Economic Capital Requirement Estimation using Quantum Computing

March 2023

Team ZuriQ

Hector Blondel, Aymane Maaitat, Arman Saulière, Noé Bosc-Haddad, Etienne Lefranc

CRÉDIT AGRICOLE
CORPORATE & INVESTMENT BANK

Achraf SEDDIK, Ali el Hamidi

I – Use case: Classical solution

$$L = \sum_{i=1}^{M} EAD_i . LGD_i . D_i$$

Loan Amount

Indicator Function of Client's default to repay

Percentage of Loan amount lost

I – Use case: Classical solution

$$L = \sum_{i=1}^{M} EAD_i . LGD_i . D_i$$

$$P(D_i = 1) = P\left(X_i < N^{-1}[PD_i]\right)$$

I – Use case: Classical solution

$$L = \sum_{i=1}^{M} EAD_i . LGD_i . D_i$$

$$P(D_i = 1) = P(X_i < N^{-1}[PD_i])$$

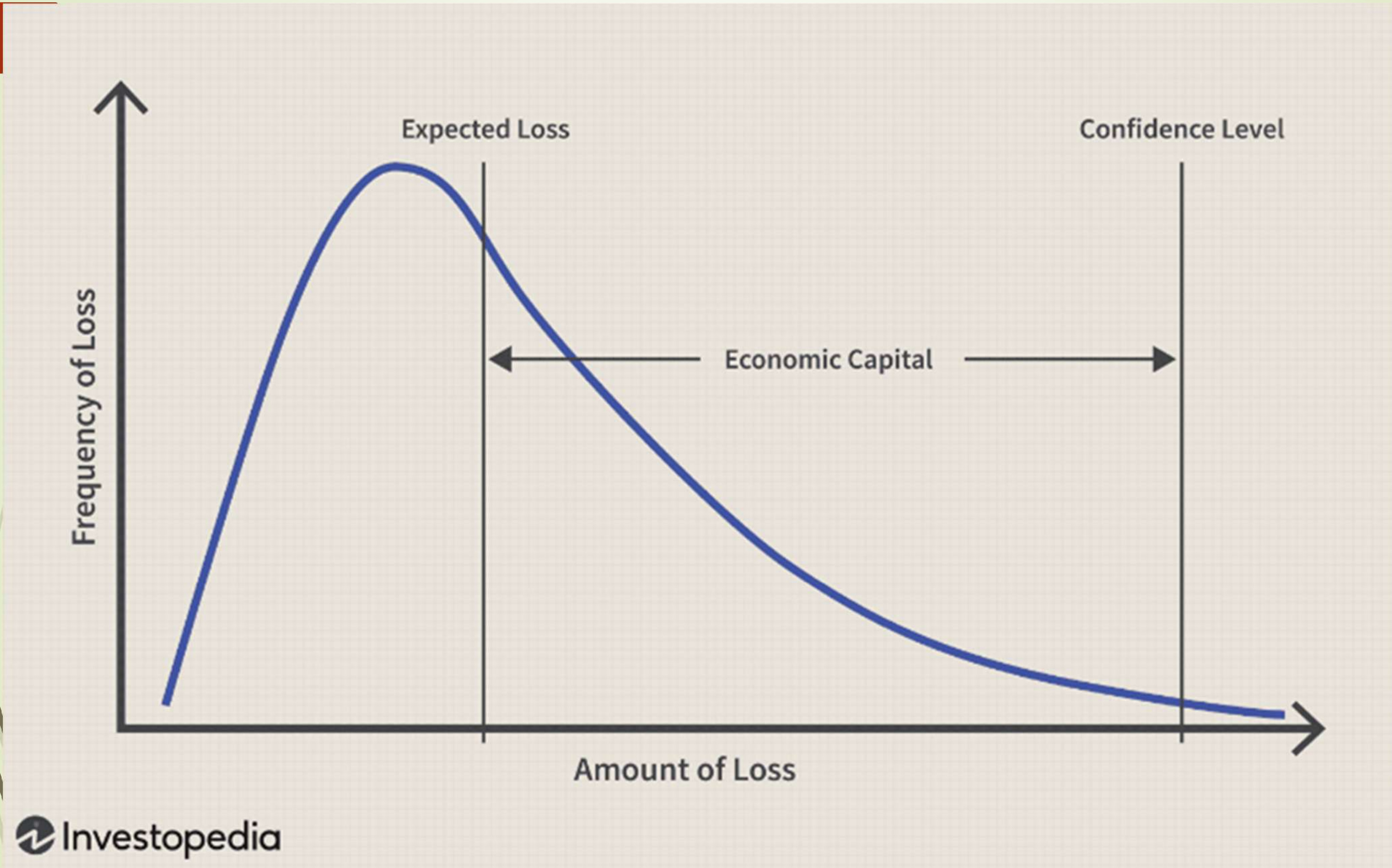
N(0,1)

N(0,1)

N(0,1)

$$X_i = r_i . Y_i + \sqrt{1 - r_i^2} . \varepsilon_i$$

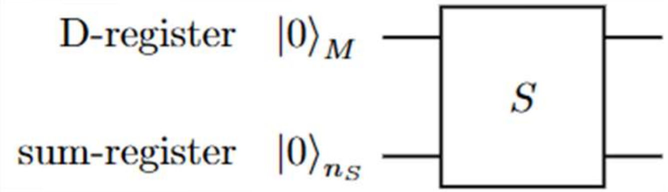
I – Use case: Classical solution



II - Quantum Algorithm Based on Quantum Amplitude Estimation (QAE)

1. Principle of the Algorithm:

First, we start by computing the total loss into a quantum qubit register.



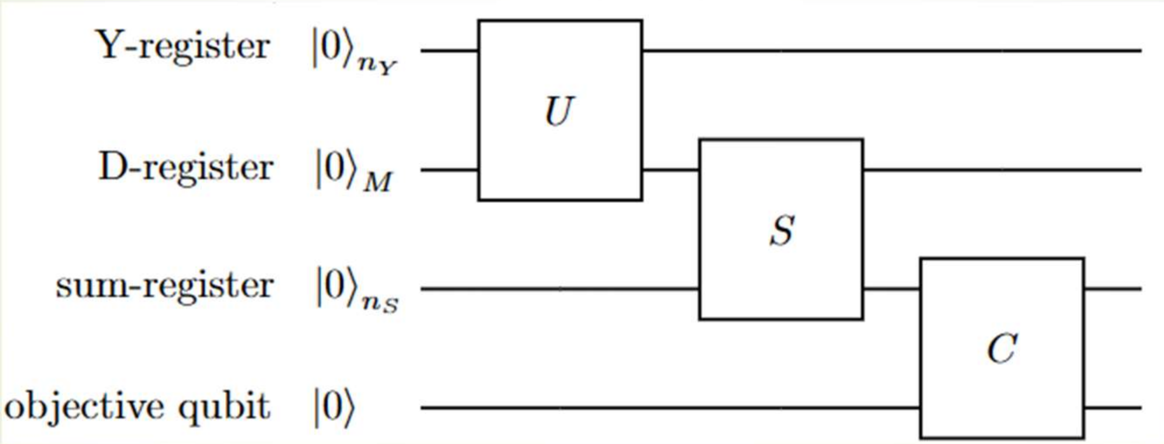
$$S : |D_1, D_2, ..., D_M\rangle|0\rangle_{n_s} \mapsto |D_1, D_2, ..., D_M\rangle|LGD_1 EAD_1 D_1 + .. + LGD_M EAD_M\rangle$$

Second, we estimate the CDF (cumulative distribution function) using QAE.

Using the CDF estimation, we can directly access the VaR_α.

II - Quantum Algorithm Based on Quantum Amplitude Estimation (QAE)

2. Quantum Circuit:

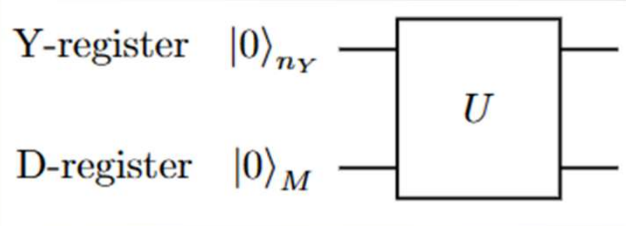


II - Quantum Algorithm Based on Quantum Amplitude Estimation (QAE)

2. Quantum Circuit:

- 1.First part: Computing the D_i distributions

We begin by generating a normal distribution on the Y register.



We then use it to generate the wave function:

$$|\Psi\rangle = \sum_{i=0}^{2^{n_y}-1} \sqrt{p_y^i}|y_i\rangle \bigotimes_{k=1}^{M} \left(\sqrt{1-p_k(y_i)}|0\rangle + \sqrt{p_k(y_i)}|1\rangle\right)$$

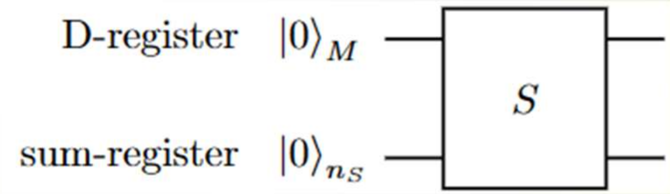
II - Quantum Algorithm Based on Quantum Amplitude Estimation (QAE)

2. Quantum Circuit:

- 2. Casting the sum into the S register and mapping the problem:

$$\mathrm{S} : |D_1, ..., D_M\rangle_M |0\rangle_{n_s} \mapsto |D_1, ..., D_M\rangle_M |EAG_1 LGD_1 D_1 + ... + EAG_M LGD_M D_M\rangle_{n_s}$$

S computes the weighted sum on every qubit in order to obtain a superposition of every loss values



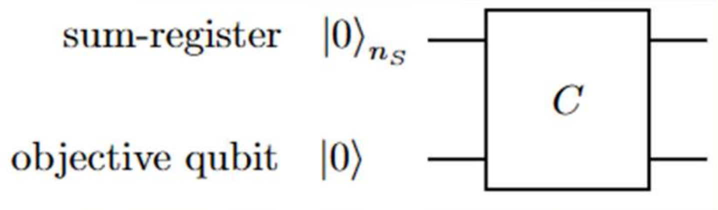
$$\sum_{L=0}^{2^{n_s}-1} \sqrt{p_L}\, |...\rangle_{nZ}\, |...\rangle_M\, |L\rangle_{n_s}\, |0\rangle$$

II - Quantum Algorithm Based on Quantum Amplitude Estimation (QAE)

2. Quantum Circuit:

- 3. Compare loss values with x and change the objective qubit

$$C \ : \ |L\rangle_{n_s} |0\rangle \mapsto \begin{cases} |L\rangle_{n_s} |1\rangle & \text{if L} \leq \text{x} \\ |L\rangle_{n_s} |0\rangle & \text{otherwise} \end{cases}$$



$$\sum_{L\leq x} \sqrt{p_L}\, |...\rangle_{n_Z} |...\rangle_M |L\rangle_{n_s} |1\rangle + \sum_{L\geq x+1} \sqrt{p_L}\, |...\rangle_{n_Z} |...\rangle_M |L\rangle_{n_s} |0\rangle$$

II - Quantum Algorithm Based on Quantum Amplitude Estimation (QAE)

3. QAE and Complexity

We then apply QAE to obtain the probability.

Using QAE, we need $O(1/\varepsilon)$ iterations/quantum iterations to get an ε precision.

Quantum amplitude amplification and estimation
G Brassard, P Hoyer, M Mosca... - Contemporary ..., 2002 - books.google.com

II - Quantum Algorithm Based on Quantum Amplitude Estimation (QAE)

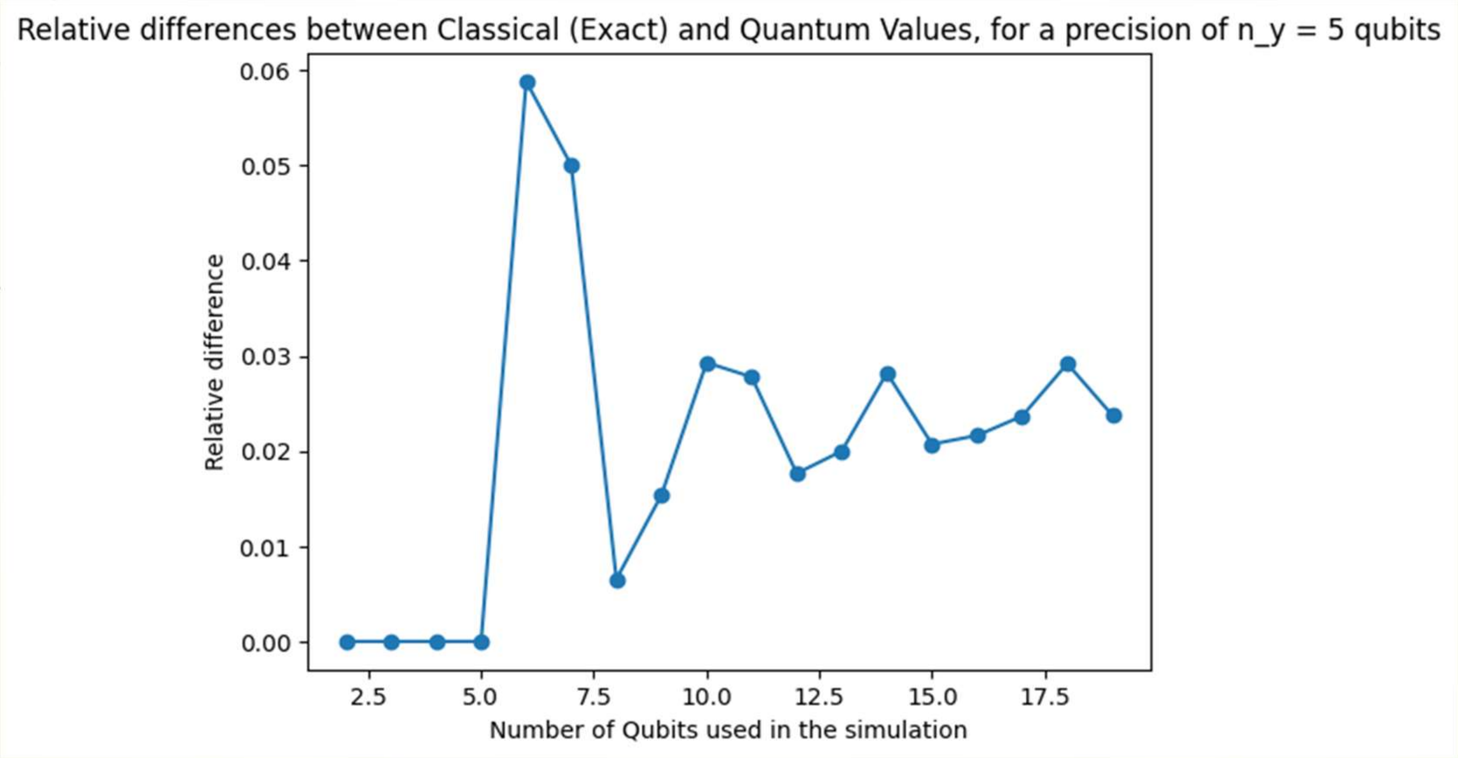
4. Potential Quantum Advantage:

Existing classical methods rely on Monte-Carlo simulations to obtain the VaR, these however require $O(1/\varepsilon^2)$ iterations/quantum iterations to get an ε precision.

Thus, the quantum advantage in this solution is quadratic and is a consequence of QAE's quantum advantage.

II - Quantum Algorithm Based on Quantum Amplitude Estimation (QAE)

5. Classical Benchmarking:

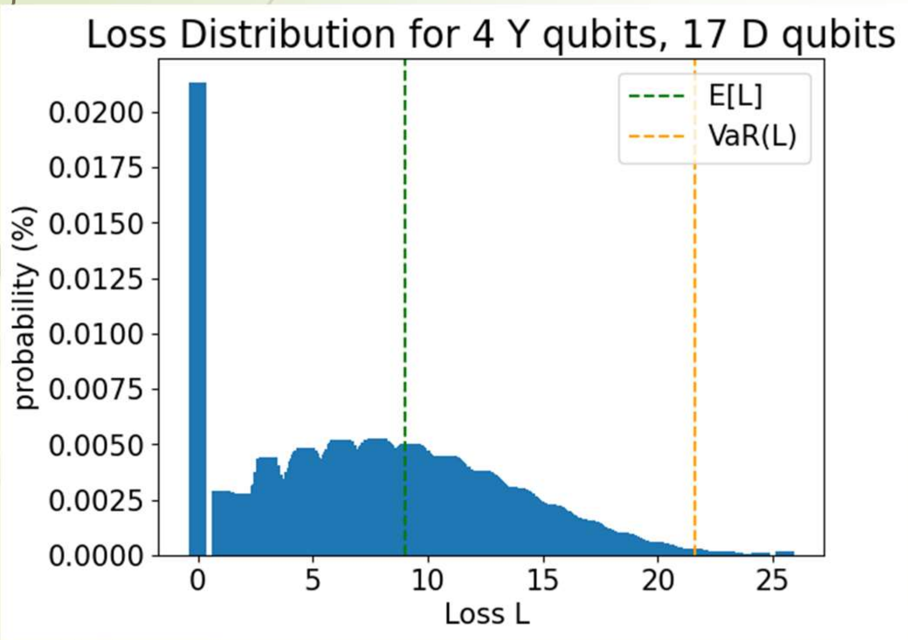


We compare the results obtained for a classical algorithm and a quantum implementation of the algorithm based on QAE

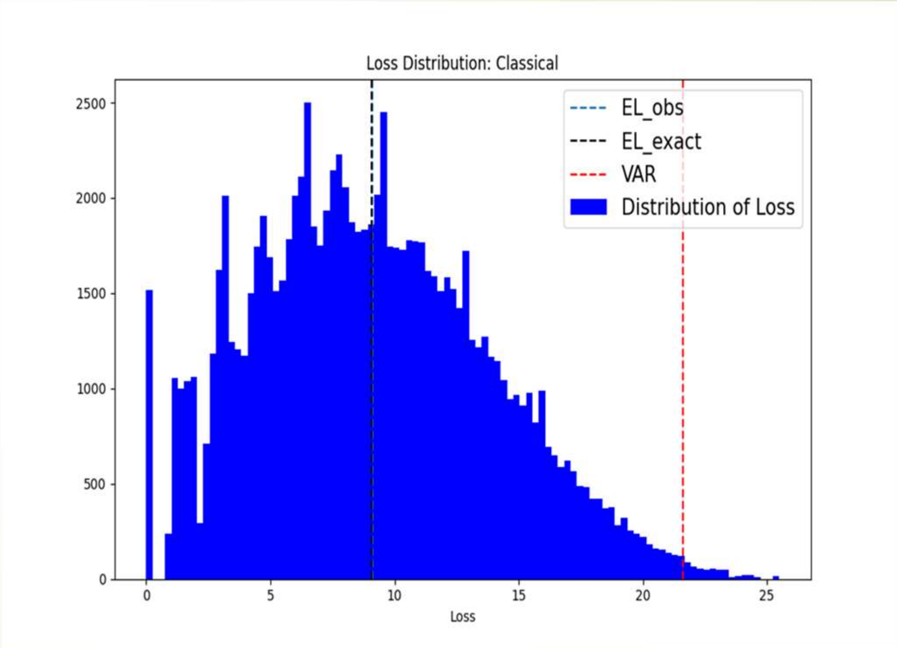
The values are to be found in the appendix

II - Quantum Algorithm Based on Quantum Amplitude Estimation (QAE)

5. Classical Benchmarking:



Loss distribution obtained using the quantum algorithm



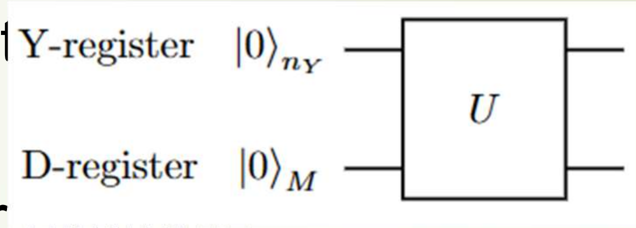
Loss distribution obtained using the classical algorithm (using the same parameters)

II - Quantum Algorithm Based on Quantum Amplitude Estimation (QAE)

6. Limitations of the algorithm:

I. Limitations of QAE

II. The precision depends on that of the Y-regist
→ additional burden on the number of qubits



III. The first part USC requires multiple gates, for example
The U part scales on O(M) (on T gates), an efficient implementation
in Egger et Al 2019 proposes a factor of 28.

Credit risk analysis using quantum computers
DJ Egger, RG Gutiérrez, JC Mestre... - IEEE Transactions on ..., 2020 - ieeexplore.ieee.org

III -Value at Risk using Extreme Value Theory, Maximum Likelihood Estimator and QUBO optimization

- We approximate the likelihood into a Qubo problem

$$f_{\sigma,\xi}(x) = \frac{1}{\sigma} H_{\sigma,\xi}(x)^{\xi+1} e^{-H_{\sigma,\xi}(x)}$$

$$H_{\sigma,\xi}(x) = 1 + \xi\left(\frac{x - \mathbb{E}_L + \sigma\frac{g_1 - 1}{\xi}}{\sigma}\right)^{-\frac{1}{\xi}}$$

$$\ell(\theta, \phi | x_d) \approx \sum_i a_i q_i + \sum_{j>i} b_{ij} q_i q_j$$

III -Value at Risk using Extreme Value Theory, Maximum Likelihood Estimator and QUBO optimization

$$\ell(\theta,\phi|x_d) \approx \sum_i a_i q_i + \sum_{j>i} b_{ij} q_i q_j$$

We solve for q using Qubo solver

$$\xi = \sum_{i_\xi=1}^{M} q_{i_\xi} 2^{-i}$$

$$\sigma = \sum_{j_\sigma=M+1}^{M+N} q_{j_\sigma} 2^{j_\sigma-(M+1)+9}$$

We compute the value at risk

$$VaR_{0.999} = [(-ln(0.999))^{-\xi} - 1]^{\frac{\sigma}{\xi}} + \mathbb{E}_L - \sigma\frac{(ln2)^{-\xi}-1}{\xi}$$

IV -Value at Risk using Extreme Value Theory, Maximum Likelihood Estimator and QUBO optimization

```
values_L_i, values_L = Calcul_L(nb_simulation,M1,r1,N_1_PD1,EAD1,LGD1)

Q_matrix =  qubo_representation(values_L)

print(Q_matrix)

bqm = dimod.BinaryQuadraticModel.from_qubo(Q_matrix, offset=0.0)
sampler = dimod.SimulatedAnnealingSampler()
response = sampler.sample(bqm, num_reads=100)
optimum = response.first.sample
print(optimum)

res_xi = xi_value(list(optimum.values())[:15])
res_sigma = sigma_value(list(optimum.values())[15:])
```

We obtain the following results:

xi : 0.999969482421875
sigma : 2462208

V – Possible extensions and improvements

1. Using Quantum Machine Learning to estimate VAR (example using QCNN)
2. Bayesian Estimation
3. Casting the problem directly as a QUBO
4. Using Grover's algorithm

Due to a lack of time, we were not able to completely explore these solutions.

Appendix

Values used for benchmarking

```
# set problem parameters
n_y = 4
y_max = 6
N = 17
y_values = np.linspace(-y_max, y_max, 2**n_y)
K = len(p_zeros)
p_zeros = np.linspace(0.3, 0.4, N)
risq = np.linspace(0.1, 0.2, N)
lgd = np.linspace(1, 2, N)
alpha = 0.005
```

✓ 0.3s