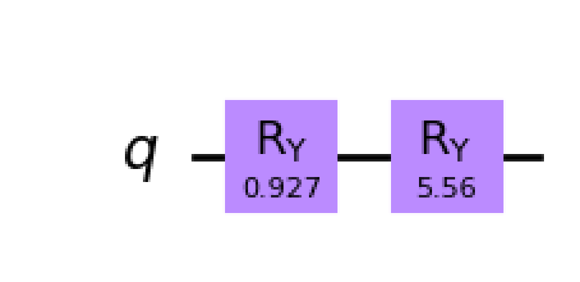
Five Types of Quantum Amplitude Estimation

- Canonical AE
- **Iterative Amplitude Estimation**
- Maximum Likelihood Amplitude Estimation
- Faster Amplitude Estimation

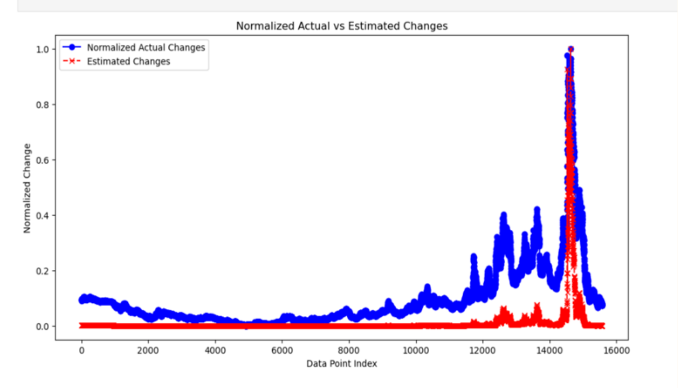
Iterative Amplitude Estimation

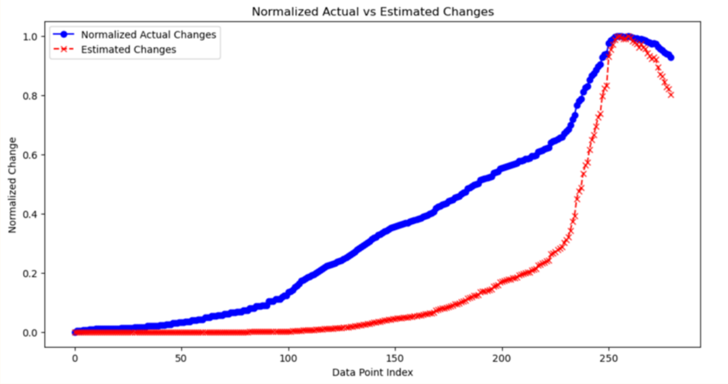
IAE is a quantum algorithm designed to estimate the probability (amplitude) of a particular outcome (state) in a quantum system with high accuracy.

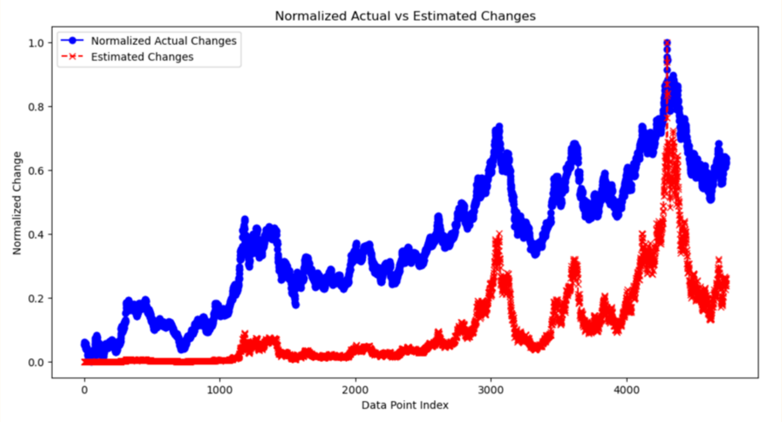
IAE refines the estimation in steps, using feedback from previous iterations to improve the accuracy of the result.



Iterative Amplitude Estimation

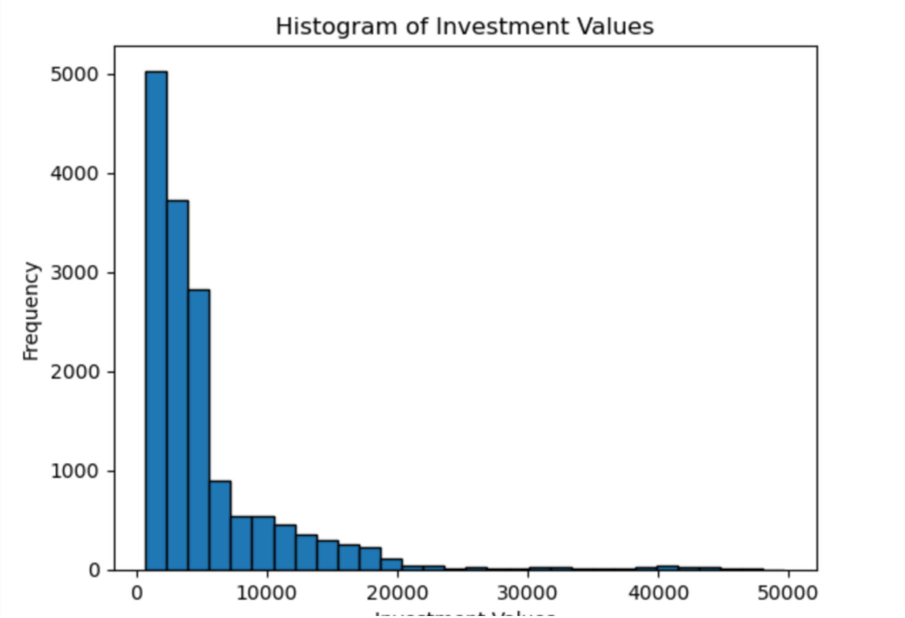






Bond Maturity : 30 Years. 1962-2024

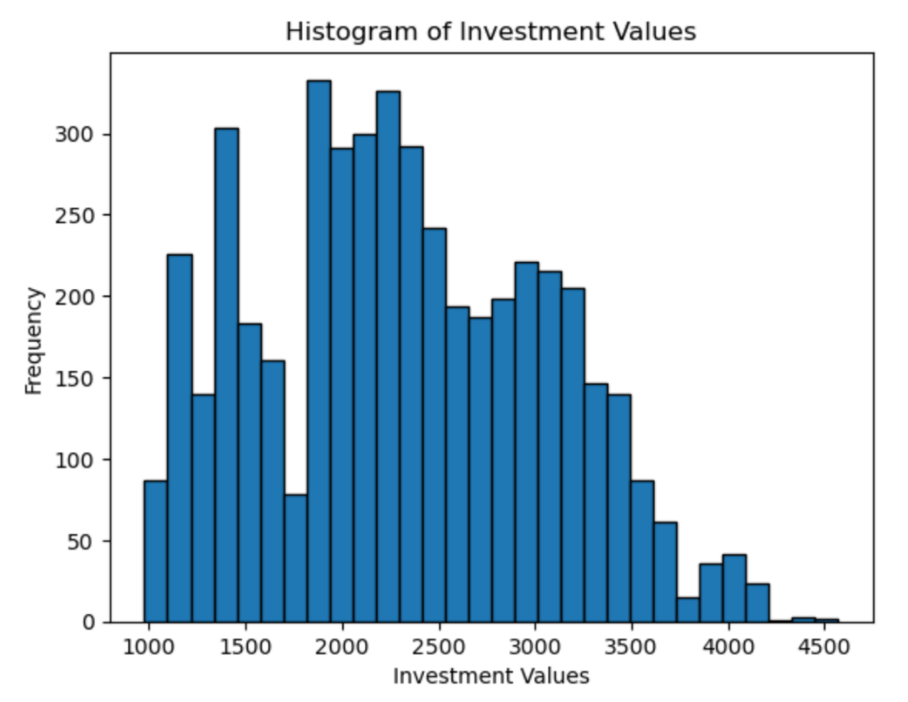
Bond Face Value = $100
Number of Bonds = 1000
Confidence Level = 95%
Simulations =1000




```
For a 95.0% Confidence Interval : VaR is 99.27860%
For a 95.0% Confidence Interval : CVaR is 99.27861%
```

Bond Maturity : 30 Years. 1981-2000

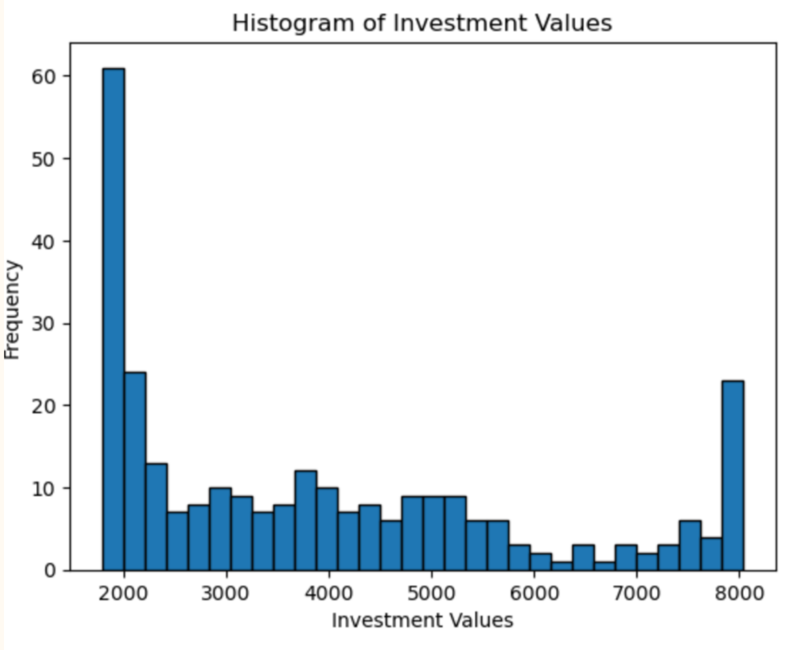
Bond Face Value = $100
Number of Bonds = 1000
Confidence Level = 95%
Simulations =1000




```
For a 95.0% Confidence Interval : VaR is 99.02292%
For a 95.0% Confidence Interval : CVaR is 99.02330%
```

Bond Maturity : 30 Years. MA-2000-2024

Bond Face Value = $100
Number of Bonds = 1000
Confidence Level = 95%
Simulations =1000




```
For a 95.0% Confidence Interval : VaR is 99.32568%
For a 95.0% Confidence Interval : CVaR is 99.32568%
```

Observations:

Dataset	N	VaR	CVaR	Time
MA2001-2024	20	99.32%	99.32%	0.01
MA2001-2024	30	99.32%	99.32%	0.01
1962-2024	20	99.28%	99.28%	0.01
1962-2024	30	99.28%	99.28%	0.01
1981-2000	20	99.43%	99.43%	0.04
1981-2000	30	99.43%	99.43%	0.01

Drawbacks of this method

Complexity of Quantum Circuits: Implementing QAE involves designing and optimizing complex quantum circuits for state preparation and Grover's operator. The growing depth and width of these circuits with problem size increase error rates, presenting a challenge given current quantum computing capabilities.

Algorithmic Assumptions: The effectiveness of QAE depends on the accurate modeling and encoding of bond price changes into quantum states. Deviations from these assumptions can lead to incorrect amplitude estimations, resulting in unreliable financial risk metrics.

Monte Carlo vs Quantum Amplitude Estimation

Monte Carlo Simulation:

Sampling Methods: Monte Carlo relies on random sampling methods such as the Monte Carlo integration or Markov Chain Monte Carlo (MCMC) to estimate quantities of interest.

Classical Execution: It operates on classical computers, making it accessible and applicable to a wide range of industries and problem domains.

Accuracy vs. Computational Cost: The accuracy of Monte Carlo simulations can be improved by increasing the number of samples, but this often comes at the expense of computational cost.

Quantum Amplitude Estimation (QAE):

Quantum Superposition: QAE utilizes quantum superposition to represent and process multiple states simultaneously, potentially leading to more efficient computation for certain estimation problems.

Quantum Circuit Design: QAE involves designing quantum circuits to encode and manipulate information about probability amplitudes, requiring expertise in quantum algorithms and circuit optimization.

Challenges and Opportunities: While QAE offers the prospect of exponential speedup over classical methods, challenges such as quantum hardware limitations and algorithmic complexity need to be addressed for practical implementation.

Future Scope

Implementing on other Financial Instruments:

Applying this integrated approach to a wider range of financial instruments beyond Treasury bonds, including corporate bonds, equities, and derivatives.

- **Cross-Market Analysis:** Conducting cross-market analysis to understand the ripple effects of sentiment-driven changes in one market on others.
- **Customized Solutions:** Developing customized risk assessment tools for different types of investors, such as institutional investors, hedge funds, and retail investors, tailored to their specific needs and strategies.

Real-Time Data Processing:

Implementing systems for real-time data processing and analysis. This involves:

- **Stream Processing:** Setting up infrastructure to process streaming data from news feeds and financial markets in real time.
- **Continuous Updates:** Continuously updating risk models with the latest sentiment scores and market data to maintain accuracy and relevance.

Future Scope

Integration of Sentiment Analysis:

Incorporate sentiment analysis of financial news to gauge market sentiment. This involves:

- **Natural Language Processing (NLP):** Utilizing advanced NLP techniques to process and analyze news articles, reports, and social media feeds.
- **Sentiment Scoring:** Developing algorithms to assign sentiment scores (positive, negative, neutral) to news items related to economic indicators, interest rates, and geopolitical events.

Impact on Bond Prices: Establishing models to correlate sentiment scores with bond price movements. By understanding how news sentiment affects market perception, we can improve the accuracy of risk assessments.

Conclusion

Higher Accuracy in Risk Assessment: Quantum Amplitude Estimation (QAE) demonstrated more conservative and consistent Value at Risk (VaR) and Conditional Value at Risk (CVaR) values compared to Monte Carlo Simulation (MCS), indicating a potential for higher accuracy in financial risk assessments.

Significant Computational Efficiency: QAE outperformed MCS in terms of computational speed, processing risk metrics in approximately 0.01 seconds compared to MCS's 0.28-0.29 seconds, highlighting QAE's efficiency for real-time risk analysis.

Future Potential: The integration of quantum computing in financial risk management offers transformative potential, enhancing the precision and speed of risk assessments and paving the way for its broader adoption in the industry. Further exploration and analysis are warranted to fully realize these benefits.

Time Line of Activities

- Reading and Understanding Research Paper
- Writing Pseudo Code
- Scraping Website for Dataset
- Calculating using Code of Monte Carlo
- Calculating using Quantum Amplitude Estimation
- Analyzing Results
- Preparing Project report/ Presentation
- Presentation Day (Today)

References

- https://qiskit-community.github.io/qiskit-finance/tutorials/00_amplitude_estimation.html
- https://www.nature.com/articles/s41534-019-0130-6
- https://fiscaldata.treasury.gov/datasets/qtcb-historical-interest-rates/historical-qualified-tax-credit-bond-interest-rates
- https://www.youtube.com/@qiskit/featured

Thanks!
Q&A

Matta Rithik Reddy
SE20UECM045
www.rithikreddy.com

Contact Info: +91 7032536194 mattarithikreddy@gmail.com