Assignment - Financial Engineering: Structuring a Bonus Certificate using the MCD stock as an underlying

Ward Vandecruys, r0908427

May 2025

Contents

1 Introduction

Are you looking to invest in a global, well-known, and recognizable brand like McDonald's, but you have doubts about the potential downturns in the stock market? Then we have made a product tailored to your needs! By investing in our McDonald's Bonus Certificate, you are giving yourself the potential to reap the rewards as if you were to buy the stock, while also having the comfort of having a cushion to fall onto when the stock incurs limited losses. That's right, we will give you an attractive payout at the bonus level as long as the losses are not too dire, because we make it our business to give you the best support to grow your wealth, all tailored to your risk profile.

By purchasing our McDonald's Bonus Certificate, you will be guaranteed a payout on the 18th of June 2026 that varies based on the performance of the McDonald's stock (MCD). When the product outperforms our bonus guarantee, we will pay you the value of the stock at the time of maturity, S_T. If the stock underperforms between the bonus guarantee and the lower barrier level, then we guarantee you the payout of the bonus guarantee ($B_{BC} = 1.1 * S_0 = 343.11$), which nets a return higher than the risk-free rate ($r_{BC} = 5\%$). Your only exposure to losses comes when the stock breaches this barrier ($H_{BC} = 0.8 * S_0 = 249.54$) at any time between the time of purchase and the time of maturity of the product; this deactivates the bonus guarantee protection, making our product equivalent to owning and selling the stock at the time of maturity. This means that to take a loss on our product, the barrier has to be breached and the stock has to be valued lower than what you paid at inception.

We take pride in our work and guarantee all this at the reasonable price of 326.79 dollars, which is only 14.87 dollars more than buying the stock itself; the price of having a buffer to feel more secure and comfortable towards the future has never been this cheap!

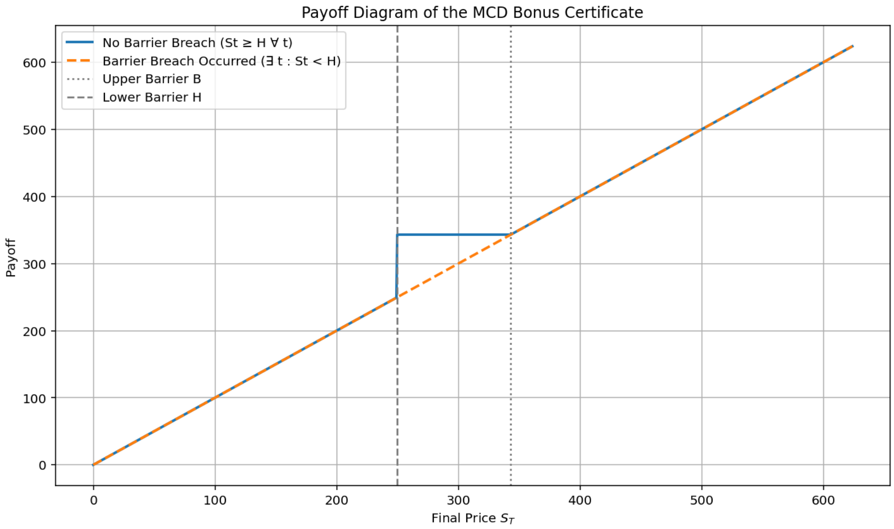


Figure 1: McDonald's Bonus Certificate Payoff Diagram

2 Technical Analysis For Decision Makers

In this section, we will discuss the whole technical analysis of this product, ranging from introducing the product to valuation, and hedging.

2.1 The Product

We have come up with the idea to construct a structured product based on the McDonald's stock that has the following piece-wise payout structure:

$$\text{payoff}_{BC} = \begin{cases} S_T, & \text{if } S_T > B \\ B, & \text{if } H \leq S_T \leq B \text{ and } S_t \geq H \;\forall\, t \in [0, T] \\ S_T, & \text{if } \exists\, t \in [0, T] \text{ such that } S_t < H \end{cases} \tag{1}$$

Where:

- S_T is the price of the underlying at maturity T.
- S_t is the price of the underlying at any time $t \in [0, T]$.
- T is the time of maturity of the Bonus Certificate, expressed in years.
- B is the bonus level, a predetermined value above the price at inception to guarantee profit.
- H is the barrier level, which is a lower bound at which the protection feature is deactivated.

This product can be constructed out of a zero-strike call option, which is equivalent to owning the underlying stock, and a down-and-out put option (a type of knock-out option). While the first one can be easily obtained through regular market channels, the latter can not. A down-and-out put option has contingent behavior depending on a barrier that can not be **optimally** constructed using principal option components.

Here, the zero-strike call has the following has the following payout, given that the stock price can not go lower than zero:

$$\text{payoff}_{\text{ZSC}} = S_T \tag{2}$$

While the second component, the down-and-out put option, has a more exotic payoff behavior:

$$\text{payoff}_{\text{DOBP}} = \begin{cases} \max(B - S_T, 0), & \text{if } S_t \geq H \;\forall\, t \in [0, T] \\ 0, & \text{if } \exists\, t \in [0, T] \text{ such that } S_t < H \end{cases} \tag{3}$$

2.2 Valuation

In the process of calibrating the Bates model and pricing the product, we will rely on all the mid prices for call and put options retrieved from Yahoo Finance for each maturity date listed in Table 2 by utilizing the package 'yfinance' in Python.

2.2.1 Risk-free rates and dividends

The risk-free rates were taken from the US treasury due the fact that our stock is traded on the New York Stock Exchange, it also has a dividend payout. All of these features are put in dollars; hence, for simplicity, we will continue to work in dollars.

The forward annual dividend yield of the MCD stock is listed at 2.26% according to Morningstar. The treasury data that we need is the yield curve, specifically, we will restrict ourselves to yield data on a yearly basis. Table 1 shows the yield data and dividend data used in this study.

Table 1: U.S. Treasury Yields by Maturity and MCD Dividend

Maturity (years)	**Yield (%)**
1	4.11
2	3.98
3	3.97
5	4.09
7	4.27
10	4.45
20	4.92
30	4.89
Annual Dividend Yield (MCD)	**2.26**

Our product will reach maturity on the 18th of June 2026, matching the maturity date of MCD options, however, we will calibrate the Bates model on four additional maturities spread evenly around this date. We therefore need to find the yields for each maturity. To find these yield rates, we will use a simplification by using linear interpolation between the yields listed in Table 1.

$$r(T) = r_i + \left(\frac{T - T_i}{T_{i+1} - T_i} \right) (r_{i+1} - r_i) \tag{4}$$

By utilizing this equation, we can calculate the following yields for the following maturity dates listed in Table 2

Table 2: Interpolated Risk-Free Yields by Maturity

Maturity Date	**Years to Maturity**	**Interpolated Yield (%)**
2026-01-16	0.681725	4.11%
2026-03-20	0.854289	4.11%
2026-06-18	1.10062	4.10%
2026-12-18	1.60164	4.03%
2027-01-15	1.6783	4.02%

2.2.2 The Bates Model

Our chosen model is the Bates model, which combines the benefits of the Heston model with the addition of jumps in our process (similar to a Merton model). Outside of these jumps, the Bates model also enables us to use the stochastic volatility of the Heston model. This stochastic volatility is a shortcoming of the Black-Scholes model, as it is not accounted for there. The Bates model with dividends can be written down in the following equation:

$$\frac{dS_t}{S_t} = (r - q - \lambda \mu_J)dt + \sqrt{v_t} dW_t + J_t dN_t, \quad S_0 \geq 0, \tag{5}$$

$$dv_t = \kappa(\theta - v_t)\,dt + \sigma\sqrt{v_t}\,dW_t^v, \quad v_0 \geq 0 \tag{6}$$

Due to the 10-page limit, we will omit the explanation of each term and the mechanics of the Bates model, because this would be rather long. We assume that the risk managers know of the mechanics of the Bates model itself and instead need technical information on the calibration and Monte Carlo methods used.

This model needs to be fitted to the option data that is retrieved through the 'yfinance' package. This fitting procedure itself requires a set of numerical techniques, such as the Fast Fourier (FFT) algorithm, specifically the Carr-Madan FFT formula, as well as optimization through the minimization of the sum of squared errors between the model and market data. These steps will all be explained in the calibration section.

2.2.3 Calibration

As mentioned, we will be using the Carr-Madan Formula to fit the Bates model, This is done by using the characteristic equation of the Bates model

$$C(K) = \frac{e^{-\alpha k}}{\pi}\mathrm{Re}\left[\int_0^\infty e^{-iuk}\frac{e^{-rT}\phi(u - i(\alpha+1))}{\alpha^2+\alpha-u^2+i(2\alpha+1)u}\,du\right] \tag{7}$$

Numerical integration can be a slow process; using the FFT algorithm greatly improves our efficiency in pricing the options, which is also made possible because we are supplied a characteristic equation of our model. This is done by using the following numerical approximation:

$$C(K) \approx \frac{e^{-\alpha k}}{\pi}\cdot \mathrm{Re}\left\{\sum_{j=0}^{N-1} e^{-iu_j b}\cdot\left[\frac{e^{-rT}\cdot\phi(u_j - i(\alpha+1))}{\alpha^2+\alpha-u_j^2+i(2\alpha+1)u_j}\right]\cdot w_j \cdot e^{iu_j k}\right\} \tag{8}$$

Where $\phi(u_j)$ is the log-characteristic equation of the Bates model, which can be divided into many components.

$$\phi(u) = \exp\Big(C(u,T) + D(u,T)\cdot v_0 + iu\ln(S_0 e^{(r-q)T}) + J(u,T)\Big) \tag{9}$$

Here d, g, $C(u,T)$, and $D(u,T)$ are the same as their Heston model equivalent. These parameters supply information on the volatility component of the model.

Firstly, $C(u,T)$ captures the mean-reverting variance component of $\phi(u)$.

$$C(u,T) = (r-q)iuT + \frac{\kappa\theta}{\eta^2}\left[(\kappa - \rho\eta iu - d)T - 2\ln\left(\frac{1-ge^{-dT}}{1-g}\right)\right] \tag{10}$$

Secondly, $D(u,T)$ is the sensitivity of $phi(u)$ to the initial variance v_0

$$D(u,T) = \frac{\kappa - \rho\eta iu - d}{\eta^2}\cdot\frac{1-e^{-dT}}{1-ge^{-dT}} \tag{11}$$

The component d comes into existence by solving the Riccati differential equations.

$$d = \sqrt{(\rho\eta iu - \kappa)^2 + \eta^2(iu+u^2)} \tag{12}$$

Lastly g, expresses the exponential decay in $C(u,T)$ and $D(u,T)$

$$g = \frac{\kappa - \rho\eta iu - d}{\kappa - \rho\eta iu + d} \tag{13}$$

The previously listed components are identical to those of the Heston model; the difference between the Heston model and the Bates model is the inclusion of a jump component $J(u,T)$.

$$J(u,T) = \lambda T\left(e^{iu\mu_J - \frac{1}{2}\sigma_J^2 u^2} - 1\right) \tag{14}$$

These equations gave rise to many different parameters that need to be fitted to find the final Bates model that will be used for options pricing. To optimize it, we will use the sum of squared errors (SSE) between the predicted price by the model for a call and a put and subtract the real market price to find the residual, which is then squared to find the size of the error on an absolute scale (similar to OLS). We also use the Feller condition as a constraint to ensure this condition is met: $2\kappa\theta - \eta^2 > 0$.

$$\min_{\kappa,\theta,\eta,\rho,v_0,\lambda,\mu_J,\sigma_J} \quad \mathrm{SSE}_{\text{calls+puts}} \quad \text{subject to} \quad 2\kappa\theta - \eta^2 > 0 \tag{15}$$

We can view the $\mathrm{SSE}_{\text{calls+puts}}$ as a function of all these parameters, which yields the following equation:

$$\begin{aligned} \mathrm{SSE}_{\text{calls+puts}} = & \sum_{i=1}^{N_C} \left[C^{\text{model}}(K_i, T_i;\, \kappa, \theta, \eta, \rho, v_0, \lambda, \mu_J, \sigma_J) - C^{\text{market}}(K_i, T_i)\right]^2 \\ & + \sum_{j=1}^{N_P} \left[P^{\text{model}}(K_j, T_j;\, \kappa, \theta, \eta, \rho, v_0, \lambda, \mu_J, \sigma_J) - P^{\text{market}}(K_j, T_j)\right]^2 \end{aligned} \tag{16}$$

2.2.4 Calibration results

In this subsection, we will cover the results of the calibration procedure. This is done in an iterative manner, where in each iteration the model parameters change to find a (local) minimum in the sum of squared errors, while subject to the Feller condition.

In the course of the assignment, a lot of iterations of the price were made. Listed below is a general outline of the program that calibrates the Bates model. To save space, we have not included any functions themselves outside of the SSE optimizing routine.

Algorithm 1 Bates Model Calibration via Carr–Madan FFT

Require: Underlying symbol s, target expiry T^*, dividend yield q, parameter bounds $\{[l_i, u_i]\}_{i=1}^{8}$
Ensure: Calibrated parameters $\theta^* = (\kappa, \theta, \eta, \rho, v_0, \lambda, \mu_J, \sigma_J)$

$S_0 \leftarrow \mathrm{fetchSpotPrice}(s)$
$\mathcal{E} \leftarrow \mathrm{sortedExpiries}(s)$
Find index i such that $\mathcal{E}[i] = T^*$
$\mathcal{E}_{\text{sel}} \leftarrow \{\mathcal{E}[i-2], \mathcal{E}[i-1], \mathcal{E}[i], \mathcal{E}[i+1], \mathcal{E}[i+2]\}$

for all $e \in \mathcal{E}_{\text{sel}}$ **do**
  $T \leftarrow \mathrm{yearFrac}(\mathrm{today}, e)$
  $r \leftarrow \mathrm{interpolateRate}(e)$
  $(K_c, C^m) \leftarrow \mathrm{fetchCallMidQuotes}(s, e)$
  $(K_p, P^m) \leftarrow \mathrm{fetchPutMidQuotes}(s, e)$
  collect $(S_0, r, T, K_c, C^m, K_p, P^m)$ into data set $\mathcal{D}$
end for

SSE_θ
$E \leftarrow 0$
for all $(S_0, r, T, K_c, C^m, K_p, P^m) \in \mathcal{D}$ **do**
  $C_{\text{mod}} \leftarrow \mathrm{FFTPrice}(\theta, S_0, r, T, K_c)$
  $E \leftarrow E + \|C_{\text{mod}} - C^m\|^2$
  $P_{\text{mod}} \leftarrow C_{\text{mod}} - S_0 e^{-qT} + K_p e^{-rT}$
  $E \leftarrow E + \|P_{\text{mod}} - P^m\|^2$
end for
return E

$\theta^* \leftarrow \mathrm{DifferentialEvolution}(\mathrm{SSE}, \{[l_i, u_i]\})$
return θ^*

We used the Differential Evolution solver to run the optimization under the Feller condition constraint. In this iteration, we also used the put-call parity to price put options. This algorithm uses the bounds in table 3, the resulting optimized variables can be found in the same table

Table 3: Calibrated Bates Model Parameters, Feller Margin, and DE Bounds

Parameter	Value	Lower Bound	Upper Bound
κ	7.365507	1×10^{-3}	10
θ	0.015767	1×10^{-4}	2
η	0.235477	1×10^{-4}	1
ρ	-0.664023	-0.99	0.99
v_0	0.025546	1×10^{-6}	1
λ	0.301500	1×10^{-4}	1
μ_J	-0.215104	-1	1
σ_J	0.460238	1×10^{-4}	1
$2\kappa\theta - \eta^2$	0.1768145	—	—

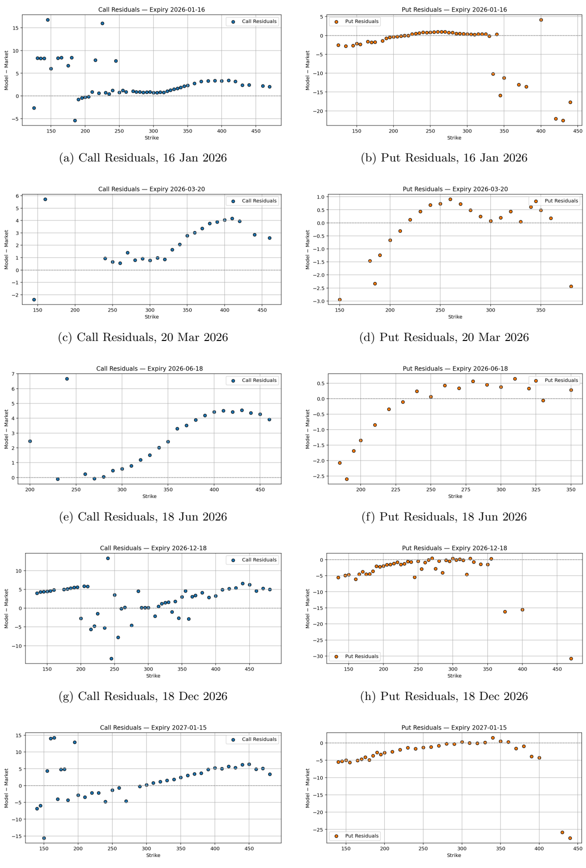
These parameters were found by getting the following minimized errors shown in table 4. I have also listed the Root Mean Square Error (RMSE) and the Mean Absolute Error to get a better view on the expected pricing errors that can be made by the current model. The residuals for each maturity can be analyzed by looking at Figure 2.

Table 4: Fit Statistics for Bates Model Calibration

Metric	Value
Data Points	358
SSE	11840.8309
RMSE	5.7511
MAE	3.4574

A kappa of 7.37 indicates a mean reversion every 35 days, which is plausible for equity. A long-run variance θ of 0.0158 means that the long run volatility is approximately 12.6%, which is in line with the SP500. An initial variance v_0 of 0.0255 indicates that the volatility starts at 16%, while this is large, it could be that the calibration period contained this pattern. A volatility of volatility η of 0.235 indicates a realistic implied volatility surface. A correlation ρ of -0.66 shows the presence of the leverage effect that can be seen in equity instruments, which is good. The jump intensity λ is only 0.3 per year, which corresponds to a jump every 3-4 years. This is rather low but not improbable. The jump size μ_J equals -0.215, equivalent to an average drop in price of approximately 19% of the value. The jump volatility σ_J is set at a value of 0.46; this is large but can be due to fat tails in our underlying.

When we look at table 4 we can derive that the calibration is a moderate success. Judging by the mean absolute error (MAE) we can say that our model makes an average error of 3.46 dollars. When we look at the residuals in Figure 2, we can conclude that our model has issues pricing the outlier regions. We can see the same pattern in our error results because the MAE is less sensitive to outliers; it has a significantly lower value than the root mean square error (RMSE), which is 5.75.



(a) Call Residuals, 16 Jan 2026

(b) Put Residuals, 16 Jan 2026

(c) Call Residuals, 20 Mar 2026

(d) Put Residuals, 20 Mar 2026

(e) Call Residuals, 18 Jun 2026

(f) Put Residuals, 18 Jun 2026

(g) Call Residuals, 18 Dec 2026

(h) Put Residuals, 18 Dec 2026

(i) Call Residuals, 15 Jan 2027

(j) Put Residuals, 15 Jan 2027

Figure 2: Model vs. Market: Call (left) and Put (right) residuals for each of the five maturities.

2.2.5 Monte Carlo Simulation

Barrier options such as the down-and-out put option are path-dependent, meaning they are dependent on the trajectory. This makes pricing a lot more difficult, so we turn to Monte Carlo methods to price the derivative.
To achieve this, we must discretize the Bates model to simulate paths step by step. The Bates model, like the Heston model, relies on the Cox-Ingersoll-Ross (CIR) process for the stochastic evolution of its variance.

$$dv_t = \kappa(\theta - v_t)\,dt + \sigma\sqrt{v_t}\,dW_t^v \tag{17}$$

To discretize this, we must rely on schemes such as an Euler-Maruyama, Milstein, or QE scheme. One caveat is the fact that we cannot go negative ($v_t \geq 0$), which is guaranteed for the QE scheme and can be forced in the others, such as a semi-implicit Euler scheme that uses clipping. In the context of this assignment, I opted for Andersen's Quadratic Exponential (QE) scheme. Because it already guarantees $v_{t+\Delta t} \geq 0$ without fixes, it also matches the two moments, which is helpful because we know the conditional mean and variance of the CIR process. The following paragraph will go deeper into the mechanics of the QE scheme.
The first step in the QE scheme is to calculate the conditional moments at time t:

$$m = \mathbb{E}[v_{t+\Delta t} \mid v_t] = \theta + (v_t - \theta)e^{-\kappa\Delta t} \tag{18}$$

$$s^2 = \mathrm{Var}[v_{t+\Delta t} \mid v_t] = \frac{\kappa v_t \sigma^2 e^{-\kappa\Delta t}(1 - e^{-\kappa\Delta t})}{\kappa} + \frac{2\kappa\theta\sigma^2}{\kappa^2}(1 - e^{-\kappa\Delta t})^2 \tag{19}$$

$$\psi = \frac{m^2}{s^2} \tag{20}$$

Next, we choose the simulation regime by evaluating ψ by comparing it to $\psi_n = 1.5$. By evaluating the dispersion, we can check how stable the variance is if it is stable ($\psi \leq \psi_n$) we go to a quadratic regime to better capture the left-skewedness and bounded properties of the CIR process. As a result, we use a quadratic form for the conditional distribution $a(b + Z)^2$. When the dispersion is high ($\psi > \psi_n$), we go into the exponential regime, which sets a conditional distribution that is heavily skewed towards zero by using a mixture of a point mass at zero and an exponential tail.

$$v_{t+\Delta t} = \begin{cases} \dfrac{m}{1 + \left(\frac{\sqrt{2\psi} - 1 + \frac{1}{2\psi}}{\sqrt{2\psi} - 1}\right)^2} \left(\dfrac{\sqrt{2\psi} - 1 + \frac{1}{2\psi}}{\sqrt{2\psi} - 1} + Z\right)^2, & \text{if } \psi \leq \psi_n \quad \text{(Quadratic Regime)} \\[2ex] \begin{cases} 0 & \text{with probability } \dfrac{\psi - 1}{\psi + 1} \\ -\dfrac{1 - \frac{\psi - 1}{\psi + 1}}{m} \log\left(\dfrac{1 - U}{1 - \frac{\psi - 1}{\psi + 1}}\right) & \text{with probability } 1 - \dfrac{\psi - 1}{\psi + 1} \end{cases}, & \text{if } \psi > \psi_n \quad \text{(Exponential Regime)} \end{cases} \tag{21}$$

We use this v_t to feed the variance path step by step in the log asset price equation. As you can see, v_t is critical for the diffusion and drift term. It is therefore imperative to correctly simulate v_t because it will lead to mispricing of options if done incorrectly.

$$\ln S_{t+\Delta t} = \ln S_t + \underbrace{\left(r - q - \frac{1}{2}v_t - \lambda\bar{k}\right)\Delta t}_{\text{Drift}} + \underbrace{\sqrt{v_t \Delta t} \cdot Z_1}_{\text{Diffusion}} + \underbrace{N_{\Delta t} \cdot \mu_J + \sigma_J\sqrt{N_{\Delta t}} \cdot Z_J}_{\text{Jump}_t} \tag{22}$$

We also need to make sure that the barrier has not been breached at any step in the path simulation to be able to price the down-and-out put option.

$$\ln S_{t+\Delta t} > \ln H \quad \textbf{OR} \quad S_t > H \tag{23}$$

We can keep track of a Not Breached (NB) Boolean marker for each path that uses the following condition:

$$\mathrm{NB}_i \;=\; \mathbf{1}_{\left\{\, S_t^{(i)} > B \;\forall\, t \in [0,T] \,\right\}} \tag{24}$$

The final payoff of this path is calculated by using the payoff equation of the down-and-out put option conditioned on the NB indicator.

$$\text{payoff}_i \;=\; \mathrm{NB}_i \;\cdot\; \max\!\big(K - S_T^{(i)},\, 0\big) \tag{25}$$

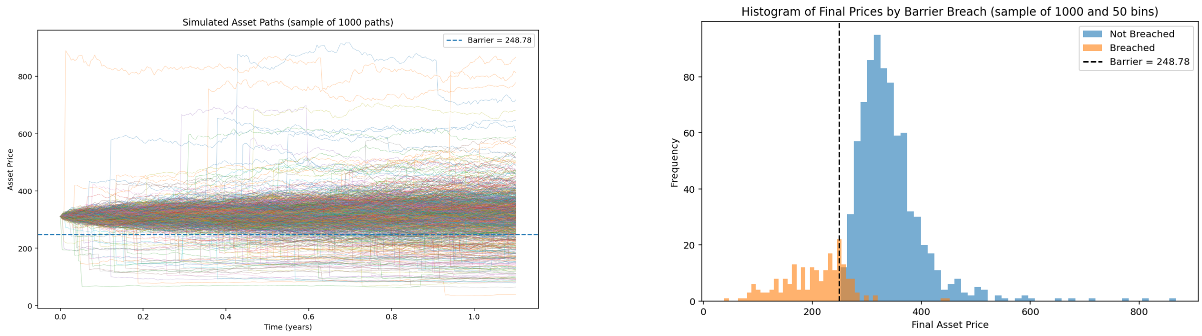
Once we have simulated a large number of paths, each consisting of n we can approximate the price of the down-and-out option by taking the average of all simulated payoffs and then discounting it to the present with the risk-free rate because we are in the risk-neutral world.

$$\text{Price} \approx e^{-rT} \cdot \frac{1}{N} \sum_{i=1}^{N} \text{payoff}_i \tag{26}$$

2.2.6 Pricing the options and product

In this subsection, we will apply the Monte Carlo methodology to compute the price of the MCD down-and-out put option. To do this, we first of all use the calibrated parameters for the Bates model shown in table 3 with a risk-free rate of 4.11%, a dividend rate of 2.26%, and a maturity of 1.10062 years.

We then continue to simulate 1000000 paths consisting of 1024 steps using our Monte Carlo method. Figure 3 shows a sample of the 1000 paths till maturity as well as a histogram of the final prices conditioned on whether the barrier was breached or not.



(a) Sample of the paths taken in the Monte Carlo Simulation

(b) Sample histogram of the final prices and the classification

Figure 3: Sample of the Monte Carlo Method

The final results of the Monte Carlo pricing can be found in Table 5. Here we see that the final price for a Down-And-Out Put Option given a barrier level of $0.8 * S_0$ and a bonus level of $1.1 * S_0$, when the underlying is the MCD stock, is 16.0746 dollars.

The last step before finding the final of our bonus certificate is pricing the zero-strike call option, which is the equivalent of buying a stock. The overall price of a zero-strike call option can be found by discounting the stock price S_0 by the dividend rate, as it will be paid out over time.

$$C_0 \;=\; S_0\, e^{-qT} = 311.92 * e^{-0.0226*1.10062} = 304.256998 \tag{27}$$

The final price of the bonus certificate is the sum of these two components times a multiplier that signifies the margin that we get for structuring this product. For the sake of keeping the product attractive, we want to ensure the overall rate at the bonus level is higher than the current risk-free rate until the 18th of June 2026. We therefore pick a profit margin of 2%, yielding the following overall price for our bonus certificates:

$$P_{BC} \;=\; (P_{DAOP} + C_0) * (1 + m) = (16.123153 + 304.256998) * (1 + 0.02) = 326.787754 \tag{28}$$

We have put the price of each component and the parameters that were needed to find the price in Table 5.

Table 5: Bonus Certificate Pricing with maturity 18th of June 2026

Component	Value
Monte Carlo DaO put	**$16.123153**
Initial stock price	$311.92
Annual dividend rate	2.26%
Time to Maturity	1.10062
Zero-strike Call	**$304.256998**
Profit Margin	2%
Bonus Certificate	**$326.79**

Lastly, we want to evaluate the equivalent interest rate when receiving the bonus level of the bonus certificate between $t = 0$ and $t = T$.

$$r_{BC} = \frac{B_{BC} - P_{BC}}{P_{BC}} = \frac{1.1 * S_0 - P_{BC}}{P_{BC}} = \frac{343.11 - 326.79}{326.79} = 0.04994 \approx 5\% \tag{29}$$

We can therefore state that our bonus certificate is an interesting investment opportunity that gives exposure to the upward potential of the stock, while having an above risk-free rate bonus level to fall onto if the option is not knocked out.

2.3 Hedging

The last objective is to delta hedge our position when we go short on our structured product. We will realize this by using a finite-difference approximation. Before we do this, we will first explain the mechanics of a delta hedge.

The goal of delta hedging is to adjust our portfolio in such a way that the underlying asset's price is no longer of consequence for the portfolio value. In our case, this implies that a shift in the value of the MCD stock does not impact our net short position in these bonus certificates. This is done by taking a position in the underlying asset, where we either buy or sell a specific amount of shares to make our portfolio delta-neutral.

$$\Delta \; = \; \frac{\partial V}{\partial S_0} \tag{30}$$

Here V consists out of the components of our product without taking into account the profit margin.

$$V(S_0) \; = \; C_0(S_0) \; + \; P_{\mathrm{DaO}}(S_0) = S_0\, e^{-qT} + P_{\mathrm{DaO}}(S_0) \tag{31}$$

By taking the derivative, we find the following equation for the Δ_{cert}.

$$\Delta_{\mathrm{cert}} = \frac{\partial V}{\partial S_0} = e^{-qT} \; + \; \frac{\partial P_{\mathrm{DaO}}(S_0)}{\partial S_0} \tag{32}$$

We then use finite-difference approximation to find Δ_{cert}. Here, we first need to find set a relative perturbation $\epsilon = 1 * 10^{-4}$ to find the absolute perturbation $\Delta S \; = \; \varepsilon \cdot S_0$. This absolute perturbation is then used to find two spot price around S_0.

$$\begin{aligned} S_u &= S_0 + \Delta S \\ S_d &= S_0 - \Delta S \end{aligned} \tag{33}$$

We then calculate Δ_{cert} using a central difference scheme:

$$\Delta_{\mathrm{cert}} \; \approx \; \frac{V\big(S_u\big) \; - \; V\big(S_d\big)}{2\,\Delta S} = \frac{V\big(S_0 + \Delta S\big) \; - \; V\big(S_0 - \Delta S\big)}{2\,\Delta S} \tag{34}$$

These two situations need to be priced similarly to those done in the previous sections. Namely, we need to price two down-and-out put options at both S_u and S_d and reconstruct the price of the product with the respective stock price.

Once Δ_{cert} is known, we can move on to finding the amount of bonus certificates that we are short when someone invests 1000000 dollars into them.

$$N_{\text{cert}} \;=\; \frac{N_{\text{notional}}}{V\big(S_0\big)} \tag{35}$$

We need to hedge this by taking an opposite position in the underlying using the Δ_{cert}.

$$N_{\text{shares}} = N_{\text{cert}}\,\Delta_{\text{cert}} \tag{36}$$

- If $\Delta_{\text{cert}} > 0$ then we buy N_{shares} shares
- if $\Delta_{\text{cert}} < 0$ then we sell $|N_{\text{shares}}|$ shares

This gives the following portfolio at $t = 0$:

$$\Pi = N_{\text{cert}}\,V(S_0) \;-\; N_{\text{shares}} \tag{37}$$

Where:

$$\frac{\partial \Pi}{\partial S} = N_{\text{cert}}\,\Delta_{\text{cert}} \;-\; N_{\text{shares}} = 0 \tag{38}$$

We can therefore say that our portfolio is delta-neutral at $t = 0$. This process needs to be done dynamically over the remaining time until maturity. The results of this algorithm can be found in Table 6, where we simulate a case where we are 1000000 short in these bonus certificates. These results indicate that we need to buy 2278 MCD stocks to keep our portfolio delta-neutral.

Table 6: Hedging when short 1000000 dollars in bonus certificates at $t = 0$

Component	**Value**
Bonus Certificate	\$326.79
Δ_{cert}	0.7444
Number of certificates	3060.1
Delta hedge	**buy 2278 shares**

3 APPENDIX: Data screenshots

The stock value used (forgot to screenshot it at first) from yahoo finance:

Date	Open	High	Low	Close	Adj Close	Volume
May 12, 2025	312.01	312.18	305.01	311.92	311.92	4,137,900

Figure 4: Stock value found on yahoo value for the 12th of May 2025

Annual dividend rate found on Morningstar:

Figure 5: Dividend rate found on morningstar

Yields from the US Treasury, specifically the rates on the 12th of May 2025 (from 1Y onwards):
Annual dividend rate found on Morningstar:

Date	1 Mo	1.5 Mo	2 Mo	3 Mo	4 Mo	6 Mo	1 Yr	2 Yr	3 Yr	5 Yr	7 Yr	10 Yr	20 Yr	30 Yr
05/01/2025	4.38	4.36	4.34	4.31	4.38	4.22	3.92	3.70	3.69	3.81	4.02	4.25	4.75	4.74
05/02/2025	4.38	4.36	4.34	4.33	4.41	4.26	4.00	3.83	3.82	3.92	4.11	4.33	4.81	4.79
05/05/2025	4.38	4.36	4.34	4.33	4.40	4.27	4.02	3.83	3.78	3.95	4.14	4.36	4.84	4.83
05/06/2025	4.37	4.37	4.33	4.33	4.38	4.25	3.98	3.78	3.73	3.90	4.10	4.30	4.82	4.81
05/07/2025	4.37	4.37	4.33	4.34	4.39	4.27	4.00	3.78	3.72	3.87	4.06	4.26	4.78	4.77
05/08/2025	4.37	4.37	4.35	4.34	4.40	4.28	4.05	3.90	3.85	4.00	4.18	4.37	4.86	4.83
05/09/2025	4.37	4.36	4.34	4.34	4.40	4.28	4.05	3.88	3.85	4.00	4.18	4.37	4.86	4.83
05/12/2025	4.38	4.37	4.36	4.42	4.45	4.29	4.11	3.98	3.97	4.09	4.27	4.45	4.92	4.89

Figure 6: Yields from the US Treasury

The options themselves were retrieved from Yahoo Finance through a package within Python.