Option price modelling using the Black-Scholes model and the VarQITE quantum algorithm.

Felix Xu
Pau Torrente Badia
Adrià Blanco Culla

1st of April 2023

Abstract

In this project the main objective is to study the behaviour and evolution of option prices following the Blach-Scholes model by using a variational quantum imaginary time-evolution method, or VarQITE in short. While this financial model is quite simple, the predictions in derivative pricing that come out of it are still relevant. On the other hand, the partial differential equation at its core can be easily mapped into a quantum mechanics problem, allowing it to be translated into something suitable to be treated by quantum computers,

1 Brief Introduction

Call options are a kind of finantial derivative that give the holder the right, but not the obligation, to buy an underlying asset at a certain time in the future and at a fixed price. Though there are many kinds of options, in this work we focus on the so called European option, which can only be executed on the expiring date.
The Black-Scholes model is an option pricing model developed in the 1970s that allows for the prediction of the fair price of an option before its execution. The main two assumptions that this model makes are:

- The market is efficient: Information spreads throughout the whole market instantaneously and, hence, the prices reflect that information at all times.
- Arbitrage-free market : there is no possible riskless profit.

And at the end of the day, this ends up translating into the assumption that the underlying asset prices follow brownian geometric random walks.

The stocastic differential equation at the core of the theory is the following:

$$\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS\frac{\partial C}{\partial S} = rC \tag{1}$$

And the boundary conditions:

$$C(S=0,t) = 0, C(S_{max},t) = S_{max} - K\exp\{-r(T-t)\} \quad 0 \leq t \leq T \tag{2}$$

$$C(S,t=T) = max(S-K,0) \quad 0 \leq S \leq S_{max} \tag{3}$$

Where the different variables at play are the following:

- C: Call option price, our variable of interest.
- S: Asset price. The stochastic independent variable.
- K: Strike price. The purchase price of the underlying asset that the contract specifies. We have used K=100 throughout the project.

- r: Risk-free interest rate. We assume it is a constant between 0 and 1. In our case, we take $r = 0$
- σ: Volatility of the stocks return. Also a parameter between 0 and 1 that we assume to be constant and equal to 0.4.
- T: Time to option maturity (when the contract expires) in years. We take $T = 1$.

From all of this, a key point to understand and keep in mind at all times is that the boundary condition at the time of maturity is used as an initial condition to propagate the solution backwards in time, since our objective is to predict the right price today knowing how will the return look like at $t = T$ in the future.

2 Translation of the Black-Scholes equation into a quantum computing problem

To treat the problem with a quantum approach the first step is to encode the values of C as a function of S in the computational basis of our quantum computer in a discrete form. To do so, we suppose that our desired state can be constructed in the following form:

$$|\phi(S,t)\rangle = V(S,t)\,|0\rangle \tag{4}$$

The problem therefore is translated into knowing how this operator evolves. Through some transformations:

$$S = e^x \tag{5}$$
$$\tau = \sigma^2(T-t) \tag{6}$$
$$V(\tau,x) = e^{ax+b\tau}u(\tau,x) \tag{7}$$

the BS equation for the state construction operator becomes the following:

$$-i\partial_\xi u(\xi,x) = \frac{1}{2}\partial_x^2 u(\xi,x) \tag{8}$$

Where now ξ, x and u play the role of t, S and V respectively. Our equation then is just a Schrödinger equation for a plane wave, where the hamiltonian in our discrete x space looks like:

$$H = \frac{1}{2\Delta_x^2}\begin{pmatrix} -2b\Delta_x^2 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -2b\Delta_x^2 \end{pmatrix} \tag{9}$$

Where the parameter b encodes the boundary conditions of our problem (this can be derived by analyzing the time derivative of $C(S_{max},t)$ and $C(S_{min},t)$ using that $r << 1$, which leads to $u' = -bu$ and then $H = -b$.

From this point on, one can try to evolve in time by applying the unitary evolution operator through Trotter decompositon, for example, or to approximate the solution by applying a variational principle. The latter is the one that this work has been based on, using the VarQITE algorithm, which initializes the states trough a parametrized circuit and updates the parameters at each timestep by solving a set of linear equations on the time derivatives of the parameters.

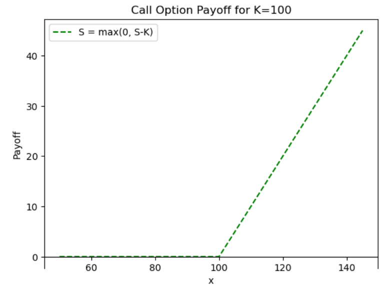

Figure 1: Payoff graph at the maturity price as a function of the asset price.

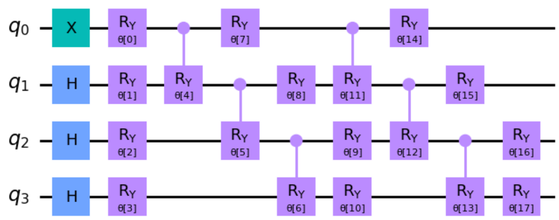

Figure 2: Parametrized quantum circuit used to initialize the quantum state

3 Initial state preparation

A we have said, the function we want to model has to be initiated at maturity, when the returns are known and follow:
To initialize our state to mimic this function we have used the parametrized circuit shown in **??**, on top of which we have optimized an l1-norm cost function that evaluates the difference between the coefficients with which the state leaves the circuit and the ones that it should have to replicate the function shown in 1. From the minimization of this cost function, taking 'X = np.linspace(50,150,16)', the values of the parameters and the resulting initial state are shown in 3 and 4.

```
array([-1.37826427, -0.05099204,  0.57874074,  0.3682538 , -1.17277047,
        0.0474472 , -0.48152146,  0.21603564, -1.27985454,  0.44746483,
        0.12984739, -1.85275176, -0.20815855,  0.32391571,  1.12250843,
        0.02967961, -0.20241489,  1.15836057])
```

Figure 3: Optimal parameter vector (one component for each parametrized gate) to replicate the desired initial state.

4 Variational approximation of the state

A we have already said in the introduction, once we have an initial set of optimal parameters we can propagate the evolution through solving a set of linear equation for $\dot{\theta}$. Instead of doing all of this manually, we have resorted to the qisktit.algorithms class "VarQITE", which takes in the hamiltonian in the form of a factorized opflow operator, a specification of the problem in terms of a "TimeEvolutionProblem" object, and the variational principle to be followed, which in our case is an imaginary time one. Once the object has been created, it can be called to evolve the parametrized circuit so that it can now "create" the desired quantum state encoding the objective fair price of the option we want to model. The final parameters and the distribution of returns compared to the initial one are shown in figures 5 and 6. Looking at these figures we can see that the values of the functions didn't change much, probably because of the small value of maturity time $T = 1$.

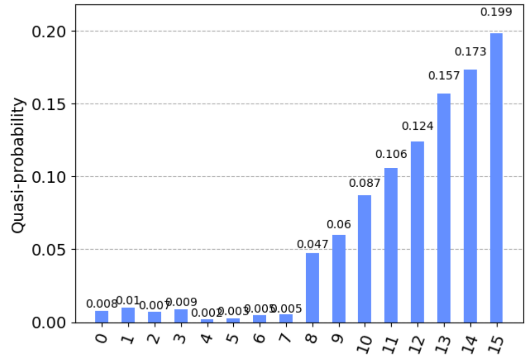

Figure 4: Resulting state from the optimized parametrized circuit.

```
array([-0.31742589,  0.06662123,  0.72621177,  1.17814289, -0.62795781,
        0.54255213, -0.01232603,  0.68475468,  0.319338  ,  0.21762686,
        1.2884094 , -0.2308394 , -0.76798694,  0.34621899, -1.83493719,
        0.72685287, -0.18076942, -1.22423315])
```

Figure 5: Final parameters for the circuit, obtained through the variational process.

5 Multi-dimensional Black-Scholes model

The multi-dimensional Black-Scholes model is essentially an extension of the original Black-Scholes model to multiple dimensions. It is used to estimate the price of options on assets that are affected by multiple sources of risk, such as commodity futures or interest rate options [1].

The d-dimensional Black-Scholes PDE is[2]:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sum_{i}^{d}\sum_{j}^{d} a_{ij} S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j} + \sum_{i}^{d} b_i S_i \frac{\partial V}{\partial S_i} - rV + rv = 0 \tag{10}$$

To simply the Eqn. 10, we use the same transformation as

$$x_i = \ln S_i \tag{11}$$

The simplified d-dimensional Black-Scholes PDE will be

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sum_{i,j}^{d} a_{ij} \frac{\partial^2 V}{\partial x_i \partial x_j} + \sum_{i}^{d} (b_i - \frac{1}{2} a_i) \frac{\partial V}{\partial x_i} - rV + rv = 0 \tag{12}$$

Potentially, further transformation can simplify this differential equation to second-order Schrödinger equation with aspect to d directions. We can then use the separation of variables and find the coupled solution. For a d-dimensional BS model, to get the same accuracy as the model in this Hackathon, the number of qubits we need will be 4^d, and the Hamiltonian will be a $d+1$ dimensional tensor.

6 Approaching using plane waves

Since the Hamiltonian is a linear combination of the momentemom operators and constants, the plane wave basis is the set of eigenvalues. Using the relation

$$\frac{1}{2}\sigma^2\hbar^2 k_l^2 + i(\frac{1}{2}\sigma^2 - r)\hbar k_l + r = \hbar\omega = \lambda_l, l = 1, 2, ..., 2^n \tag{13}$$

using this 2^{n+1} plane waves, we can approach the payoff curve.

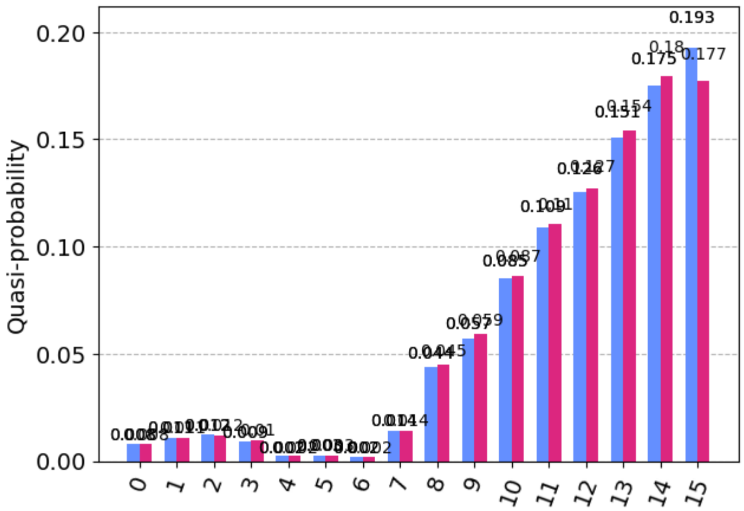

Figure 6: Initial values of the function in blue, and the final ones in red.

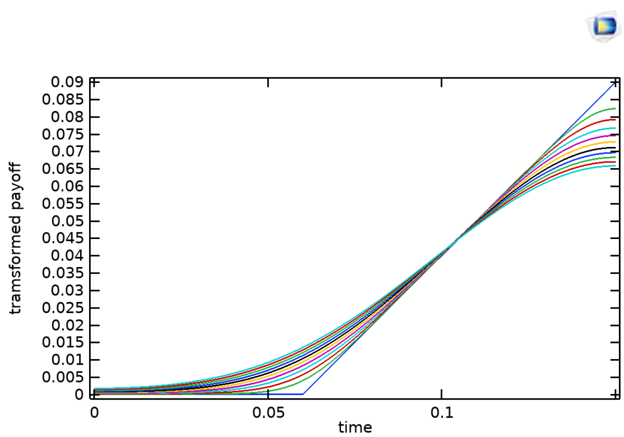

Figure 7: The payoff calculated using Finite-element Method

7 Time-dependent Finite-element Method

he time-dependent finite-element method (TDFEM) is a numerical technique used to solve time-dependent partial differential equations (PDEs) using the finite-element method (FEM)[3]. It involves discretizing the time interval of interest and solving the PDE at each time step using the FEM. The weak form is obtained by multiplying the PDE by a test function and integrating over the domain of interest. This results in a set of integral equations that are typically easier to handle than the original PDE.

The weak form of the simplified Hamiltonian (Eqn. 8) is

$$\int_x -(\frac{\partial u}{\partial t}\tilde{u} - \frac{1}{2}\frac{\partial u}{\partial x}\frac{\partial \tilde{u}}{\partial x}) = 0 \tag{14}$$

A corresponding payoff price solved by any COMSOL multiphysics is given below.

8 Final Conclusions

Even though the results obtained more or less match the analytical solutions of a smoothing diffusion process (as ashown in figure 7), we cannot say that the evolution is being translated clearly into the graphs on which we are basing our judgement. On top of that, some of the variable changes have not been fully understood, so we suspect that in the evolving phase using the VarQITE class, some errors of this nature have contributed to this problem. In any case, further research should be done to minimize these kinds of errors, and also real quantum simulation could be tried to see if these problems also translate into the real world.

References

[1] John C Hull. *Options, Futures, and Other Derivatives.* Pearson Education, 10th edition, 2017.

[2] Dajun Li and Xin Chen. Numerical solution of multi-dimensional black-scholes equations with applications to bond and commodity options. *Applied Mathematics and Computation*, 186(1):775–792, 2007.

[3] O. C. Zienkiewicz. The finite element method for time-dependent problems. *International Journal for Numerical Methods in Engineering*, 12(9):1593–1619, 1977.