Hedging and Pricing European Options with Transaction Costs: From Stochastic Control to Deep Learning

Jules Arzel

March 2025

Abstract

This paper studies the problem of hedging a short European call option under a utility maximization framework. We first derive the optimal strategy using stochastic control, both in the frictionless case and with proportional transaction costs, and implement a dynamic programming scheme for numerical solutions. In the second part, we adopt a deep hedging approach, training neural networks to learn optimal policies from simulated data. The classical analysis provides key structural insights—such as the no-transaction region and the pricing-hedging duality—that inform network design and interpretation, creating a bridge between model-based and data-driven methods for option pricing under frictions.

Key Words : Stochastic control, Deep Learning, Option Hedging, Transaction costs

1 Optimal investment problem without transaction costs

1.1 Problem setting

We consider a portfolio with two assets : one risky and one non risky asset. The first one follows a geometric brownian motion :

$$dS = \alpha S dt + \sigma S dZ_t \tag{1}$$

The second is such that :

$$B_t = Be^{rt} \tag{2}$$

with B the initial amount invested in the asset. Lastly, the portfolio has a short position in an European call option on the risky asset, with strike K and maturity T, with payoff :

$$C_T = (S_T - K)^+ \tag{3}$$

The investor has a utility function u twice differentiable and concave, such that his objective is to maximise his expected utility of the terminal wealth W_T, with :

$$W_T = B_T + y_T S_T - C_T \tag{4}$$

$y_T S_T$ being the cash value of his position on the risky asset at maturity.

1.2 Optimal investment without options

There is no consumption before maturity so the portfolio is self-financing, so when there is no option position in the portfolio, we have :

$$dW_t = dB_t + y_t dS_t \tag{5}$$
$$= rB_t dt + y_t(\alpha S_t dt + \sigma S_t dZ_t) \tag{6}$$

By setting $\pi_t W_t = y_t S_t$ (the proportion of wealth allocated in the risky asset at time t), and therefore $(1 - \pi_t)W_t = B_t$, we have :

$$dW_t = (\alpha - r)\pi_t W_t dt + rW_t dt + \sigma \pi_t W_t dZ_t \tag{7}$$

As we mentioned, the investor's goal is to maximize his expected utility at maturity, which translates as :

$$\max_{\pi} E(u(W_T)) \tag{8}$$

and we define the indirect function utility function at time t for a wealth W (the value function of our stochastic control problem) as :

$$U(t, W) = \max_{\pi} E_t(u(W_T)) \tag{9}$$

By Itô's formula, we have :

$$dU(t,W) = \frac{\partial U}{\partial t}dt + \frac{\partial U}{\partial w}dW_t + \frac{1}{2}\frac{\partial^2 U}{\partial w^2}(dW_t)^2 \tag{10}$$
$$= \frac{\partial U}{\partial t}dt + \frac{\partial U}{\partial w}((\alpha - r)\pi_t W_t dt + rW_t dt + \sigma \pi_t W_t dZ_t) \tag{11}$$
$$+ \frac{1}{2}\frac{\partial^2 U}{\partial w^2}\sigma^2 \pi^2 W_t^2 dt \tag{12}$$

Therefore :

$$E(dU(t,W)) = \frac{\partial U}{\partial t}dt + \frac{\partial U}{\partial w}((\alpha - r)\pi_t W_t + rW_t)dt + \frac{1}{2}\frac{\partial^2 U}{\partial w^2}\sigma^2 \pi^2 W_t^2 dt \tag{13}$$

And :

$$\max_{\pi} E(dU(t,W)) = \frac{\partial U}{\partial t}dt + \max_{\pi}(\frac{\partial U}{\partial w}((\alpha - r)\pi_t W_t + rW_t)dt + \frac{1}{2}\frac{\partial^2 U}{\partial w^2}\sigma^2 \pi_t^2 W_t^2 dt) \tag{14}$$

By Bellman's principle of optimality ("If I'm already following the optimal strategy, then doing an optimal action in the next dt should give no advantage — I'm already on the best possible path."), if U is optimal for t onward, at time t the best decision we can make regarding the control π leads to $E(\frac{dU}{dt}) = 0$, which leads to the Hamilton-Jacobi-Bellman equation, dividing by dt and setting the derivative to 0 for optimality :

$$\boxed{\frac{\partial U}{\partial t} + \max_{\pi} L^{\pi} U = 0} \tag{15}$$

with the following differential operator :

$$L^{\pi}U = \frac{\partial U}{\partial w}((\alpha - r)\pi_t W_t + rW_t) + \frac{1}{2}\frac{\partial^2 U}{\partial w^2}\sigma^2\pi_t^2 W_t^2 \tag{16}$$

and the boundary condition :

$$U(T, W) = u(W), \forall W \geq 0 \tag{17}$$

Once we know the indirect utility function U, we can find the optimal allocation π^* using first order conditions to solve for $\max_{\pi} L^{\pi}U$:

$$\frac{\partial L^{\pi}U}{\partial \pi} = \pi_t W^2\sigma^2\frac{\partial^2 U}{\partial w^2} + (\alpha - r)W\frac{\partial U}{\partial w} \tag{18}$$

which leads to :

$$\pi^* = -\frac{(\alpha - r)U_w}{W\sigma^2 U_{ww}} \tag{19}$$

with $U_w = \frac{\partial U}{\partial w}$.

In this paper, we will consider two utility function :

1. $u(W) = 1 - e^{-\gamma W}$, of the class CARA of utilities
2. $u(W) = \frac{W^{\gamma}}{\gamma}$, of the class CRRA of utilities.

- **CRRA Utility** : When there is no option position in the portfolio and for a utility function of the shape $u(W) = \frac{W^{\gamma}}{\gamma}, 0 < \gamma < 1$, the boundary condition gives $U(T, W) = \frac{W^{\gamma}}{\gamma}$ and then the indirect utility function can be found as (Proof in the annex) :

$$U(t, W) = g(t)\frac{W^{\gamma}}{\gamma} \tag{20}$$

with :

$$g(t) = e^{vr(T-t)} \tag{21}$$

and

$$v = \frac{1}{2}(\frac{\alpha - r}{\sigma})^2\frac{1}{1 - \gamma} + r \tag{22}$$

This leads to :

$$\boxed{\pi^* = -\frac{(\alpha - r)U_w}{W\sigma^2 U_{ww}} = \frac{\alpha - r}{\sigma^2(1 - \gamma)}} \tag{23}$$

- **CARA Utility** : When there is no option position in the portfolio and for a utility function of the shape $u(W) = 1 - e^{-\gamma W}, 0 < \gamma < 1$, the boundary condition gives $U(T, W) = 1 - e^{-\gamma W}$ and then the indirect utility function can be found as (Proof in the annex) :

$$U(t, W) = 1 - e^{-\gamma\phi(t)W + \psi(t)} \tag{24}$$

with :

  - $\phi(t) = e^{r(T-t)}$
  - $\psi(t) = \frac{(\alpha - r)^2}{2\sigma^2}(t - T)$

This leads to :

$$\boxed{\pi^* = \frac{(\alpha - r)}{W\sigma^2\gamma e^{r(T-t)}}} \tag{25}$$

And if we define y_t as the optimal holdings in the risky asset at time t, we have :

$$y^* = \frac{(\alpha - r)}{S\sigma^2\gamma e^{r(T-t)}} \tag{26}$$

1.3 When there is a short position in option

In the case where the investor has a short position in an European call option , the terminal wealth becomes :

$$\overline{W_T} = W_T - (S_T - K)^+ \tag{27}$$

With this new formulation, the indirect utility function (value function of the stochastic control problem) becomes :

$$U(t, S, W) = \max_{\pi} E_t(u(\overline{W_T})) \tag{28}$$

In this case, the stock price becomes a variable in the indirect utility function, which changes the computations leading to the HJB equation :

$$dU(t,S,W) = \frac{\partial U}{\partial t}dt + \frac{\partial U}{\partial w}dW_t + \frac{\partial U}{\partial S}dS_t + \frac{1}{2}\frac{\partial^2 U}{\partial w^2}(dW_t)^2 + \frac{1}{2}\frac{\partial^2 U}{\partial S^2}(dS_t)^2 + \frac{\partial^2 U}{\partial S \partial W}dS_t dW_t \tag{29}$$

We have $dS_t = \alpha S_t dt + \sigma S_t dZ_t$ and $dW_t = ((\alpha - r)\pi_t W_t + rW_t)dt + \sigma\pi_t W_t dZ_t$.
These two are correlated (same brownian motion), therefore $dS_t dW_t = \sigma^2\pi_t S_t W_t dt$. This leads to :

$$E(dU(t,S,W)) = (\frac{\partial U}{\partial t} + \frac{\partial U}{\partial w}((\alpha - r)\pi_t W_t + rW_t) + \frac{\partial U}{\partial S}\alpha S_t + \frac{1}{2}\frac{\partial^2 U}{\partial w^2}\sigma^2\pi_t^2 W_t^2 \tag{30}$$

$$+ \frac{1}{2}\frac{\partial^2 U}{\partial S^2}\sigma^2 S_t^2 + \sigma^2\pi_t S_t W_t \frac{\partial^2 U}{\partial S \partial W})dt \tag{31}$$

This time, we define :

$$\overline{L}^{\pi}U = \frac{\partial U}{\partial w}((\alpha - r)\pi_t W_t + rW_t) + \frac{\partial U}{\partial S}\alpha S_t + \frac{1}{2}\frac{\partial^2 U}{\partial w^2}\sigma^2\pi_t^2 W_t^2 + \frac{1}{2}\frac{\partial^2 U}{\partial S^2}\sigma^2 S_t^2 + \sigma^2\pi_t S_t W_t \frac{\partial^2 U}{\partial S \partial W} \tag{32}$$

and by the same argument, this leads to the HJB equation :

$$\boxed{\frac{\partial U}{\partial t} + \max_{\pi} \overline{L}^{\pi}U = 0} \tag{33}$$

with the boundary condition :

$$U(T, S, W) = u(W - (S - K)^+) \tag{34}$$

The same way as before, we can derive the optimal strategy which will depend on the form of the indirect utility function :

$$\boxed{\pi^* = -\frac{\alpha - r}{\sigma^2 W}\frac{U_w}{U_{ww}} - \frac{S}{W}\frac{U_{Sw}}{U_{ww}}} \tag{35}$$

We can prove that in this case, the indirect utility function is linked to the one when there is no position in the option, by the following relation :

$$\overline{U}(t, W) = U(t, W - f(t, S)) \tag{36}$$

where $f(t, S)$ is the price of the call option under the Black-Scholes framework.
Since we know the explicit form of both indirect utility functions, it allows us to get :

- **CRRA Utility** :

$$\boxed{\pi^* = \frac{\alpha - r}{\sigma^2(1-\gamma)} \frac{W - f}{W} + \frac{S}{W} f_S} \tag{37}$$

And :

$$\pi^* W = \frac{\alpha - r}{\sigma^2(1-\gamma)}(W - f) + S f_S \tag{38}$$

- **CARA Utility** :

$$\boxed{\pi^* = \frac{S}{W} f_S - \frac{(\alpha - r)}{W \sigma^2 \gamma e^{r(T-t)}}} \tag{39}$$

And :

$$y^* = f_S - \frac{(\alpha - r)}{S \sigma^2 \gamma e^{r(T-t)}} \tag{40}$$

Interpretation for the CRRA Utility : The proportion of wealth allocated to the risky asset can be decomposed into two components:

- **Classic Merton Component (adjusted):**

$$\frac{\alpha - r}{\sigma^2(1-\gamma)}(W - f)$$

This term reflects the investment proportion based on *wealth at risk*, i.e., $W - f$. Since we are effectively short an option worth f, we cannot treat this value as available wealth. It must be treated as if it is already lost. Thus, the classic Merton proportion still appears, but scaled to reflect the wealth actually at risk.

- **Delta-Hedge Component:**

$$S f_S$$

This is the delta of the option, scaled by the spot price S. It represents the portion of the portfolio used to replicate the option's payoff, thereby hedging our exposure.

Asymptotic Behavior:

- **When W is large:**

$$\pi \to \frac{\alpha - r}{\sigma^2(1-\gamma)}$$

The Merton line re-emerges as the dominant term, and the impact of the option becomes negligible relative to total wealth.

- **When $W \to f$:**

$$\pi \to \frac{S f_S}{W}$$

In this case, we are highly exposed and have no surplus to speculate with. The portfolio is primarily governed by the necessity to hedge the option.

2 Numerical Scheme Without Transaction Costs Using a Binomial Tree

In practical applications, it is often difficult to derive an explicit form for the indirect utility function

$$U(t,S,W) = \max_{\pi} E_t\Big[u\Big(W_T - (S_T - K)^+\Big)\Big], \tag{41}$$

especially when additional complications such as transaction costs are present. In this section, we present a numerical scheme based on a binomial tree method that approximates the value function $U(t,S,W)$ and yields the optimal strategy by backward induction.

2.1 Overview of the Scheme

The aim is to compute, for each node on a discrete grid in time, stock price, and wealth, the optimal expected utility when starting from that node and acting optimally thereafter. In particular, the scheme will:

1. **Model the stock process:** We assume the stock follows a recombining binomial tree. Over a time step Δt, the stock price evolves as

$$S_{t+\Delta t} = \begin{cases} S_t\, u, & \text{with probability } p, \\ S_t\, d, & \text{with probability } 1-p, \end{cases} \tag{42}$$

where

$$u = \exp\Big(\sigma\sqrt{\Delta t}\Big), \quad d = \exp\Big(-\sigma\sqrt{\Delta t}\Big), \tag{43}$$

and the probability p is given by

$$p = \frac{e^{\alpha\,\Delta t} - d}{u - d}. \tag{44}$$

2. **Model the wealth dynamics:** The investor allocates a fraction π_t of wealth W_t to the risky asset. For a small time step Δt, the wealth update is approximated by

$$W_{t+\Delta t} = W_t\Big(1 + r\,\Delta t + \pi_t\Big[(\alpha - r)\Delta t \pm \sigma\sqrt{\Delta t}\Big]\Big), \tag{45}$$

where the "+" corresponds to the up move (i.e., $S_{t+\Delta t} = S_t\, u$) and the "−" to the down move (i.e., $S_{t+\Delta t} = S_t\, d$).

3. **Set the terminal condition:** At maturity T, the investor's terminal wealth, is

$$\overline{W_T} = W_T - (S_T - K)^+. \tag{46}$$

Therefore, for the utility function $u(x) = \frac{x^\gamma}{\gamma}$ (with $0 < \gamma < 1$), the terminal condition is

$$U(T,S,W) = \frac{\Big(W - (S-K)^+\Big)^\gamma}{\gamma}. \tag{47}$$

4. **Dynamic Programming Recursion:** At an arbitrary node (t,S,W), the value function is computed by optimizing over the control π:

$$U(t,S,W) = \max_{\pi\in\Pi}\Big\{p\,U\Big(t+\Delta t, S\,u,\, W_{\text{up}}(\pi)\Big) + (1-p)\,U\Big(t+\Delta t, S\,d,\, W_{\text{down}}(\pi)\Big)\Big\}, \tag{48}$$

where the wealth updates for the up and down moves are:

$$W_{\text{up}}(\pi) = W\Big(1 + r\,\Delta t + \pi\Big[(\alpha - r)\Delta t + \sigma\sqrt{\Delta t}\Big]\Big), \tag{49}$$

$$W_{\text{down}}(\pi) = W\Big(1 + r\,\Delta t + \pi\Big[(\alpha - r)\Delta t - \sigma\sqrt{\Delta t}\Big]\Big). \tag{50}$$

5. **Backward Induction:** Starting from the terminal condition at $t = T$, we recursively compute the value function at each earlier time step. At every node, a discrete set of control values $\pi \in \Pi$ is tested. For each candidate π, the next-step wealth levels are computed and the corresponding U-values are interpolated (if necessary) on the wealth grid, since W_{up} and W_{down} might not belong in the wealth grid used to compute the U-values at the next step. The control that maximizes the expected utility is chosen and the value function is updated accordingly.

2.2 Detailed Algorithm

We now summarize the steps of the algorithm:

1. **Grid Setup:** Discretize the time interval $[0, T]$ into N_t steps with $\Delta t = T/N_t$, the stock price over a range $[S_{\min}, S_{\max}]$, and the wealth over a grid $[W_{\min}, W_{\max}]$.
2. **Terminal Condition:** For each terminal node at time T, where the stock price is

$$S_T = S_0\, u^i\, d^{N_t - i} \quad (i = 0, 1, \ldots, N_t), \tag{51}$$

and for each wealth value W, set

$$U(T, S_T, W) = \frac{\Big(W - (S_T - K)^+\Big)^\gamma}{\gamma}. \tag{52}$$

3. **Backward Induction:** For $n = N_t - 1, N_t - 2, \ldots, 0$ and for each node (characterized by the number of up moves i, with stock price $S = S_0\, u^i\, d^{n-i}$):
   (a) For each wealth grid point W, loop over the discrete set of candidate controls $\pi \in \Pi$.
   (b) Compute the next-period wealth in both the up and down scenarios:

$$W_{\text{up}}(\pi) = W\Big(1 + r\,\Delta t + \pi\Big[(\alpha - r)\Delta t + \sigma\sqrt{\Delta t}\Big]\Big), \tag{53}$$

$$W_{\text{down}}(\pi) = W\Big(1 + r\,\Delta t + \pi\Big[(\alpha - r)\Delta t - \sigma\sqrt{\Delta t}\Big]\Big). \tag{54}$$

   (c) For the up move, the next node is $(t + \Delta t, S\,u)$; for the down move, it is $(t + \Delta t, S\,d)$. Use linear interpolation on the wealth grid to approximate the value function at these nodes, i.e.,

$$U\Big(t + \Delta t, S\,u, W_{\text{up}}(\pi)\Big) \quad \text{and} \quad U\Big(t + \Delta t, S\,d, W_{\text{down}}(\pi)\Big). \tag{55}$$

   (d) Compute the expected utility for the control π:

$$V(\pi) = p\,U\Big(t + \Delta t, S\,u, W_{\text{up}}(\pi)\Big) + (1 - p)\,U\Big(t + \Delta t, S\,d, W_{\text{down}}(\pi)\Big). \tag{56}$$

   (e) Choose the control π^* that maximizes $V(\pi)$, and set

$$U(t, S, W) = \max_{\pi \in \Pi} V(\pi). \tag{57}$$

4. **Optimal Control Recovery:** At the end of the backward induction, the value function $U(0, S_0, W)$ at the root node (time 0, stock price S_0) is obtained. Also, by recording the control π^* that maximizes $V(\pi)$ at each node, we can recover the optimal investment strategy.

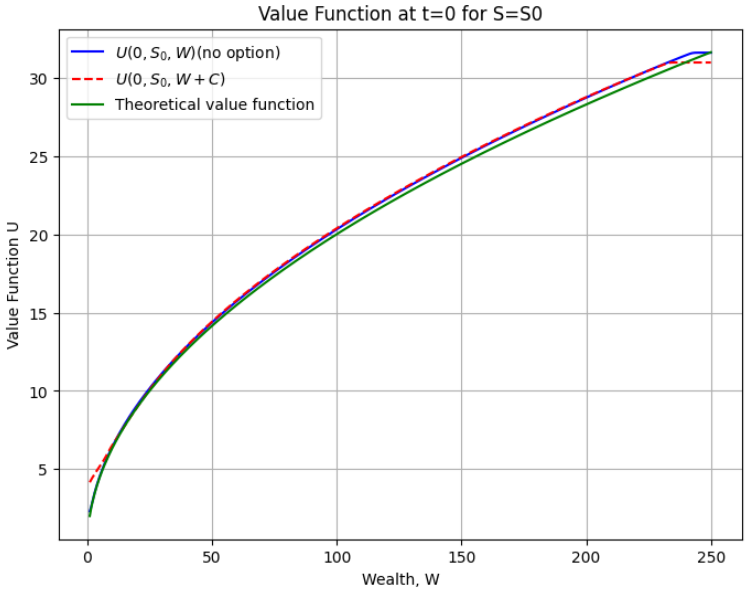
2.3 CARA Utility

The numerical scheme for the CARA Utility is essentially the same, but thanks to a trick explained in the next section related to the shape of this utility function, the price and optimal strategy are independent of the wealth, which allow us to greatly simplify the numerical scheme by eliminating the need for a grid of W values, the indirect utility function depending on W only by a multiplicationg factor.

2.4 Results

2.4.1 CRRA Utility

For the following parameters : $T = 1, S_0 = 100, \sigma = 0.2, \alpha = 0.05, r = 0.03, K = 100, \gamma = 0.5$, we observe that when adding the Black-Scholes price (C) to the initial wealth for the portfolio with a position in option, the curve of the indirect utility function at $t = 0$, with regards to the intial wealth, corresponds with the one with no option position, and the theoretical one.

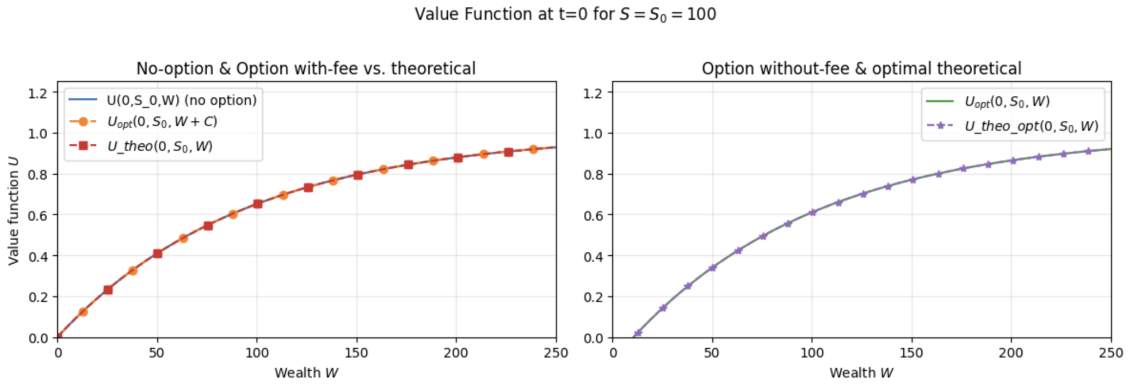


2.4.2 CARA Utility

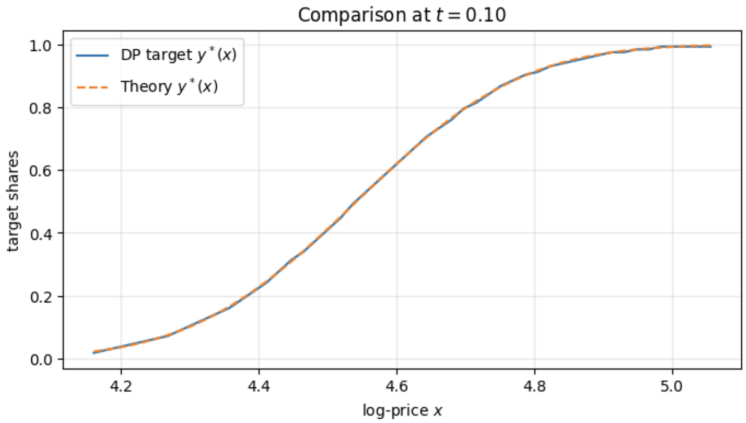
For the following parameters : T = 1, S0 = 100, $\sigma = 0.2$, $\alpha = 0.05$, r = 0.05, K = 100, $\gamma = 0.001$, the price obtained is :

$$C = 10,45$$

corresponding exactly with the Black Scholes price.
We observe that by adding to the initial wealth for the portfolio with a position in option C, the error between the value function when there is no option in the portfolio and the one with an option, is minimised (RMSE = 8,3e-16).
These two are also aligned with the theoretical value function in absence of option, with a RMSE of 0.0 between the theoretical value function and the one found with DP.
Finally, the value function found by DP when there is an option aligns perfectly (RMSE of 0.0001) with the theoretical one ($\overline{U}(t, W, S) = U(t, W - f)$) when the risk aversion coefficient γ is close to 0 (making the investor risk-neutral). This comes from the fact that the price without transaction costs diverges slightly from the Black-Scholes price as the investor becomes more risk averse. This is explained by the fact that Black-Scholes always assumes the investor as risk-neutral.



We can also observe that the optimal policy found by DP corresponds to the theoretical one, with a RMSE of approximately 0.001 across prices.



3 Optimal investment with transaction costs

We now extend the problem to account for proportional transaction costs. The investor can trade a risky asset S_t and a non-risky one B_t, and is charged a proportional transaction cost $\lambda > 0$ on both buying and selling. His holdings are represented by:

- y_t: quantity of risky asset held
- W_t: wealth held in the cash account

We model the system as follows:

$$dS_t = \alpha S_t dt + \sigma S_t dZ_t, \tag{58}$$
$$dy_t = l_t dt - m_t dt, \tag{59}$$
$$dW_t = rW_t dt - (1+\lambda) S_t l_t dt + (1-\lambda) S_t m_t dt, \tag{60}$$

where $l_t, m_t \geq 0$ are the purchases and sales of the risky asset, and λ is the transaction cost (assumed symmetric in our study).

Remark. When setting the transaction costs to 0, we observe that the portfolio has the following dynamics (total wealth being $\bar{W}_t = W_t + y_t S_t$) :

$$d\bar{W}_t = dW_t + dS_t y_t + dy_t S_t \tag{61}$$
$$= rW_t dt - S_t dy_t + y_t(\alpha S_t dt + \sigma S_t dZ_t) + S_t dy_t \tag{62}$$
$$= r(X_t - y_t S_t)dt + \alpha y_t S_t dt + \sigma y_t S_t dZ_t \tag{63}$$
$$= r\bar{W}_t dt + y_t S_t(\alpha - r)dt + y_t S_t \sigma dZ_t. \tag{64}$$

By setting $\pi_t = \frac{y_t S_t}{\bar{W}_t}$ (proportion of wealth in risky asset), we find the exact same expression as in Part 1, which confirms coherence between the formulations of both problems.

3.1 Optimal investment without options

The investor's objective is now to maximize expected utility of his terminal wealth including the liquidation value of his risky holdings:

$$\sup_{(l,m)} \mathbb{E}\left[u\left(W_T + y_T S_T - \lambda S_T |y_T|\right)\right]. \tag{65}$$

This leads us to define the indirect utility function (value function):

$$U(t,S,y,W) = \sup_{(l,m)} \mathbb{E}_{t,S,y,W}\left[u\left(W_T + y_T S_T - \lambda S_T |y_T|\right)\right]. \tag{66}$$

We apply Itô's lemma to the function $U(t, S_t, y_t, W_t)$, assuming regularity:

$$dU = \frac{\partial U}{\partial t}dt + \frac{\partial U}{\partial S}dS_t + \frac{\partial U}{\partial y}dy_t + \frac{\partial U}{\partial W}dW_t + \frac{1}{2}\frac{\partial^2 U}{\partial S^2}(dS_t)^2 \tag{67}$$
$$= \left[\frac{\partial U}{\partial t} + \alpha S\frac{\partial U}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 U}{\partial S^2} + rW\frac{\partial U}{\partial W}\right.$$
$$\left. + l\left(\frac{\partial U}{\partial y} - (1+\lambda)S\frac{\partial U}{\partial W}\right) + m\left(-\frac{\partial U}{\partial y} + (1-\lambda)S\frac{\partial U}{\partial W}\right)\right]dt + \sigma S\frac{\partial U}{\partial S}dZ_t. \tag{68}$$

Then, we use the same reasoning as in Section 1.2, taking the expectation and using the dynamic programming principle (and optimality), we obtain the HJB equation:

$$\boxed{\frac{\partial U}{\partial t} + \alpha S \frac{\partial U}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 U}{\partial S^2} + rW\frac{\partial U}{\partial W} + \max_{l\geq 0, m\geq 0}\left\{ l\left(\frac{\partial U}{\partial y} - (1+\lambda)S\frac{\partial U}{\partial W}\right) + m\left(-\frac{\partial U}{\partial y} + (1-\lambda)S\frac{\partial U}{\partial W}\right)\right\} = 0.} \tag{69}$$

From this equation, we can isolate the terms that drive the control l and m:

$$F(U) := \max\left\{\frac{\partial U}{\partial y} - (1+\lambda)S\frac{\partial U}{\partial W},\ -\frac{\partial U}{\partial y} + (1-\lambda)S\frac{\partial U}{\partial W}\right\}. \tag{70}$$

Then the problem can be reformulated as a variational inequality:

$$\max\left\{F(U),\ -\frac{\partial U}{\partial t} - \alpha S\frac{\partial U}{\partial S} - \frac{1}{2}\sigma^2 S^2\frac{\partial^2 U}{\partial S^2} - rW\frac{\partial U}{\partial W}\right\} = 0. \tag{71}$$

By setting :

$$A = \frac{\partial U}{\partial y} - (1+\lambda)S\frac{\partial U}{\partial W}(\text{Selling term}) \tag{72}$$

$$B = -\frac{\partial U}{\partial y} + (1-\lambda)S\frac{\partial U}{\partial W}(\text{Buying term}) \tag{73}$$

$$C = -\frac{\partial U}{\partial t} - \alpha S\frac{\partial U}{\partial S} - \frac{1}{2}\sigma^2 S^2\frac{\partial^2 U}{\partial S^2} - rW\frac{\partial U}{\partial W}(\text{No Transaction term}) \tag{74}$$

the Variational Inequality becomes :

$$\max\{A, B, C\} = 0 \tag{75}$$

This divides the (S, y, W) space in three regions, each characterized by one of the terms being active (=0).

No Transaction region : Defined by

$$(1-\lambda)S \leq \frac{\partial U/\partial y}{\partial U/\partial w} \leq (1+\lambda)S \tag{76}$$

$$-\frac{\partial U}{\partial t} - \alpha S\frac{\partial U}{\partial S} - \frac{1}{2}\sigma^2 S^2\frac{\partial^2 U}{\partial S^2} - rW\frac{\partial U}{\partial W} = 0 \tag{77}$$

Selling region : Defined by

$$A = 0 \tag{78}$$

$$B < 0 \tag{79}$$

$$C \leq 0 \tag{80}$$

Buying region : Defined by

$$A < 0 \tag{81}$$

$$B = 0 \tag{82}$$

$$C \leq 0 \tag{83}$$

The terminal condition reflects the full liquidated value of the portfolio at maturity:

$$U(T, S, y, W) = u\left(W + yS - \lambda S|y|\right). \tag{84}$$

3.2 When there is a short position in option

We now consider the case where the investor holds a short position in a contingent claim (e.g., an option) with a payoff function $\phi(S_T)$. This means that, in addition to managing their liquid and risky positions, the investor must deliver a terminal payoff equal to $\phi(S_T)$ at time T.

Modified objective. The optimization problem becomes:

$$\sup_{(l,m)} \mathbb{E}\left[u\left(W_T + y_T S_T - \lambda S_T |y_T| - \phi(S_T)\right)\right]. \tag{85}$$

This corresponds to an investor aiming to maximize expected utility after liquidating all risky holdings and delivering the payoff of the contingent liability.

Value function. The associated value function is:

$$U(t,S,y,W) = \sup_{(l,m)} \mathbb{E}_{t,S,y,W}\left[u\left(W_T + y_T S_T - \lambda S_T |y_T| - \phi(S_T)\right)\right]. \tag{86}$$

HJB and variational inequality. As before, the HJB equation and the variational inequality remain unchanged in structure since the dynamics of the controls and state variables are unaffected by the presence of the claim. The only modification lies in the terminal condition.

Terminal condition. The new terminal condition becomes:

$$U(T,S,y,W) = u\left(W + yS - \lambda S|y| - \phi(S)\right). \tag{87}$$

Remark. If the investor receives a premium p at time t for selling the claim, one may define an indifference price by comparing the value functions with and without the claim, and solving:

$$U^{\text{with claim}}(t,S,y,W+p) = U^{\text{without claim}}(t,S,y,W). \tag{88}$$

3.3 Dimension Reduction via CARA Utility

We will now study the problem when using the exponential utility, which will help us reduce the dimension of the problem, by factoring out the wealth of the maximization problem. This comes from the following reformulation (with $\delta(T,t) = e^{-r(T-t)}$) :

$$W_T = \frac{B_s}{\delta(T,s)} + \int_s^T \frac{1}{\delta(T,t)}((1-\lambda)S_t dL_t - (1+\lambda)S_t dM_t) \tag{89}$$

Therefore, we can write the maximization problem as :

$$V(t,B,y,S) = \max_{\pi} \mathbb{E}_{t,B,y,S}(u(W_T + y_T S_T - \lambda S_T |y_T| - \phi(S_T))) \tag{90}$$

$$= 1 - e^{-\gamma \frac{B}{\delta(T,t)}} Q(t,y,S) \tag{91}$$

with :

$$Q(t,y,S) = \min_{\pi} \mathbb{E}_{t,y,S}(e^{-\gamma \int_t^T \frac{1}{\delta(T,s)}((1-\lambda)S_s dL_s - (1+\lambda)S_s dM_s)} e^{-\gamma(y_T S_T - \lambda S_T |y_T| - \phi(S_T))}) \tag{92}$$

$$Q(t,y,S) = 1 - V(t,0,y,S) \tag{93}$$

Therefore, we can transform the variational inequality we obtained for U, into the following p.d.e for $Q(t,y,S)$:

$$\min\{Q_y + \frac{\gamma(1+\lambda)S}{\delta(T,t)}Q, -(Q_y + \frac{\gamma(1-\lambda)S}{\delta(T,t)}Q), Q_t + \alpha S Q_S + \frac{1}{2}\sigma^2 S^2 Q_{SS}\} = 0 \tag{94}$$

We write this using the previous formula and noticing that :

- $V_B = 0$ ($Q_B = 0$)
- $V_y = -e^{-\gamma B/\delta} Q_y$
- $V_S = -e^{-\gamma B/\delta} Q_S$
- $V_{SS} = -e^{-\gamma B/\delta} Q_{SS}$
- $V_t = -e^{-\gamma B/\delta}(\frac{\gamma r B}{\delta} Q - Q_t)$

The boundary condition is :

$$Q(T, y, S) = \exp(-\gamma(y_T S_T - \lambda S_T |y_T| + \phi(S_T))) \tag{95}$$

(For the case without the option, only the boundary condition changes and becomes $Q(T, y, S) = \exp(-\gamma((1-\lambda) y_T S_T)$.

This transformation allows us to solve for Q, which doesn't depend on the wealth level, and then simply multiply by a quantity depending only on the initial wealth B to get V, instead of solving for V directly, allowing for a much simpler numerical scheme.

3.4 Indifference Pricing under CARA Utility

An important feature of the CARA utility function $u(x) = 1 - \exp(-\gamma x)$ is that it allows for an explicit expression of the indifference price of a contingent claim. In particular, the **utility indifference price** of an option can be computed directly without solving the full HJB equation twice — once with the option and once without — as is required for general utility functions.

Let us denote by $Q(t, y, S)$ the function introduced previously such that:

$$V(t, B, y, S) = 1 - \exp\left(-\gamma \frac{B}{\delta(T, t)}\right) Q(t, y, S)$$

where $\delta(T, t) = e^{-r(T-t)}$.

Then, the **buyer's price** p^b and **writer's price** p^w at time t are given by:

$$p^b(t, y, S) = \frac{\delta(t_0, T)}{\gamma} \log Q^{\text{without option}}(t, y, S) - \frac{1}{\gamma} \log Q^{\text{with option, buyer}}(t, y, S)$$

$$p^w(t, y, S) = \frac{\delta(t_0, T)}{\gamma} \log Q^{\text{with option, writer}}(t, y, S) - \frac{1}{\gamma} \log Q^{\text{without option}}(t, y, S)$$

This property arises due to the **translation-invariance and exponential form** of the CARA utility, which allows factoring out the dependence on initial wealth B. Hence, the pricing depends only on the marginal cost in utility space, captured by the difference in $\log Q$ terms.

This result is powerful: instead of solving two PDEs for the full utility function U, one only needs to solve the reduced problem for $Q(t, y, S)$ — once with the option and once without — and the price follows directly. This makes exponential utility particularly tractable for utility-based pricing with transaction costs.

Remark. In the presence of transaction costs, the utility indifference price depends on the position taken by the investor. This asymmetry stems from the fact that buying and selling the contingent claim lead to different terminal liabilities and trading incentives.

Therefore, we must distinguish between two distinct indifference prices:

- The **buyer's price**, computed by comparing the value function when *purchasing* the option versus not holding it.
- The **writer's price**, computed by comparing the value function when *writing* (selling) the option versus not holding it.

These two prices are generally *not equal* when transaction costs are present, due to the non-linearity and irreversibility of trading costs. In particular, the option writer must hedge a liability, possibly incurring high transaction costs to maintain replication, whereas the buyer holds a passive position and faces a different risk profile. This creates a *bid-ask spread* in utility-based pricing, even in the absence of market illiquidity or model uncertainty.

3.5 Derivation of the Indifference Pricing Formula under CARA Utility

The principle behind utility indifference pricing is the following: the price of a contingent claim (e.g., an option) is the amount of cash that makes the investor indifferent, in terms of expected utility, between holding the claim and not holding it.

Let us consider a risk-averse investor with exponential utility function

$$u(x) = 1 - \exp(-\gamma x)$$

and initial cash wealth B at time t. We denote by $V^{\text{with}}(t, B, y, S)$ and $V^{\text{without}}(t, B, y, S)$ the value functions corresponding respectively to the case where the investor holds a contingent claim with terminal payoff $\phi(S_T)$, and the case where they do not.

The **indifference price** p is defined by the equation:

$$V^{\text{with}}(t, B + p, y, S) = V^{\text{without}}(t, B, y, S)$$

Thanks to the structure of the exponential utility, and assuming the price dynamics and trading constraints are the same in both cases, we can factor out the dependence on B using the change of variables introduced earlier:

$$V(t, B, y, S) = 1 - \exp\left(-\gamma \frac{B}{\delta(T, t)}\right) Q(t, y, S)$$

with $\delta(T, t) = e^{-r(T-t)}$, and Q solving a reduced HJB variational inequality independent of B.

Plugging this expression into the indifference pricing identity, we get:

$$1 - \exp\left(-\gamma \frac{B + p}{\delta(T, t)}\right) Q^{\text{with, writer}}(t, y, S) = 1 - \exp\left(-\gamma \frac{B}{\delta(T, t)}\right) Q^{\text{without}}(t, y, S)$$

Canceling the constants and solving for p, we obtain

$$p = \frac{\delta(t_0, T)}{\gamma} \left(\log(Q^{\text{without}}(t, y, S)) - \log(Q^{\text{with, writer}}(t, y, S))\right)$$

This expression directly links the price of the option to the difference in utility between the two control problems (the derivation is the same for the buyer price, except that the initial wealth is $B - p$ and we look at $Q^{with,buyer}$).

4 Numerical Scheme for the Problem with Transaction Costs and Option Position

In this section, we present a numerical scheme to compute the indirect utility function $U(t, S, y, W)$ in the case where the investor is subject to proportional transaction costs and holds a short position in a European option. We recall that the effective terminal wealth accounts for the liquidation value of the remaining position in the risky asset:

$$\bar{W}_T = W_T + S_T y_T - \lambda S_T |y_T|.$$

As a result, the terminal condition becomes:

$$U(T, S, y, W) = u(W + Sy - \lambda S|y|).$$

4.1 Discretization of the State Space

We discretize the continuous variables t, S, y, and W as follows:

- The time interval $[0, T]$ is divided into N steps of size $\Delta t = T/N$.
- The stock price S_t is approximated by a binomial process:

$$S(i+1) = \begin{cases} S(i) \cdot e^{\alpha \Delta t + \sigma\sqrt{\Delta t}} & \text{with probability } 1/2, \\ S(i) \cdot e^{\alpha \Delta t - \sigma\sqrt{\Delta t}} & \text{with probability } 1/2. \end{cases}$$

- The number of shares held y is discretized in steps of Δy:

$$y_k = k\Delta y, \quad k \in \{-M, ..., M\}.$$

- The cash wealth W is discretized in steps of ΔW:

$$W_l = l\Delta W, \quad l \in \{0, ..., K\}.$$

We denote the discretized utility value by:

$$u(i, j, k, l) \approx U(i\Delta t, S_j, y_k, W_l).$$

4.2 Terminal Condition

At maturity $t = T$, we initialize:

$$u(N, j, k, l) = u\left(W_l + S_j y_k - \lambda S_j |y_k|\right).$$

4.3 Backward Recursion over Time Steps

For $i = N - 1$ down to 0, we compute $u(i, j, k, l)$ at each grid point by considering the three possible actions: do nothing, buy the minimal allowed number of shares, or sell the minimal allowed number of shares.

1. **No Transaction (NT):**
   If no transaction occurs, wealth grows at the risk-free rate:

$$W' = W_l \cdot e^{r\Delta t}.$$

   Let l^+ be the index closest to W'. The continuation value is:

$$u^{\mathrm{NT}}(i, j, k, l) = \frac{1}{2}\left[u(i+1, j+1, k, l^+) + u(i+1, j-1, k, l^+)\right].$$

2. **Buy Action:**
   The investor buys Δy shares:

$$y' = y_k + \Delta y, \quad W' = W_l - (1+\lambda) S_j \Delta y.$$

   Let l_b be the nearest index to W', then:

$$u^{\text{Buy}}(i,j,k,l) = u(i,j,k+1,l_b).$$

3. **Sell Action:**
   The investor sells Δy shares:

$$y' = y_k - \Delta y, \quad W' = W_l + (1-\lambda) S_j \Delta y.$$

   Let l_s be the nearest index to W', then:

$$u^{\text{Sell}}(i,j,k,l) = u(i,j,k-1,l_s).$$

4. **Value and Optimal Action:**
   The optimal decision is to take the maximum value among the three possibilities:

$$u(i,j,k,l) = \max\left\{u^{\text{NT}}(i,j,k,l), u^{\text{Buy}}(i,j,k,l), u^{\text{Sell}}(i,j,k,l)\right\}.$$

   The action that maximizes $u(i,j,k,l)$ determines whether the point lies in region $\mathcal{NT}$, $\mathcal{B}$ (buy), or $\mathcal{S}$ (sell).

4.4 Implementation Notes

- In practice, we use interpolation when W' does not exactly fall on a mesh point.
- The scheme tests only minimal trade increments ($\pm\Delta y$), rather than computing the exact trade required to re-enter $\mathcal{NT}$.
- Nevertheless, this recursive maximization allows us to capture the optimal control logic implied by the variational inequality.

4.5 CARA Utility

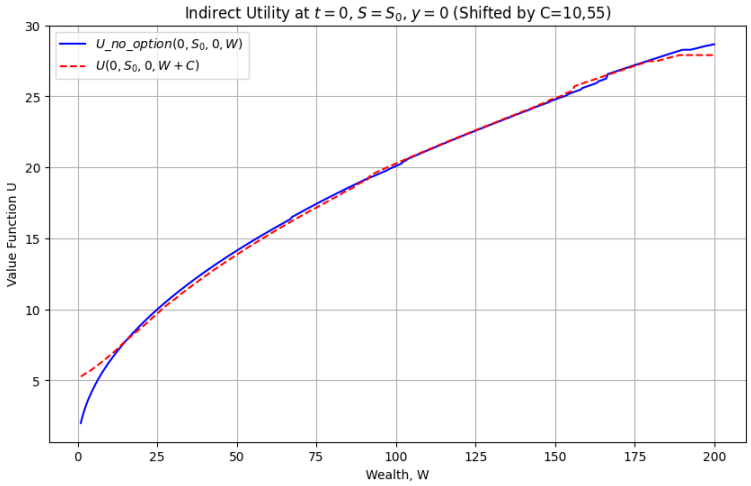
The numerical scheme for the CARA Utility is essentially the same, but thanks to the trick we used in the previous section, the price and optimal strategy are independent of the wealth. The Dynamic Programming scheme minimizes $Q(t,y,S)$ for the three possible actions, at each step, which doesn't depend on W, therefore there is no need to create a grid for W.
The indirect utility function does depend on W, but only by a multiplication factor once we have found Q, we have :

$$U(0,W,y,S) = 1 - e^{-\frac{\gamma W}{\delta(T,0)}} Q(0,y,S) \tag{96}$$

4.6 Results

4.6.1 CRRA Utility

For the following parameters : $T = 1, S_0 = 100, \sigma = 0.2, \alpha = 0.05, r = 0.03, K = 100, \gamma = 0.5, \lambda = 0.01$, we observe that by adding to the initial wealth for the portfolio with a position in option $C = 11$, the error between the value function when there is no option in the portfolio and the one with an option, is minimised. This relates to the indifference pricing method.



4.6.2 CARA Utility

For the following parameters : $T = 1, S_0 = 100, \sigma = 0.2, \alpha = 0.05, r = 0.05, K = 100, \gamma = 0.5$ and $\lambda = 0.01$ (transaction costs), the price of the option is :

$$C = 14.65$$

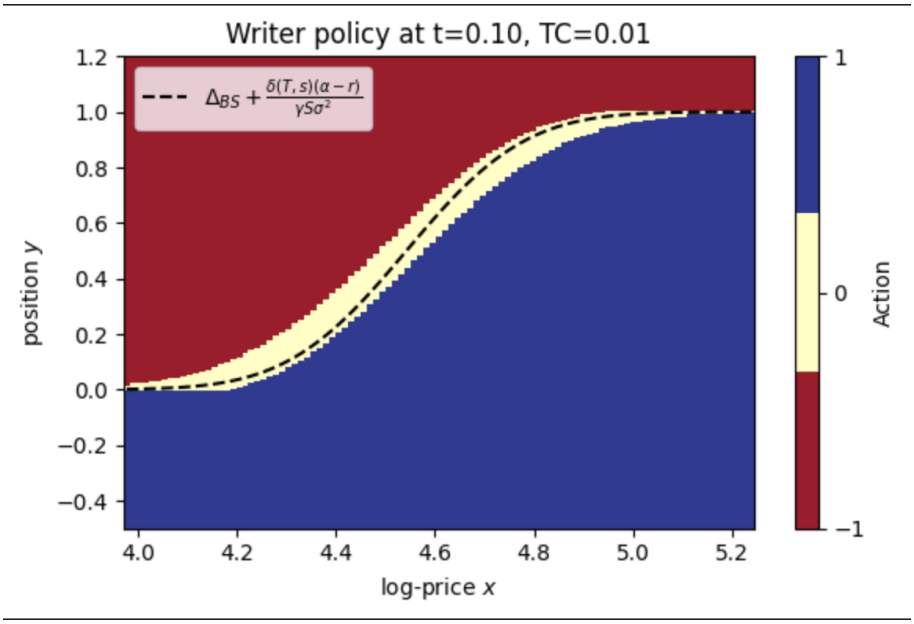
which gives a 40% increase compared to the price without transaction costs.
To compare the way the optimal strategies differ with and without transaction costs, we introduce the following metrics :

- Average shares moved by node (AS) $= \frac{2}{(N+1)(N+2)} \sum_{t=1}^{N} \sum_{i=0}^{t-1} \sum_{j=i,i+1} |y(t,j) - y(t-1,i)|$
- Trade frequency (TF) $= \frac{2}{(N+1)(N+2)} \sum_{t=1}^{N} \sum_{i=0}^{t-1} \sum_{j=i,i+1} \mathbb{1}_{|y(t,j)-y(t-1,i)|>0}$

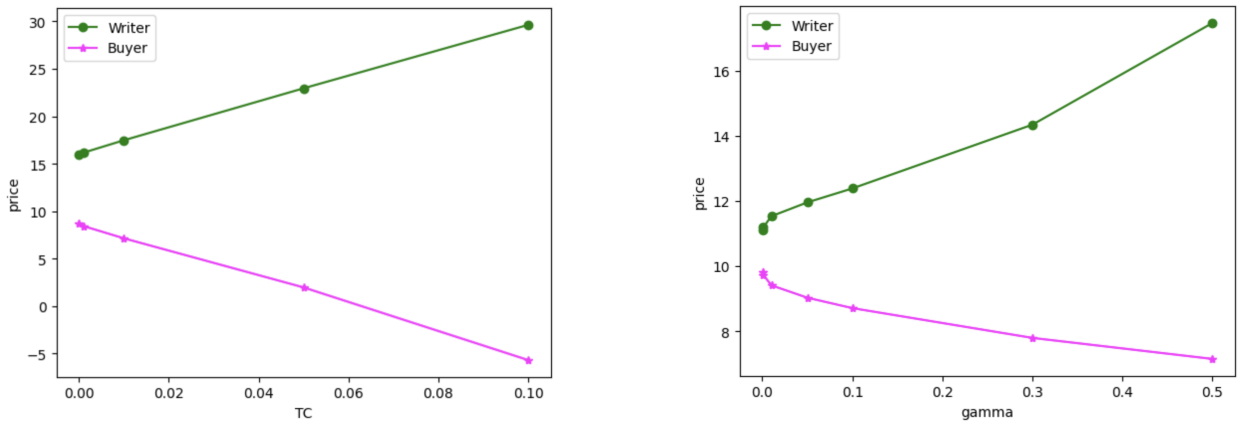
We observe that without transaction costs, TF = 99.7% and AS = 0.009 (we only allow moves of $\pm\Delta_y$, which are fairly small), while for transaction costs of 1%, TF = 75% and AS = 0.0047.
Below is a table comparing price, TF, AS for different strikes, risk aversion, TC levels.

	Gamma	Strike	TC (%)	TC Price	BS Price	RelDiff (%)	Intensity	AvgTrades
0	0.5000	80	0.0	33.231924	24.588835	35.150460	0.007630	0.853091
1	0.5000	80	1.0	34.458102	24.588835	40.137184	0.007573	0.846693
2	0.5000	80	5.0	39.353441	24.588835	60.045974	0.007507	0.839296
3	0.5000	100	0.0	14.973899	10.450584	43.282895	0.007639	0.854102
4	0.5000	100	1.0	16.507851	10.450584	57.961043	0.007585	0.848068
5	0.5000	100	5.0	21.996888	10.450584	110.484785	0.007521	0.840913
6	0.5000	120	0.0	3.801460	3.247477	17.058870	0.007648	0.855107
7	0.5000	120	1.0	5.371964	3.247477	65.419593	0.007598	0.849462
8	0.5000	120	5.0	10.458447	3.247477	222.048332	0.007536	0.842557
9	0.0001	80	0.0	24.589463	24.588835	0.002552	0.008921	0.997434
10	0.0001	80	1.0	25.534968	24.588835	3.847814	0.008496	0.949914
11	0.0001	80	5.0	29.321968	24.588835	19.249112	0.008443	0.943917
12	0.0001	100	0.0	10.453841	10.450584	0.031169	0.008921	0.997435
13	0.0001	100	1.0	11.108797	10.450584	6.298341	0.008494	0.949704
14	0.0001	100	5.0	13.874412	10.450584	32.762082	0.008438	0.943349
15	0.0001	120	0.0	3.250088	3.247477	0.080381	0.008921	0.997435
16	0.0001	120	1.0	3.544063	3.247477	9.132810	0.008501	0.950470
17	0.0001	120	5.0	4.908421	3.247477	51.145655	0.008442	0.943795

Now, one of the most interesting results of this scheme, was the clear apparition of the three regions mentioned in the previous section. Indeed, a 'No-Trade Region' appears in the optimal trading strategy around the theoretical optimal trading strategy without transaction costs, with a 'Buy Region' below and a 'Sell Region' above.



Finally, another interesting observation is the increase of the 'Writer-Buyer' window with regards to transaction costs and risk aversion of the investor, as exposed below.



(a) Sensitivity to Transaction Costs

(b) Sensitivity to risk aversion

Figure 1: Writer-Buyer price sensitivity

5 Bull Callspread Pricing and Hedging

In this section, we study the application of our model to a bull call spread to highlight the fact that linearity in price and hedging strategies disappear for products like call spreads once transaction costs are introduced.

5.1 Without transaction costs

Reminder : A Bull Call Spread is a strategy which consists of buying a call option at a certain strike, while simultaneously selling a call option from the same underlying with the same terminal date, with a higher strike. This allows to benefit from a moderate rise in the underlying asset, while paying a lower price than for the single call.

In the absence of transaction costs and for a risk neutral investor, the price of the callspread is equal to buying and selling the two calls separately, we have :

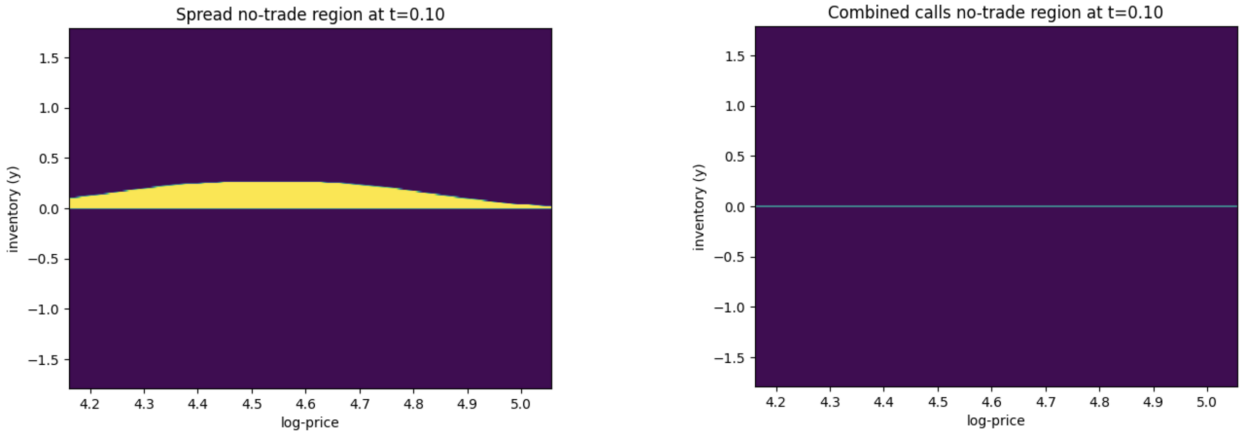
$$\boxed{p_{CS} = p_{K_1} - p_{K_2}} \tag{97}$$

Indeed, using our model with the following parameters : $S_0 = 100, r = \mu = 0.05, \sigma = 0.2, T = 1, K_1 = 95, K_2 = 105, \gamma = 0.0001, c = 0$, we have :

- $p_{CS} = 5.33$
- $p_{K_1} - p_{K_2} = 5.325$

5.2 With transaction costs

Let us now introduce proportional transaction costs and observe how the price and strategy evolve.
Indeed, once transaction costs are introduced, hedging the two calls separately becomes really expensive as it will sometimes require to buy and sell shares of the underlying at the same time, involving twice the necessary transaction costs (the no-trade region will be reduced to the intersection of both no-trade region, which is almost inexistent), which can be avoided by treating the call spread as a single product which we will hedge using our dynammic programming model.

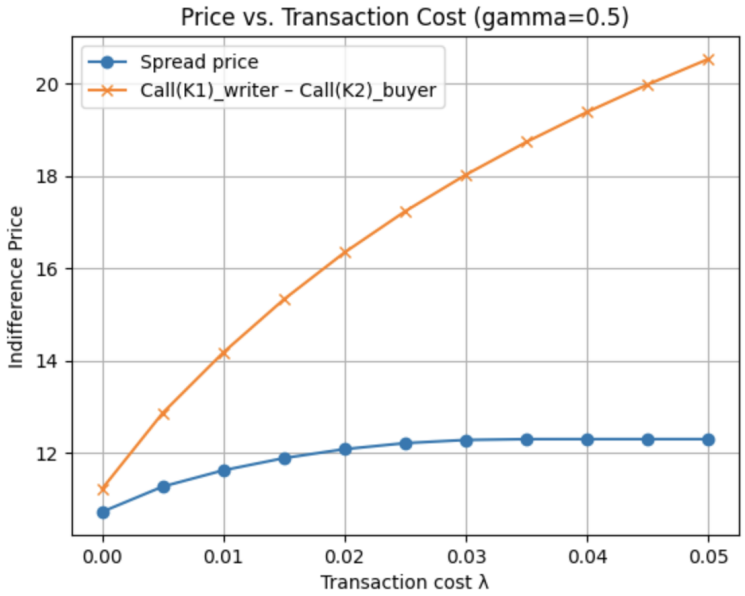


(a) No-trade region for the Spread

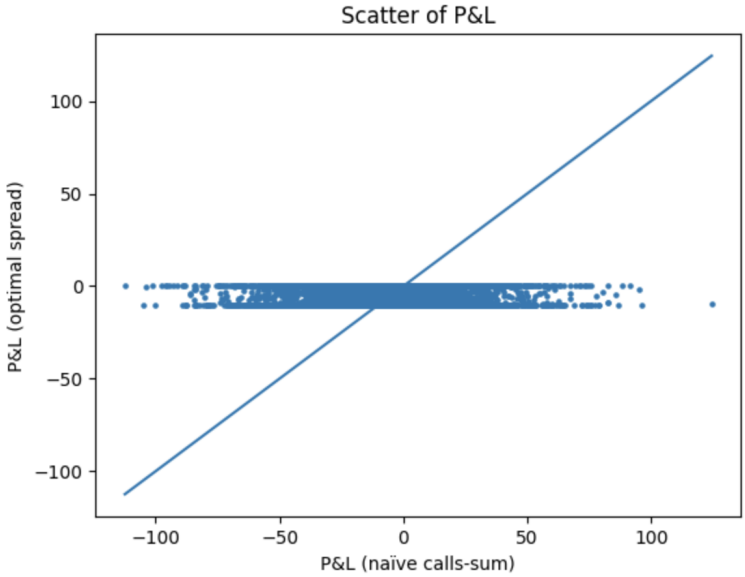
(b) Combined No-trade region

Figure 2: Difference in Hedging strategies

As the price is determined as the minimal amount necessary to hedge the product, this non linearity in hedging strategies as transaction costs will impact the price : the price of the call spread will no longer be equal to the difference of the two call prices for non-zero transaction costs.



All this goes to show the importance treating carefully the questions arising from the introduction of transaction costs in model, as assumptions made before may not hold once some hypothesis are made, which can lead to inefficient hedging and pricing of instruments, like exposed in the figure below.



6 Deep Hedging

An increasingly appealing alternative to the traditional stochastic control framework is the use of deep hedging techniques [1]. These methods, rooted in reinforcement learning and neural networks, offer a model-free approach that relies solely on historical or simulated data, circumventing the need for a fully specified and tractable market model. This is particularly advantageous when dealing with transaction costs, liquidity constraints, or other frictions that render classical models analytically intractable or computationally burdensome. In such settings, the high-dimensional optimization problems and complex dynamics involved can make the traditional approach impractical, especially when considering path-dependent features or portfolio-level constraints. Deep hedging, by contrast, can flexibly accommodate these complexities, learning optimal hedging strategies directly from data while capturing realistic market behaviors that are difficult to encode in closed-form models. This data-driven paradigm shift represents a promising direction for practical risk management under market imperfections.
It is worth emphasizing that the first part of this paper, which relies on stochastic control and dynamic programming, plays a foundational role in informing the deep hedging framework introduced here. While deep learning methods offer a flexible, data-driven alternative, they are often perceived as black-box techniques. By first solving the problem analytically under a classical framework, we gain a much deeper understanding of the mathematical structure of the hedging problem — including the nature of optimal strategies in the presence of transaction costs, such as the no-transaction region. This insight proves especially valuable when designing network architectures, like the No Transaction Band Net, which explicitly encode such theoretical results to guide learning and improve training efficiency. Furthermore, the stochastic control framework makes explicit the fundamental connection between hedging and pricing via utility indifference pricing: once the optimal hedging strategy is derived, the option price emerges naturally as the amount that compensates for the residual risk. This clean, principled relationship is rarely made explicit in deep hedging, but it serves as a conceptual backbone for interpreting the results of data-driven approaches. Hence, the classical analysis is not only of pedagogical value but also enhances the practical implementation and understanding of deep learning-based methods.

6.1 Problem setting

The deep hedging framework retains the same overall objective as in the classical stochastic control approach: to maximize the expected utility of terminal wealth for a risk-averse investor by optimally controlling the positions taken in the underlying asset of the option. However, the key difference lies in the methodology. Instead of attempting to solve the optimization problem analytically or through dynamic programming — which quickly becomes intractable in the presence of transaction costs, path dependencies, or high-dimensional state spaces — deep hedging leverages neural networks to learn optimal trading strategies. Specifically, a neural network is trained to select actions (i.e., trading decisions) that maximize the expected utility of the investor (either directly, or indirectly by minimizing a symmetric criterion such as an entropic risk measure), using data-driven feedback through simulated trajectories. This approach bypasses the need for explicit model assumptions and allows for the incorporation of realistic market features that are otherwise difficult to handle in traditional frameworks.

Remark. As stated, the objective is not exactly the same in this part. Indeed, we decide to minimize the Entropic Risk Measure to train our network, rather than maximizing the exponential utility.
We observe that these two are equivalent.
Indeed, the Entropic Risk Measure is as follows :

$$\rho(X) = \frac{1}{\gamma} \log \mathbb{E}(\exp(-\gamma X)) \tag{98}$$

It is obvious that minimizing this creterion is equivalent to maximizing the expectation of the following utility :

$$U(X) = -\exp(-\gamma X) \tag{99}$$

We make this choice because the Entropic Risk Measure is a convex, coherent risk measure which offers better numerical stability, optimization properties, and interpretability — all while remaining equivalent in terms of the optimal strategy under exponential utility preferences.

6.1.1 From Optimal Control to Neural Networks

In the classical stochastic control setting, the investor aims to maximize the expected utility of terminal wealth by directly controlling the trading strategy y_t at each time step. Mathematically, this corresponds to solving:

$$\max_{(y_t)_{t\leq T}} \mathbb{E}\left[u(W_T)\right] \quad \text{or} \quad \min_{(y_t)_{t\leq T}} \rho(W_T)$$

where ρ is a risk measure such as the entropic risk.

In Deep Hedging, we approximate this control process by replacing the sequence of control decisions y_t with the output of a neural network. The network takes in a feature vector I_t (e.g., log-moneyness, time to maturity, past trades) and produces the hedge $y_t = \pi_\theta(I_t)$, where θ denotes the set of network parameters (weights and biases).

The optimization then becomes:

$$\min_\theta \ \frac{1}{\gamma} \log \mathbb{E}\left[\exp(-\gamma W_T^\theta)\right]$$

This objective is with respect to θ, and can be minimized using *stochastic gradient descent* or its variants (e.g., Adam optimizer), updating θ iteratively based on simulated trajectories.

Neural Network Structure. In the deep hedging framework, neural networks are employed as function approximators to learn optimal trading strategies directly from data. A neural network can be viewed as a parametric function $f_\theta : \mathbb{R}^d \to \mathbb{R}$, defined by the composition of several affine transformations and nonlinear activation functions. Its structure is defined layer by layer as follows:

$$x^{(l+1)} = \sigma\left(W^{(l)} x^{(l)} + b^{(l)}\right), \quad l = 0, \ldots, L-1$$

where:

- $x^{(0)} = I_t \in \mathbb{R}^d$ is the input vector (e.g., log-moneyness, time to maturity, volatility, etc.),
- $W^{(l)} \in \mathbb{R}^{n_{l+1} \times n_l}$ are the weight matrices of layer l,
- $b^{(l)} \in \mathbb{R}^{n_{l+1}}$ are the associated bias vectors,
- σ is a non-linear activation function applied element-wise (typically the ReLU: $\sigma(x) = \max(0, x)$),
- $x^{(l)}$ denotes the intermediate representation (feature map) at layer l,
- L is the total number of layers.

The final layer typically applies no activation function and outputs a scalar value, corresponding to the control y_t, i.e., the number of shares of the risky asset to hold at time t.

The activation function σ plays a crucial role in enabling the network to model non-linear relationships. Without it, the entire network would be equivalent to a single affine map, regardless of depth. The non-linearity introduced by σ allows the network to approximate complex decision surfaces — especially important in hedging tasks where market frictions, payoff functions, or utility curvature introduce strong non-linearities.

Once the network architecture is defined, its parameters $\theta = \{W^{(l)}, b^{(l)}\}_{l=0}^{L-1}$ are optimized using stochastic gradient descent. The training objective is to minimize a suitable loss function — here, the entropic risk measure:

$$L(\theta) = \frac{1}{\gamma} \log\left(\frac{1}{N} \sum_{i=1}^{N} \exp(-\gamma W_T^{(i)}(\theta))\right)$$

where $W_T^{(i)}(\theta)$ is the terminal P&L from the hedging strategy produced by the network on the i^{th} simulated price path.

The gradients of the loss with respect to the parameters θ are computed via **backpropagation**, an efficient application of the chain rule through the network layers. This allows the use of first-order optimization methods such as Adam to update the parameters and iteratively reduce the loss. At each training iteration:

1. A batch of simulated asset paths is generated;
2. The network outputs a hedge strategy $\pi_\theta(I_t)$ for each time step;
3. The terminal wealth is computed using the corresponding P&L formula with transaction costs;
4. The loss is evaluated using the entropic risk;
5. The parameters θ are updated in the direction of the negative gradient of the loss.

This procedure enables the neural network to learn an approximately optimal policy that maximizes expected utility (or minimizes risk) across market scenarios, without requiring explicit solutions of dynamic programming equations.

6.2 Network Architecture

6.2.1 Multi Layer Perceptron

The first architecture we consider is a standard feedforward neural network, commonly known as a **Multi-Layer Perceptron (MLP)**. This type of network is widely used for function approximation tasks due to its simplicity, expressiveness, and ease of implementation. In the context of deep hedging, our goal is to approximate the mapping between the current market state and the optimal hedging action — that is, the number of risky assets to hold at each time step.

At each trading time t, the network receives as input a feature vector $I_t \in \mathbb{R}^d$, which includes:

- the log-moneyness: $\log(S_t/K)$,
- the time to maturity: $T - t$,
- the volatility estimate σ,
- the previous hedge position y_{t-1}.

The MLP consists of a stack of fully connected layers, also called dense layers. The architecture used here is composed of:

- an input layer of size d,
- **five hidden layers**, each with $n = 32$ neurons,
- ReLU activation functions between each layer: $\sigma(x) = \max(0, x)$,
- a final linear output layer with one neuron, producing the scalar hedge decision $y_t \in \mathbb{R}$.

Formally, the network computes the output as follows:

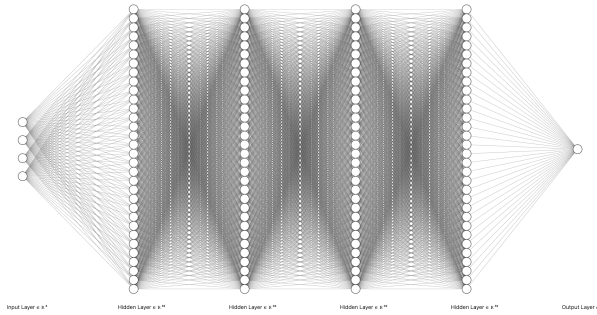
$$\begin{aligned} x^{(0)} &= I_t, \\ x^{(l+1)} &= \text{ReLU}(W^{(l)}x^{(l)} + b^{(l)}), \quad l = 0, \dots, 3, \\ x^{(5)} &= \text{ReLU}(W^{(4)}x^{(4)} + b^{(4)}), \\ y_t &= W^{(5)}x^{(5)} + b^{(5)}. \end{aligned}$$

The parameter set $\theta = \{W^{(l)}, b^{(l)}\}_{l=0}^{5}$ is optimized using stochastic gradient descent by minimizing the entropic risk loss over simulated market scenarios.

This architecture was chosen as a **baseline model** for its ability to capture moderately complex non-linear relationships between market features and hedging decisions. It is similar in spirit to the architecture proposed in [1], and provides a solid benchmark for evaluating more specialized architectures. The use of ReLU activations preserves gradient flow during training while introducing the necessary non-linear capacity to model frictions such as transaction costs.

However, a key limitation of this architecture is its **action-dependence**: the output y_t depends explicitly on the previous action y_{t-1}, which is fed as an input. This dependence increases the complexity of the optimization landscape, since the hedging trajectory becomes path-dependent. As a result, the MLP must learn both *when to act* and *how to act*, which may lead to instability and slow convergence during training.

Despite this, the MLP remains a useful benchmark model, offering valuable insights into the behavior of learned strategies in presence of market frictions.



6.2.2 No Transaction Band Net

As discussed previously, the main limitation of the classical MLP architecture lies in its **action-dependence**: the hedging decision at time t, denoted y_t, depends on the previous hedge y_{t-1}, which is part of the input. This makes the training process unstable, as the network must simultaneously learn *how to hedge* and *when to rebalance*, introducing complex temporal dependencies into the learning problem.

To mitigate this issue, we propose a second architecture inspired by theoretical results obtained in the first part of this work. Under exponential utility and proportional transaction costs, the solution of the dynamic control problem admits a well-known structure: the existence of a **no-transaction region**, defined by two state-dependent boundaries $b_\ell(t, S), b_u(t, S)$, such that:

$$\text{if } y_t \in [b_\ell(t, S_t), b_u(t, S_t)] \quad \Rightarrow \quad \text{no rebalancing is optimal.}$$

This region arises naturally from the variational inequality formulation of the Hamilton-Jacobi-Bellman (HJB) equation (cf. equation (75)), and characterizes zones where the marginal utility loss due to suboptimal allocation is smaller than the transaction cost penalty.

Network Design. Rather than predicting directly the hedge y_t, the No Transaction Band Net (NTBN) predicts two values $(b_\ell, b_u) \in \mathbb{R}^2$, representing the lower and upper bounds of the no-transaction band at each time step. The hedge is then computed via a **clamping operation** applied to the previous position:

$$y_t = \text{clamp}(y_{t-1}, b_\ell, b_u) = \begin{cases} b_\ell, & \text{if } y_{t-1} < b_\ell, \\ y_{t-1}, & \text{if } y_{t-1} \in [b_\ell, b_u], \\ b_u, & \text{if } y_{t-1} > b_u. \end{cases}$$

In this way, the network learns *when to act* (i.e., when to rebalance) rather than *how to act*, which substantially reduces the complexity of the optimization landscape. The architecture is thus more aligned with the theoretical structure of the optimal policy in the presence of transaction costs.

Architecture Details. The NTBN retains the same internal structure as the MLP: five hidden layers with 32 neurons each, and ReLU activations. However, it differs in two key aspects:

- The previous hedge y_{t-1} is *not* part of the input — the network only receives market features $I_t \in \mathbb{R}^d$, such as log-moneyness, time to maturity, and volatility.
- The final layer has two outputs: (b_ℓ, b_u), which define the no-transaction interval.

Formally, the network computes:

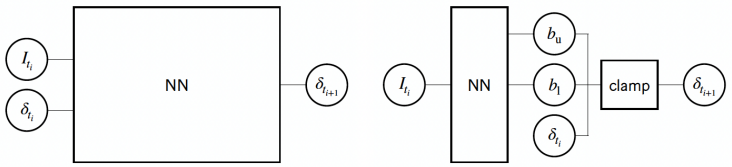
$$\begin{aligned} x^{(0)} &= I_t, \\ x^{(l+1)} &= \mathrm{ReLU}(W^{(l)} x^{(l)} + b^{(l)}), \quad l = 0, \dots, 4, \\ (b_\ell, b_u) &= W^{(5)} x^{(5)} + b^{(5)}, \\ y_t &= \mathrm{clamp}(y_{t-1}, b_\ell, b_u). \end{aligned}$$

Note that the clamping operation is non-differentiable with respect to y_{t-1}, but this poses no issue for backpropagation since y_{t-1} is treated as a constant during the forward pass of each batch.

Motivation and Advantages. By explicitly modeling the no-transaction band, the NTBN:

- removes the need for the network to infer action-thresholds implicitly;
- improves convergence and training stability;
- reduces variance in gradient estimates during backpropagation;
- exhibits better risk-adjusted performance in high-friction regimes;
- aligns closely with theoretical results from stochastic control.

This architecture can thus be interpreted as a hybrid between model-based and model-free methods: it uses neural networks to learn from data, but its structure is guided by analytical insights from the control-theoretic solution. As a result, the NTBN shows improved robustness and interpretability compared to unconstrained architectures.



(a) Feed-forward network. (b) No-transaction band network.

6.3 Algorithm

Training the model follows the standard deep hedging pipeline, which consists in simulating price trajectories, applying the neural network-based hedging strategy, computing the terminal P&L including transaction costs, and updating the network parameters to minimize a chosen loss function — here, the entropic risk measure.
For simplicity reasons and since computations are essentially the same, we assume $r = 0$ in this part.

Step 1: Simulate asset price paths. Generate a batch of N Monte Carlo trajectories $\{S_t^{(i)}\}_{t \le T}$ of the underlying asset, either under the historical or risk-neutral measure. These paths serve as input data for the training procedure.

Step 2: Apply the hedging policy. For each simulated path and each rebalancing date, compute the hedge $y_t^{(i)}$ using the neural network policy NN_θ. Depending on the architecture:

- The **MLP** directly outputs $y_t = NN_\theta(I_t, y_{t-1})$.
- The **No Transaction Band Net** outputs a pair $(b_\ell, b_u) = NN_\theta(I_t)$ and computes $y_t = \mathrm{clamp}(y_{t-1}, b_\ell, b_u)$.

Here, I_t denotes the set of features used as input at time t (e.g., log-moneyness, time to maturity, volatility, etc.).

Step 3: Compute terminal P&L. At the final time step, the P&L $W_T^{(i)}$ for each trajectory is computed as:

$$W_T^{(i)} = -\phi(S^{(i)}) + \sum_{t=0}^{T-\Delta t} y_t^{(i)}(S_{t+\Delta t}^{(i)} - S_t^{(i)}) - \sum_{t=0}^{T-\Delta t} \lambda \cdot S_t^{(i)} \cdot |y_t^{(i)} - y_{t-1}^{(i)}|$$

where $\phi(S^{(i)})$ is the payoff of the option for trajectory i, and λ is the proportional transaction cost rate. The hedge is initialized with $y_{-1} = 0$.

Step 4: Evaluate the loss (entropic risk). The loss function is the entropic risk measure, defined for each batch as:

$$\mathcal{L}(\theta) = \frac{1}{\gamma} \log \left(\frac{1}{N} \sum_{i=1}^{N} \exp(-\gamma W_T^{(i)}) \right)$$

where $\gamma > 0$ is the investor's risk aversion parameter. This loss encourages the network to find a strategy that yields high utility across the distribution of outcomes.

Step 5: Optimize the parameters. The neural network parameters θ are updated using a stochastic gradient descent method, such as Adam, to minimize $\mathcal{L}(\theta)$. This process is repeated over multiple batches until convergence.

Step 6: Price computation (utility indifference pricing). Once the policy is trained, the utility indifference price of the option is computed using an out-of-sample set of simulated price paths. For each path, two terminal P&L values are computed:

- $W_T^{\text{no option}}$: using the trained strategy without selling the option (i.e., $Z = 0$).
- $W_T^{\text{with option}}$: using the same strategy, but including the liability Z.

The utility indifference price is then estimated as the difference between the two entropic risk measures:

$$\text{Price} = \frac{1}{\gamma} \log \left(\frac{1}{N} \sum_{i=1}^{N} \exp(-\gamma W_T^{\text{no option},(i)}) \right) - \frac{1}{\gamma} \log \left(\frac{1}{N} \sum_{i=1}^{N} \exp(-\gamma W_T^{\text{with option},(i)}) \right)$$

This represents the premium that compensates the investor for the risk of selling the option under the optimal hedging policy.

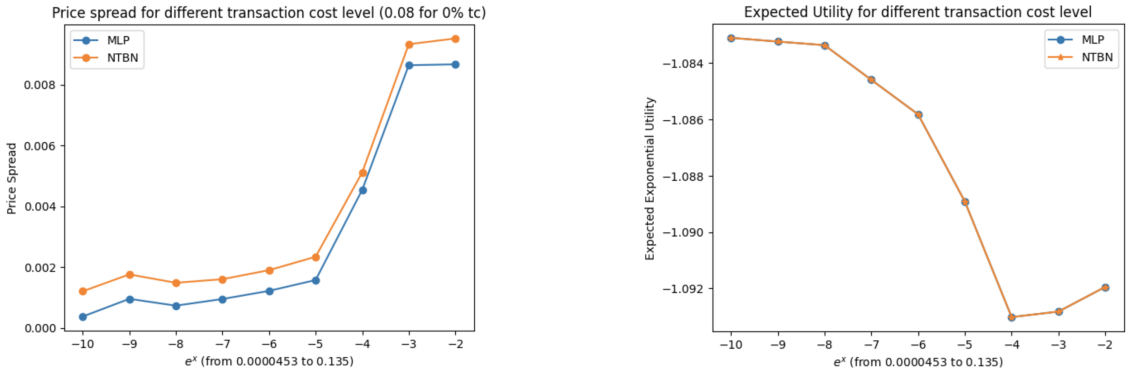
6.4 Numerical Results and Analysis

We now compare the performance of the Multi-Layer Perceptron (MLP) architecture and the No Transaction Band Net (NTBN) across a range of transaction cost levels. Figure 3 displays two key metrics: the utility indifference price spread (left) and the expected exponential utility (right), computed for different values of transaction cost rate c.

We observe in Figure 3.(a) that, as expected, the indifference price increases with transaction costs, reflecting the growing compensation required by the investor for bearing market frictions. While both networks exhibit this behavior, the NTBN consistently produces slightly higher price estimates than the MLP, which may be attributed to its more accurate identification of the no-transaction regions, leading to better protection against unnecessary trading costs.

Figure 3.(b) reports the expected utility under the optimal policy for each model. Here, both models deliver nearly identical performance across all levels of transaction costs, indicating that both networks are able to learn near-optimal policies in terms of final investor satisfaction. However, we note that the price differences in (a), despite being small, are consistent across different frictions and could be significant in practical applications involving large volumes or high-frequency hedging.

An interesting observation from Figure 3.(a) is that the option prices obtained through the stochastic control approach (dynamic programming) increase more rapidly with the transaction cost level than those estimated by the deep hedging networks. This behavior highlights a key difference in the nature of the two approaches. In the dynamic programming method, the policy is computed under worst-case or highly controlled assumptions — typically assuming the investor must follow the exact prescribed optimal strategy, with full awareness of model dynamics and transaction costs. This tends to result in more conservative pricing as frictions grow. By contrast, the deep hedging networks are trained to learn policies that perform well on average under simulated market conditions, possibly exploiting patterns in the data or absorbing some frictions dynamically. As a result, they may find more adaptive (but slightly riskier) strategies that yield lower prices, especially under higher transaction cost regimes. This gap illustrates the trade-off between theoretical robustness and data-driven efficiency: while classical methods ensure optimality under their assumptions, deep hedging may offer more cost-effective solutions in practical, data-rich environments.

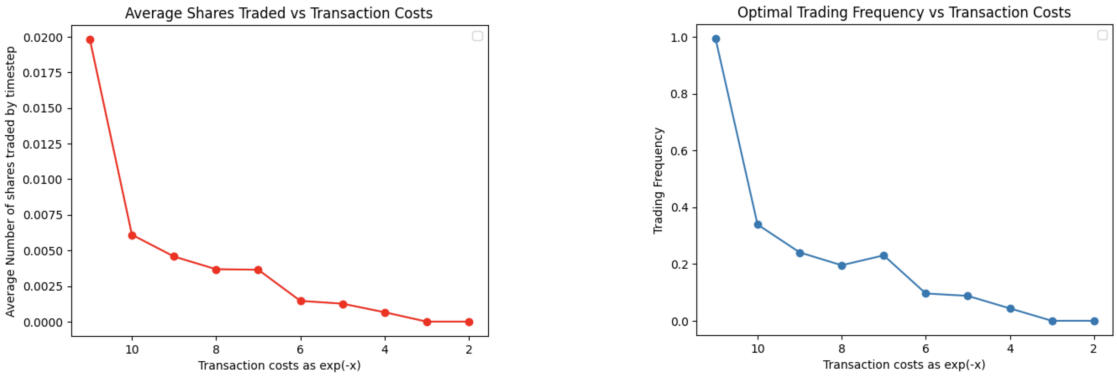


(a) Price Spread

(b) Expected Utilities

Figure 3: Prices and Utilities for different Transaction Costs levels, for both networks

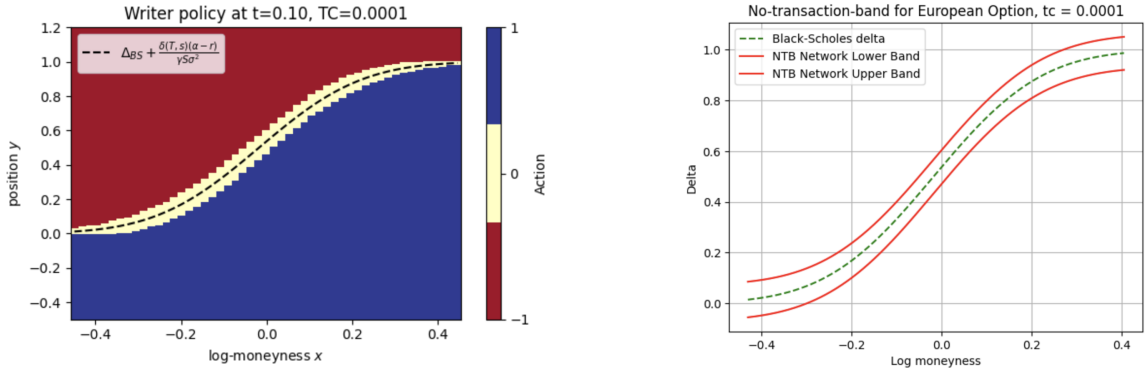
We also used the same trading measures as in the first part, to confirm the expected behavior, that trading frequency decreases as transaction costs increase, for the optimal policy learned (Figure 4).



(a) Average Shares Traded (b) Node Trading Frequency

Figure 4: Trading Measures

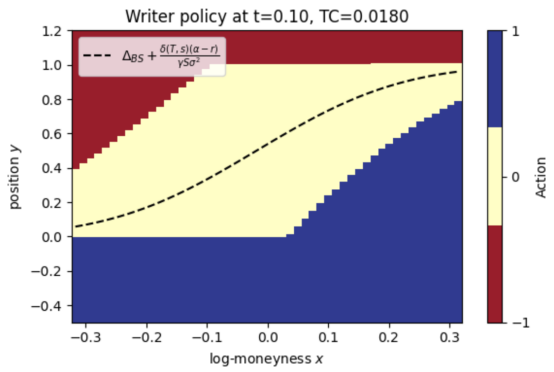
Figures 5 and 6 further explore the structure of the learned hedging strategies by comparing the no-transaction region derived from dynamic programming with that implicitly learned by the NTBN. For small transaction costs (Figure 4), we observe that the NTBN successfully replicates the shape and location of the theoretically optimal no-transaction region. The red area in (a) corresponds to "sell" actions, blue to "buy", and the central yellow band to "no action". This band matches closely the clamp region generated by the NTBN in Figure 4.(b), where the predicted lower and upper bounds encapsulate the Black-Scholes delta.



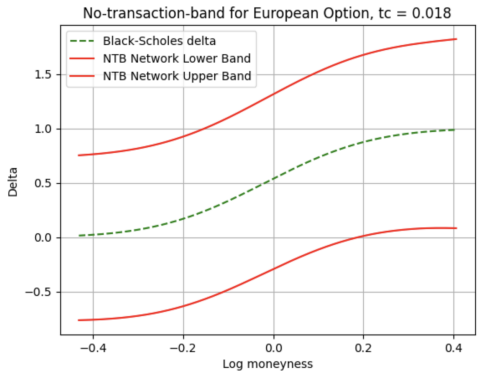
(a) No Transaction Region (b) No Transaction Bands

Figure 5: No Transaction Region found by Dynamic Programming and No Transaction Bands obtained by the network, for small transaction costs

The consistency persists for larger transaction costs, as shown in Figure 5. As costs increase, the no-transaction region widens, both in the dynamic programming solution and in the NTBN output. The network captures the shift in optimal policy by adjusting the bounds accordingly, again confirming that it internalizes the structure of the theoretical optimal solution, despite being trained only through simulations and without any explicit knowledge of the dynamic programming results.



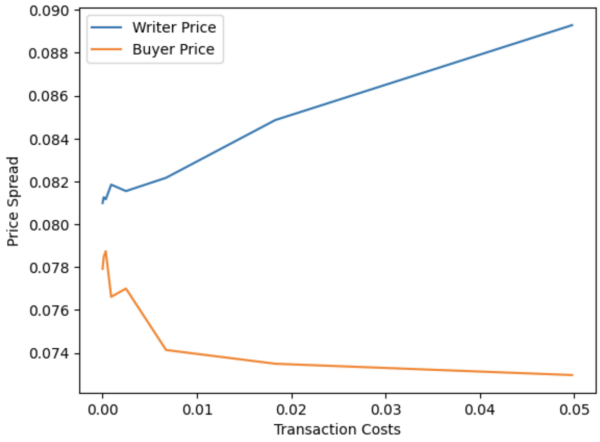
(a) No Transaction Region



(b) No Transaction Bands

Figure 6: No Transaction Region found by Dynamic Programming and No Transaction Bands obtained by the network, for high transaction costs

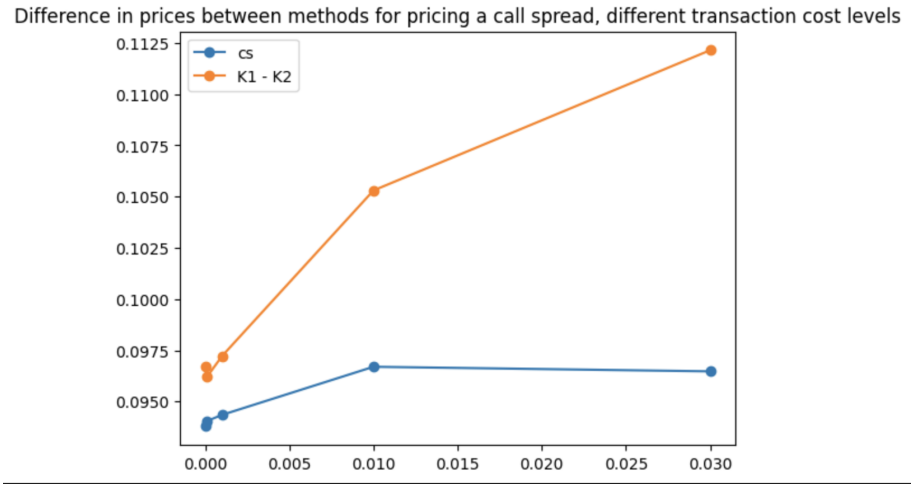
Finally, we also observe the 'Writer-Buyer' difference when transaction costs are introduced, since both hedging strategies are more costly when transaction costs increase, making the Writer ask more to sell the option, while the Buyer will be ready to pay less.

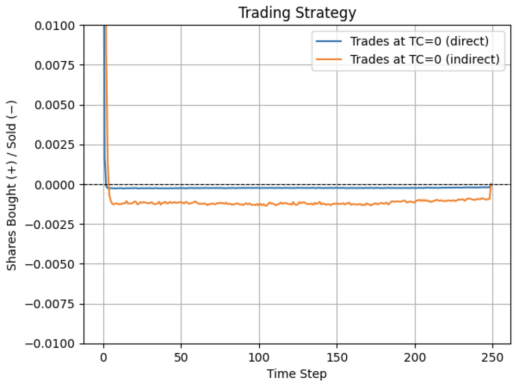


These results confirm both the effectiveness and the interpretability of deep hedging strategies. By embedding structural knowledge of the optimal policy into the network design, we achieve improved price estimates and strategies that align well with classical theoretical results — especially useful when transaction costs play a significant role.

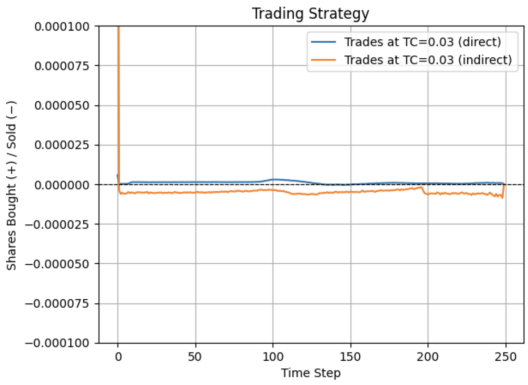
6.5 Bull Callspread Pricing

As we did previously, we used our Deep Hedging models to price and hedge a bull call spread, first indirectly (by pricing two calls with different strikes and taking their difference, and the combined strategies for the hedge) and then directly (by applying directly the payoff into our model for it to learn directly to hedge and therefore price the instruments).
We observe the same thing as in the previous part, the hedge and prices become non-linear with transaction costs : hedging strategies are less efficient and imply more trading, leading to higher prices for the indirect way, added to the appearance of a bid/ask spread with transaction costs.



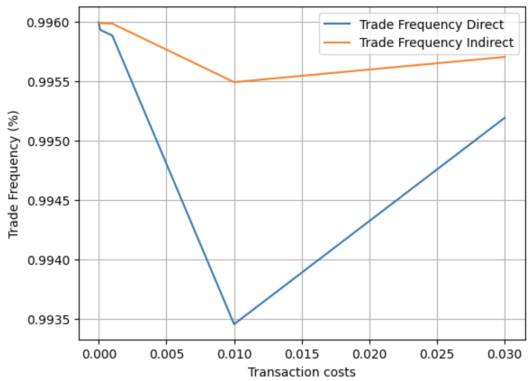


(a) 0% tc direct trading strategy

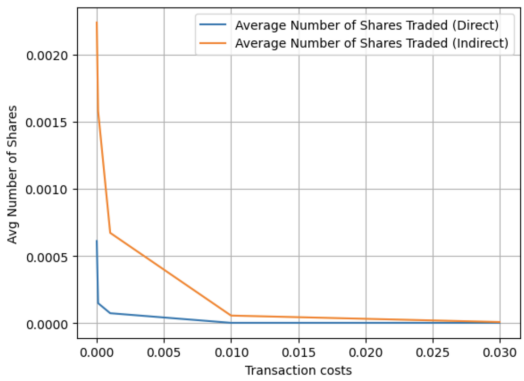


(b) 3% tc direct trading strategy

Figure 7: Trading strategies for direct and indirect approaches



(a) Trading frequency



(b) Avg number of shares traded

Figure 8: Trading metrics for direct and indirect approaches

7 Annex

7.1 Proof of the solution of the HJB (No transaction costs and no option)

Let us consider the CARA utility first.
We make a initial guess that the indirect utility function is of the following form :

$$U(t,W) = 1 - e^{-\gamma\phi(t)W+\psi(t)} \tag{100}$$

Then :

$$U_t = (\gamma\phi'(t)W - \psi'(t))e^{-\gamma\phi(t)W+\psi(t)} \tag{101}$$

$$U_w = \gamma\phi(t)e^{-\gamma\phi(t)W+\psi(t)} \tag{102}$$

$$U_{ww} = -\gamma^2\phi^2(t)e^{-\gamma\phi(t)W+\psi(t)} \tag{103}$$

and :

$$\pi^* = \frac{(\alpha - r)}{W\sigma^2\gamma\phi(t)} \tag{104}$$

Then, the HJB equation becomes :

$$(\gamma\phi'(t) + \gamma\phi(t)r)W - \psi'(t) + \frac{(\alpha - r)^2}{2\sigma^2} = 0 \tag{105}$$

which leads to :

- $\phi(t) = e^{r(T-t)}$
- $\psi(t) = \frac{(\alpha-r)^2}{2\sigma^2}(t - T)$

and :

$$\pi^* = \frac{(\alpha - r)}{W\sigma^2\gamma e^{r(T-t)}} \tag{106}$$

The reasoning behind the proof for the CRRA Utility is exactly the same.

References

[1] JOSEF TEICHMANN HANS BUEHLER, LUKAS GONON and BEN WOOD. Deep hedging.