Derivatives Credit Risk Project

Group E

Authors:
Federico La Penna (Università L. Bocconi)
federico.lapenna@master.unibocconi.it

Tommaso Zazzaron (Università L. Bocconi)
tommaso.zazzaron@master.unibocconi.it

Michele Sacerdoti (Università L. Bocconi)
michele.sacerdoti@master.unibocconi.it

Joan Matons Framis (Università L. Bocconi)
joan.matons@master.unibocconi.it

Leon Khishba (Università L. Bocconi)
leon.khishba@master.unibocconi.it

Valeriano Palmieri (Università L. Bocconi)
valeriano.palmieri@master.unibocconi.it

May 2025

Contents

Introduction

This project investigates the measurement and management of credit and funding risk in derivative portfolios by integrating stochastic interest rate modeling with equity price dynamics to produce scenario-based valuations.

The Hull–White model is implemented to simulate short-rate dynamics, providing a robust framework for generating realistic interest rate scenarios. These simulations are essential for accurately pricing interest rate-sensitive instruments. Calibration techniques are then applied to align the simulated interest rate paths with market data, with particular emphasis on reproducing the current zero-coupon bond price curve.

Subsequently, practical applications such as the pricing of Interest Rate Swaps (IRS) and the evaluation of their Mark-to-Market (MtM) across varying tenors are explored. These computations illustrate the temporal evolution of derivative values and underscore their implications for risk management and regulatory capital assessment.

The analysis then focuses on the quantification of Expected Exposure (EE) and Potential Future Exposure (PFE), which are critical inputs for calculating the Credit Valuation Adjustment (CVA) and the Funding Valuation Adjustment (FVA). CVA accounts for unilateral counterparty default risk, while FVA reflects the cost of funding derivative positions. These risk measures are computed for both the IRS and a short position in an at-the-money (ATM) Down-and-In Put options with varying barrier levels.

In sum, the project offers a comprehensive technical framework for evaluating and mitigating risks in derivative portfolios, effectively connecting theoretical modeling with practical financial applications.

Questions addressed: This project was undertaken to address nine specific questions, all of which have been thoroughly examined.

Exercise 1: Hull-White Model and Numeraire

The Hull–White model is a short-rate, mean-reverting, Ornstein–Uhlenbeck–type model. As the Vasicek model, it allows for negative interest rate, but in addition it is capable to fit the observed term structure of interest rates. The short-term interest rate SDE, under the risk neutral probability $\mathbb{Q}$, is written as:

$$\begin{cases} dr(t) = (\theta(t) - ar(t))dt + \sigma dW^{\mathbb{Q}}(t), \\ r(0) = r_0 \end{cases}$$

which is an Itô process with time-dependent drift $\theta(t)$, mean-reversion speed a, and volatility σ. The simulation of such a process has been implemented by the Euler-Maruyama discretization.
Let

$$dX_t = \mu(t, X_t)\,dt + \sigma(t, X_t)\,dW_t$$

be an SDE on $[0,T]$, and let $0 = t_0 < t_1 < \cdots < t_N = T$ be a partition of $[0,T]$. The *Euler–Maruyama approximation* is the discrete-time process $\{\hat{X}(t_n)\}_{n=0}^{N}$ defined as:

$$\hat{X}(t_{n+1}) = \hat{X}(t_n) + \mu\big(t_n, \hat{X}(t_n)\big)\,(t_{n+1} - t_n) + \sigma\big(t_n, \hat{X}(t_n)\big)\,\big(W_{t_{n+1}} - W_{t_n}\big)$$

Then, by the above mentioned discretization, the following relation for the short-term interest rate holds:

$$\hat{r}(t + \Delta t) = \hat{r}(t) + (\theta(t) - a\hat{r}(t))\Delta t + \sigma\sqrt{\Delta t}N(0,1)$$

From the interest rate's Euler scheme it is possible to define and simulate the Numeraire.
Generally speaking, for the circumstance of interest rate, the Numeraire is defined as:

$$\tilde{n} = e^{-\int_t^T r(s)ds}$$

As for the short-rate process, the integral can be discretized, and thought as a sum over infinitesimal time intervals. The simulation of the Numeraire is then:

$$\tilde{n} = e^{-\sum_{i=t}^{T} r(i)\Delta t_i}$$

The following boxes show the implementation of what has been presented.

Function r(t) under Hull-White

```
def HW(n, m, r0, theta, alpha, sigma, dt, T):
    r = np.zeros(shape=(n, m+1))
    r[:, 0] = r0
    for i in range(int(n)):
        for j in range(int(m)):
            r[i, j+1] = (r[i, j]+ (theta[j] - alpha * r[i, j]) * dt
                + sigma * np.sqrt(dt) * np.random.normal(0, 1))
    return r
```

Function numeraire

```
def numeraire(r, dt, include_T=True):
    delta = r[:, :-1] * dt
    total_integral = delta.sum(axis=1, keepdims=True)
    left_integral = np.cumsum(delta, axis=1)
    integral = total_integral - left_integral
    N = np.exp(-integral)
    if include_T:
        ones = np.ones((N.shape[0], 1))
        N = np.hstack([N, ones])
    return N
```

Exercise 2: Model Validation with the ZCB Curve of Prices

The purpose of the following section is to demonstrate the consistency of the Hull-White model, in particular it will show how the simulations will accurately reproduce the today zero-coupon bond curve of prices.
The zero-rates curve observed in the market is:

T (y)	1/12	0.5	1	3	5	10
Zero rate (%)	0.25	0.6	0.95	1.5	2.1	2.5

Table 1: Zero rates for various maturities

Using the formula for the price of a zero-coupon bond in the continuously compounded regime, namely:

$$P(0,T) = e^{-r_T \cdot T} \tag{1}$$

together with a cubic spline interpolation, the zero-coupon bond curve presents as shown in Figure 1.

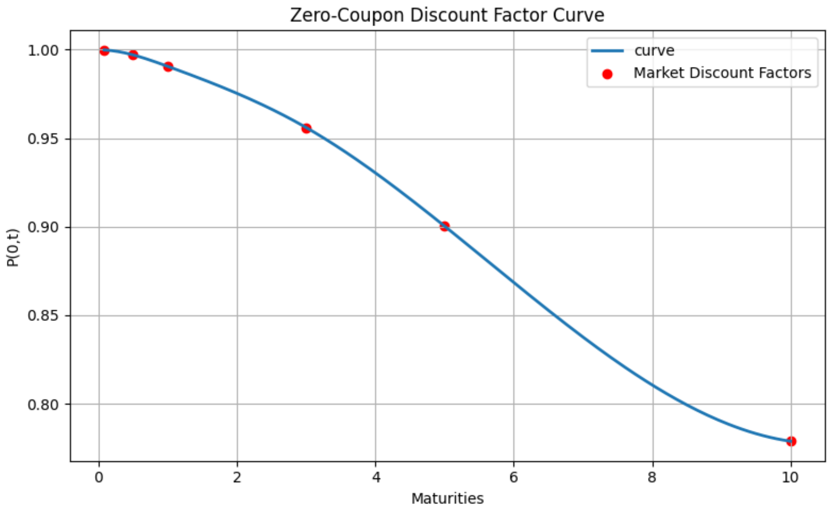


Figure 1: ZCB curve of market prices

Within a Hull-White framework, in which the short-term interest rate is modeled through a stochastic differential equation, the price of a zero coupon bond is given by the Feynman-Kac formula:

$$P(0,T) = \mathbb{E}^{\mathbb{Q}}[1 \cdot e^{-\int_0^T r_s ds}] \tag{2}$$

where the discount factor cannot be taken out of the expected value, being a function of a stochastic process. This results to be the expected value under $\mathbb{Q}$ of the Numeraire itself. By analytically computing the prices, and by interpolating them, one should retrieve the observed market curve.
Within the Hull–White model, the time-dependent drift term $\theta(t)$ is chosen so that the short-rate process mean-reverts to today's zero-curve. In other words, the simulation is calibrated such that, on average, the paths converge to the current zero-coupon yield curve. The function $\theta(t)$ is defined as follow:

$$\theta(t) \;=\; \frac{\partial f_M(0,t)}{\partial t} \;+\; a\, f_M(0,t) \;+\; \frac{\sigma^2}{2a}\bigl(1 - e^{-2at}\bigr) \tag{3}$$

with $\sigma > 0$ and $a > 0$. The inclusion of $\theta(t)$ enforces mean-reversion, so there is no need to recalibrate σ and a when matching today's curve.
The computation of the above mentioned expectation has been done directly in the implementation, confirming that the curve of simulated prices converges to the market prices curve. The graph below demonstrates an exact fit to the market curve regardless of the chosen values of σ and a, which have been randomly chosen.

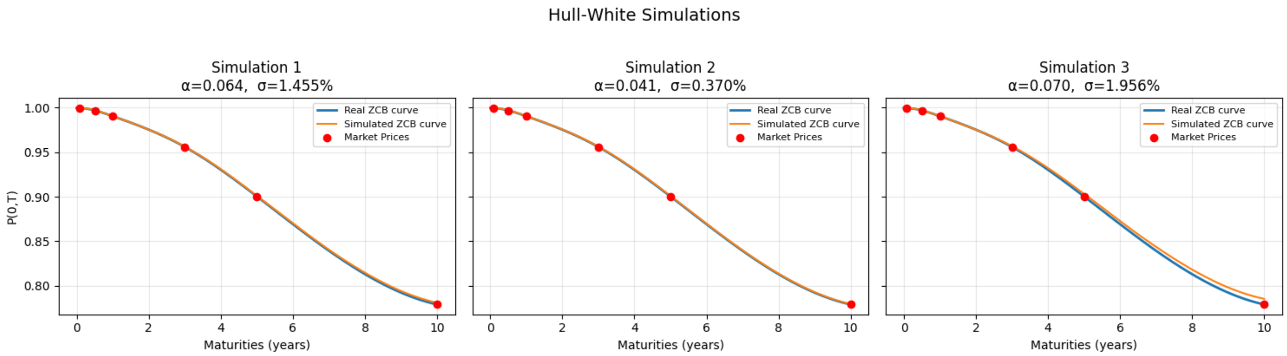


Figure 2: Fitting of today's ZCB price curve

Exercise 3: IRS Pricing

This section deals with the pricing of a interest-rate swap (IRS) using the standard market formula. At a generic time t, for a fixed-leg receiver, the swap's value is given by:

$$\text{IRS}(t) = \underbrace{\sum_{i:T_i>t} N \cdot K \cdot \tau_i \cdot P(t,T_i)}_{\text{fixed leg}} - \underbrace{\sum_{i:T_i>t} N \cdot \tau_i \cdot F(t,T_{i-1},T_i) \cdot P(t,T_i)}_{\text{floating leg}} \tag{4}$$

where $F(t,T_{i-1},T_i) = \frac{1}{\tau_i(T_{i-1},T_i)}\left(\frac{P(t,T_{i-1})}{P(t,T_i)} - 1\right)$ is the forward rate computed at time t for the time interval $[T_{i-1},T_i]$.

The fixed leg represents a stream of payments, based on a fixed rate K, discounted at time t by $P(t,T_i)$. The floating leg represents a stream of payments, based on a floating rate $F(t,T_{i-1},T_i)$ (forward rate), discounted at time t by $P(t,T_i)$.

Along with the zero-coupon curve detailed above, the standard IRS pricing formula has been applied to a 10-year swap with semiannual payments and a fixed rate K=2.488%, which should ensure a zero value at inception. As one can expect, the calculation reproduces an almost zero price:

$$IRS(K = 2.884\%) \approx -0.0003682877 \approx 0 \tag{5}$$

The small mismatch is due to interpolation errors, and consequently, to a non-perfect computation of the forward rates. Below is shown the implementation algorithm for the IRS pricing.

Function numeraire

```
def IRS(P, T, frequency, K, kind='cubic'):
    payment_dates = np.arange(t_start + frequency,T+1e-12,frequency)
    t_prev = np.insert(payment_dates[:-1], 0, t_start)
    alpha = payment_dates - t_prev
    P_prev    = P(t_prev)
    P_payment = P(payment_dates)
    L = (P_prev / P_payment - 1) / alpha
    PV_float = np.sum(alpha * L * P_payment)
    PV_fixed = K * np.sum(alpha * P_payment)
    PV_swap = PV_fixed - PV_float
    return PV_swap
```

For the IRS valuation, the zero-coupon bond prices $P(t, T_i)$ are not obtained from the Monte Carlo numéraire simulation but instead from the closed-form Hull–White formulas. Recall that, under the risk-neutral measure, the arbitrage-free price of a bond maturing at T satisfies:

$$P(t,T) \;=\; \mathbb{E}^{Q}\Big[\exp\big(-\int_t^T r(s)\,\mathrm{d}s\big)\,\big|\,\mathcal{F}_t\Big]$$

Having a solution for the market–implied dynamics of the domestic short-rate, the following closed-form can be derived:

$$P(t,T) \;=\; A(t,T)\, e^{-B(t,T)\, r_d(t)}$$

where

$$B(t,T) \;=\; \frac{1}{a}\big(1 - e^{-a\,(T-t)}\big),$$

$$A(t,T) \;=\; \frac{P^M(0,T)}{P^M(0,t)}\exp\Big\{\,B(t,T)\,f^M(0,t) \;-\; \frac{\sigma^2}{4a}\big(1 - e^{-2a\,t}\big)\,B(t,T)^2\Big\}.$$

This choice has been made to implement accuracy in the MtM simulation.

Exercise 4: MtM computation across tenors

In this section, we simulate the Mark-to-Market (MtM) price series over tenors of the zero curve using 10.000 independent paths. Throughout, we set the volatility to $\sigma = 0.015$ and the mean-reversion speed to $a = 0.01$. By construction, every simulated price is zero both at inception and at the terminal date T=10 years. Between these two endpoints, the paths diverge from zero in a manner illustrated by the plots below. The IRS pricing across time is conducted through the exploitation of the forward rates curve. Indeed as seen earlier today ZCB curve allows to price at inception the IRS which equals zero but across time the price must be recomputed. Retrieving the market expectations of the future rates, comes from the aforementioed forward rates formula: $f(k, k+1) = (\frac{P(k+1)}{P(k)} - 1) \cdot \frac{1}{\tau(k,k+1)}$. These, then, are put as input in the IRS pricing function.

MtM matrix construction

```
mtm_matrix = np.zeros((nT, n_paths))

for k in range(1, nT-1):
    t0 = payment_dates[k]
    j0 = idx_sched[k]
    for i in range(n_paths):
        # extract the short-rate at time t0 for path i
        r_u = r[i, j0]
        # define absolute bond pricer via closed-form HW
        P_abs = lambda s, t0=t0, ru=r_u: bond_price_hw(t0, s, ru, alpha, sigma,
            zero_rate, forward_rate)
        # compute MtM for this path and time
        mtm_matrix[k, i] = IRS(P_abs, T, delta, K_fair, notional=notional, t_start=t0)
```

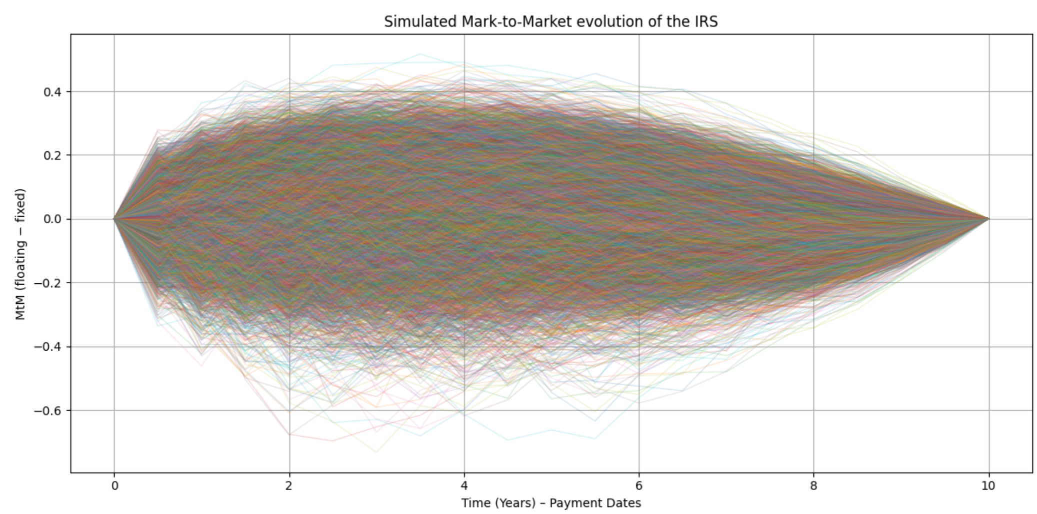


Figure 3: MtM simulation

The resulting plot exhibits the anticipated "onion" pattern: the swap's mark-to-market value is zero at both inception and maturity, while in the intervening periods the paths fan outward before reconverging at zero. The simulation was carried out using six-month time steps, exactly matching the swap's payment schedule. This choice reflects the practical reality that an IRS's mark-to-market value only changes when coupon payments are made.

Exercise 5: EE and PFE computation

This section presents the calculation of two key counterparty-credit metrics: the Expected Exposure (EE) and the 99th-percentile Potential Future Exposure (PFE). The EE at each future time point is defined by:

$$EE = \mathbb{E}^{\mathbb{Q}}[(MtM, 0)^{+}] \tag{6}$$

while the PFE is determined as the 99 % value-at-risk of the positive MtM distribution across all simulation paths. The following plot illustrates the evolution of both EE and PFE over the life of the contract.

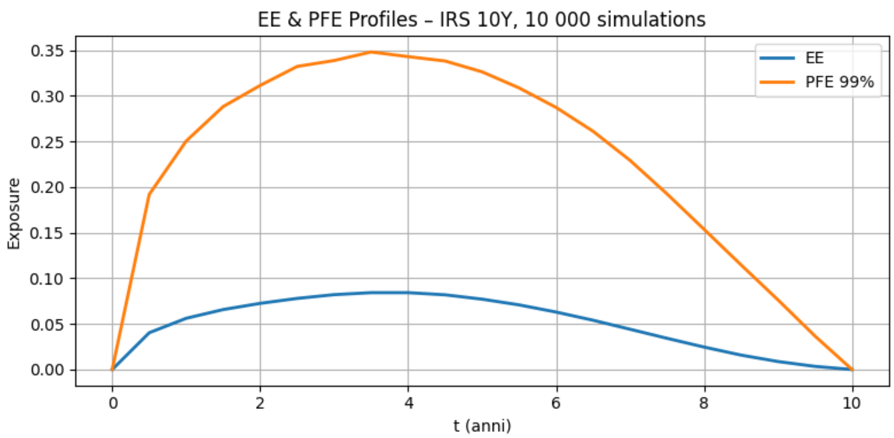


Figure 4: EE and PFE simulation

Exercise 6: CVA computation

This section presents the calculation of the Credit Valuation Adjustment (CVA) for the previously described IRS, using the following counterparty credit-spread curve:

T (y)	1	2	3	4	5	6	7	8	9	10
CDS spread (bps)	53	98	122	132	138	142	144	145	146	147

Table 2: CDS spreads (bps) by maturity

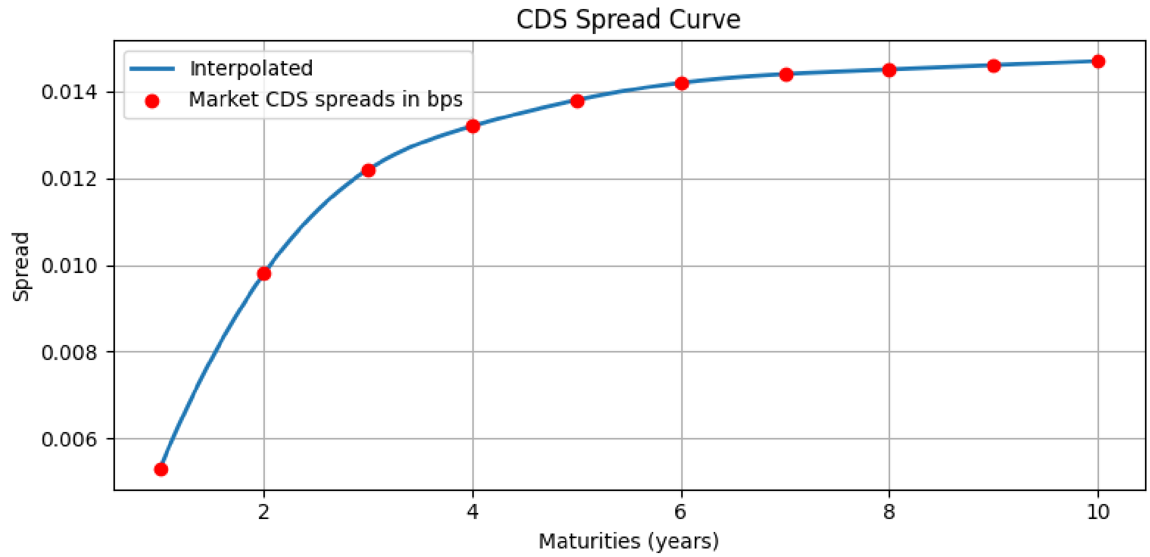


Figure 5: Credit Spread Curve of the counterpart

Unilateral CVA quantifies the expected loss due to counterparty default on a derivative, reflecting the additional cost premium embedded in the transaction. It is driven entirely by the counterparty's credit-spread curve, which encodes market-implied default risk. The CVA is computed as:

$$\text{CVA} \;:=\; \mathbb{E}_t\Big[\,(1-R_C)\mathbf{1}_{\,t<\tau_C\le\tau}\,D\big(t,\tau_C\big)\,\text{MtM}\big(\tau_C\big)^+\Big] \tag{7}$$

The indicator $\mathbf{1}_{\,t<\tau_C\le\tau}$inside the expectation restricts the loss to scenarios where the counterparty defaults during the interval $(t_{k-1}, t_k]$. Modeling the default time τ_c as exponentially distributed with constant hazard rate λ, the probability of default between t_{k-1} and t_k is: $e^{-\lambda(t-1)\cdot(t-1)} - e^{-\lambda(t)\cdot(t)}$.
Recalling that $R_c = 1 - LGD$ and by discetization of the CVA formula, the following holds:

$$\text{CVA} \;=\; \text{LGD}\,\sum_{k=1}^{N} \text{EE}(t_k)\;\text{DF}(t_k)\;\text{dPD}_k \tag{8}$$

where:

$$dPD_k = e^{-\frac{s}{LGD}\cdot t_{k-1}} - e^{-\frac{s}{LGD}\cdot t_k} \tag{9}$$

Here, $\frac{s}{LGD}$ is the hazard rate, namely $\lambda(\cdot)$.
The implementation has been done by the following algorithm:

CVA computation

```
schedule = np.arange(0, T+1e-12, delta)
LGD      = 0.40
# survival probabilities and default prob.
S    = np.exp(-cds_curve(schedule)/LGD * schedule)
dPD  = S[:-1] - S[1:]
# discount factors and expected exposure
DF   = P_mkt(schedule)
EE   = exposure.mean(axis=1)
# CVA at time 0
CVA  = LGD * np.sum(EE[1:] * DF[1:] * dPD)
print("CVA at time 0 is:", CVA)
```

At inception, the unilateral CVA over [0,T] is computed as

$$\mathrm{CVA}(0,T) \approx 0.006290983212815195$$

Because CVA is entirely driven by the counterparty's credit-spread curve, wider spreads translate into higher default probabilities and thus greater CVA, we conduct a sensitivity test. By applying a parallel shift of +1 bps to the entire spread curve and recomputing, the CVA rises to

$$\mathrm{CVA}(0,T) \approx 0.00962441256855835$$

demonstrating the expected increase in counterparty-risk adjustment when perceived default risk grows.

Exercise 7: FVA computation

In this section, the funding valuation adjustment (FVA) for the interest-rate swap described above has been computed. In the risk-neutral measure $\mathbb{Q}$, the standard FVA formula is

$$\mathrm{FVA} = \mathbb{E}^{\mathbb{Q}}[P(0,t)(R_{\text{fund}} - r_f)max(\mathrm{MtM(t)},0|\mathcal{F}_0]$$

where by assumption the funding spread is $R_{\text{fund}} - r_f = 0.004$. Plugging this into the formula and evaluating the expectation yields

$$\mathrm{FVA} \approx 0.0019162888985067245 \tag{10}$$

Exercise 8: Collateralized IRS contract

In this section, the same implementation as for the previous IRS transaction has been applied, with one important distinction: the position is now fully collateralized. Counterparty exposure is covered by collateral at each repricing date and remains uncollateralized only over the 10-day margin period of risk (MPOR). Expected exposure (EE), the 99th-percentile potential future exposure (PFE), credit valuation adjustment (CVA) and funding valuation adjustment (FVA) are computed as before, but several formulas are modified under collateralization. In particular, the CVA under this regime is given by:

$$\mathrm{CVA} := \mathbb{E}_t\Big[(1-R_C)\,\mathbf{1}_{t<\tau_C\le\tau}\,D\big(t,\tau_C\big)\,\big(\mathrm{MtM}\big(\tau_C+\mathrm{MPOR}\big)^+ \;-\; \mathrm{MtM}\big(\tau_C\big)^+\big)\Big].$$

By simulating the MtM, the computation of the EE and PFE have the following graphical representation.

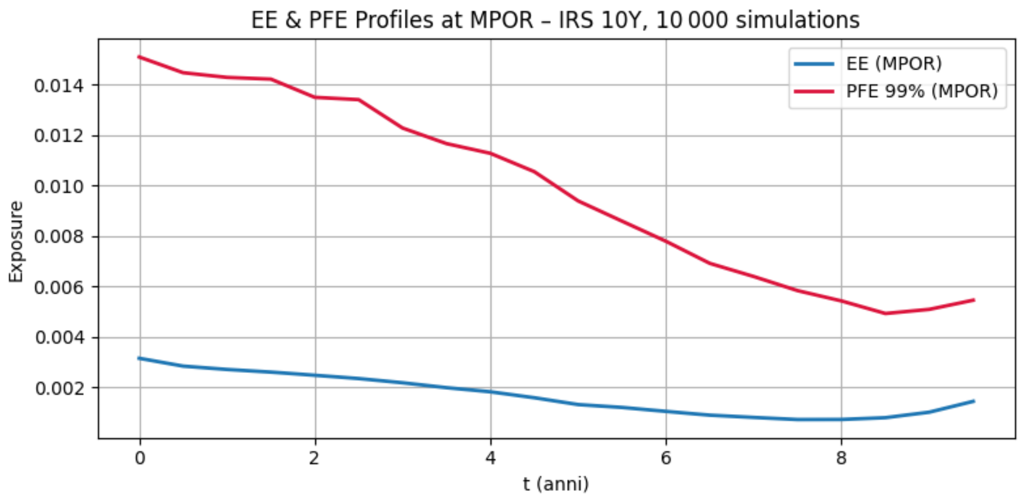


Figure 6: EE and PFE MPOR

Using the previously defined formula and the same FVA computation methodology as in the preceding section, the CVA and FVA for a collateralized contract with a 10-day margin period of risk (MPOR) are given by:

$$\mathrm{CVA} \approx 0.00017040886505553566,$$

$$\mathrm{FVA} \approx 0.000056.$$

Exercise 9: FVA computation on short Down-and-In (DI) put option

This section presents an implementation of a Monte Carlo simulation for a geometric Brownian motion model in which the drift is given by the previously generated stochastic short-rate paths. Ten thousand scenarios are produced for a share with initial price $S(0) = 100$, volatility $\sigma = 0.3$, and maturity $T = 1.5$ years. For each scenario, a down-and-in European put option (short position of size –1) with strike $K = 100$ and barrier levels $B \in (50, 55, 60, 65, 70, 75, 80)$ is priced with Least Square Monte Carlo. This is due to the complexity of the MtM simulation that a nested Monte Carlo would have.
The Funding Valuation Adjustment (FVA) is then calculated for each barrier by applying the constant funding spread $R_{\text{fund}} - r_f = 0.004$ and taking the risk-neutral expectation of the discounted positive

mark-to-market exposures.

The underlying of the Down-and-in put, by hypothesis, follows a geometric brownian motion (GBM):

$$\begin{cases} dS(t) = \mu\, S_t\, \mathrm{d}t \;+\; \sigma\, S(t)\, \mathrm{d}W^{\mathbb{P}}(t)\,, \quad t \geq 0, \\ S_0 = 100\,. \end{cases}$$

whose exact solution is:

$$S(t) = S(0) \cdot e^{(r-\frac{\sigma^2}{2})t+\sigma W^{\mathbb{Q}}(t)}$$

However, the model assumes a constant risk-free rate, r. It is possible to extend the reasoning to the time-dependent process r(t):

$$S(t) = S(0) \cdot e^{\int_0^t r(s)ds-\frac{\sigma^2}{2}t+\sigma W^{\mathbb{Q}}(t)}$$

As r(t) has been also simulated, the implemetation of the $S(t)$ trajectories are done by inputting the discrete time r(t) values, $\forall t \in (0, T)$.

Here below is shown the algorithmic implementation.

Simulate GBM Paths

```
S = np.zeros((Nsim, m_opt+1))
S[:, 0] = 100
vol = 0.30
for i in range(Nsim):
    for j in range(m_opt):
        r = r_paths[i, j]
        z_opt  = np.random.normal()
        S[i, j+1] = S[i, j] * np.exp((r - 0.5 * vol**2) * dt_opt
        + vol * np.sqrt(dt_opt) * z_opt)
```

Subsequently, the down-and-in (DI) scenario is considered. A DI put option becomes active only if the underlying asset's path falls to or below a specified barrier level (i.e. "down"), at which point the put contract is "in". Let's see it graphically:

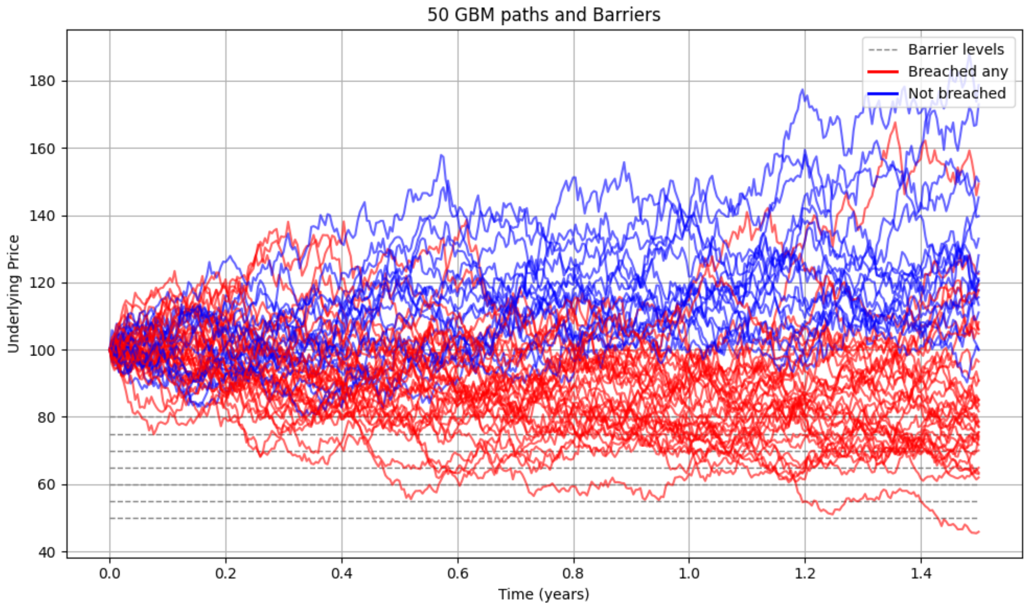


Figure 7: GBM paths simulation with barrier condition

The activation feature of a Down-and-In Put is incorporated directly into the mark-to-market simulation of the contract and ultimately used in the FVA calculation. Computing the true MtM at each time step would, in principle, require a nested Monte Carlo: for each outer scenario one would have to run an inner Monte Carlo to estimate the continuation value, leading to $O(N^2)$ total simulations (e.g. $10\,000^2$ in step 2 alone). Such exponential growth in cost makes the direct approach impractical.

Instead, a Least Squares Monte Carlo (LSMC) method of Longstaff and Schwartz (2001) has been applied, which replaces the inner Monte Carlo by a regression onto a set of basis functions. In general, at each time t_i we have

$$\begin{cases} V\big(t_i, S_{t_i}, H_{t_i}\big) = e^{-r\,\Delta t}\,\mathbb{E}^{\mathbb{Q}}\Big[V\big(t_{i+1}, S_{t_{i+1}}, H_{t_{i+1}}\big) \mid S_{t_i}, H_{t_i}\Big] \\ V\big(T, S_T, H_T\big) = H_T \max\{K - S_T, 0\} \end{cases}$$

where:

- S_{t_i} is the underlying asset price,
- $H_{t_i} = \mathbf{1}\big\{\min_{0\le u\le t_i} S_u \le B\big\}$ is the barrier-activation indicator,
- r is the simulated time-dependent risk–free rate, and $\Delta t = t_{i+1} - t_i$.

LSMC approximates the conditional expectation

$$\mathbb{E}^{\mathbb{Q}}\big[e^{-r\,\Delta t}\,V(t_{i+1}, S_{t_{i+1}}, H_{t_{i+1}}) \mid S_{t_i}, H_{t_i}\big]$$

by projecting the discounted next-step values onto a finite basis $\{\phi_k(S,H)\}_{k=1}^K$. Defining the regression matrix $X \in \mathbb{R}^{N\times K}$ with entries $X_{j,k} = \phi_k\big(S_{j,i}, H_{j,i}\big)$ and the response vector $Y_j = e^{-r\,\Delta t} V_{j,i+1}$, the LSMC coefficients are

$$\beta^{(i)} = \arg\min_{\beta\in\mathbb{R}^K} \sum_{j=1}^{N} \Big(Y_j - \sum_{k=1}^{K} \beta_k\, \phi_k(S_{j,i}, H_{j,i})\Big)^2 \quad \Longrightarrow \quad \beta^{(i)} = (X^\top X)^{-1} X^\top Y.$$

Then one sets

$$V_{j,i} \;=\; \sum_{k=1}^{K} \beta_k^{(i)}\, \phi_k\big(S_{j,i}, H_{j,i}\big) \tag{11}$$

for each scenario j, and proceeds backward in time.

Choice of basis for a Down-and-In Put: A parsimonious and effective set of basis functions is

$$\begin{aligned}
\phi_1(S,H) &= 1\\
\phi_2(S,H) &= S\\
\phi_3(S,H) &= S^2\\
\phi_4(S,H) &= H\\
\phi_5(S,H) &= H\,S\\
\phi_6(S,H) &= H\,S^2
\end{aligned}$$

Here the indicator H cleanly separates the pre-activation regime ($H = 0$) from the post-activation regime ($H = 1$), allowing the regression to capture the payoff discontinuity at the barrier without resorting to nested simulations. This LSMC-based MtM then feeds directly into the FVA integral via the positive exposure $\max(V, 0)$ under the risk–neutral measure.

Algorithmically speaking, this has been implemented by the following lines:

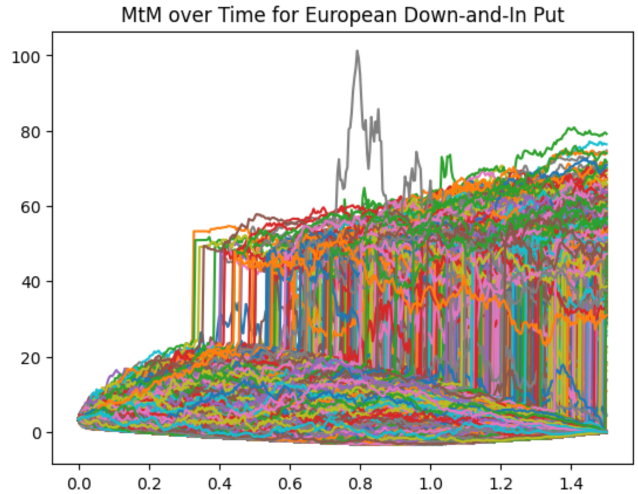
LSMC for European Down-and-In Put

```
for B in barriers:
    H = np.logical_or.accumulate(S_paths <= B, axis=1)
    V = np.zeros_like(S_paths)
    V[:,-1] = H[:,-1] * np.maximum(K - S_paths[:,-1], 0)
    for i in range(m_steps-1, -1, -1):
        Y   = V[:,i+1] * np.exp(-r_paths[:,i] * dt)
        S_i = S_paths[:,i]
        H_i = H[:,i].astype(float)
        X   = np.column_stack([np.ones(n_scenarios),
            S_i, S_i**2, H_i, H_i * S_i, H_i * (S_i**2)])
        beta, *_ = np.linalg.lstsq(X, Y, rcond=None)
        V[:,i]    = X.dot(beta)
    MtM0 = V[:,0].mean()
    MtMmean = V.mean(axis=0)
    results.append((B, MtM0))
    exposure = np.maximum(V, 0)
    EE = exposure.mean(axis=0)
    fund_spread = 0.004
    DF_us = P_mkt(ts)
    FVA = np.sum(EE[1:] * DF_us[1:] * fund_spread)
    FVA_DI.append(FVA)
```

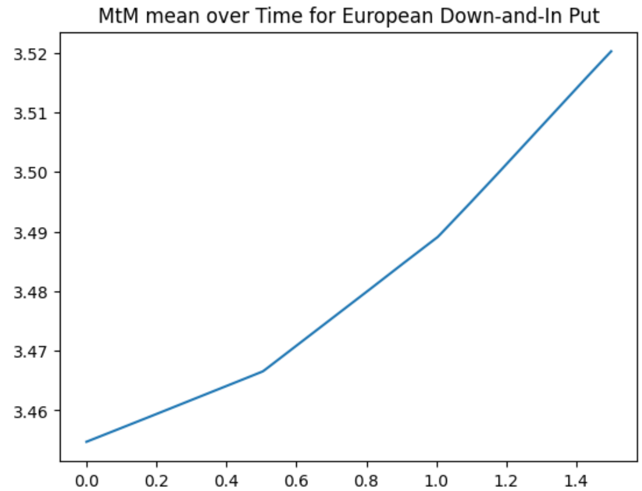
From this implementation different FVA values for the different barriers rise:

Barrier	FVA
50	0.023081
55	0.034302
60	0.045866
65	0.057191
70	0.067233
75	0.073756
80	0.078147

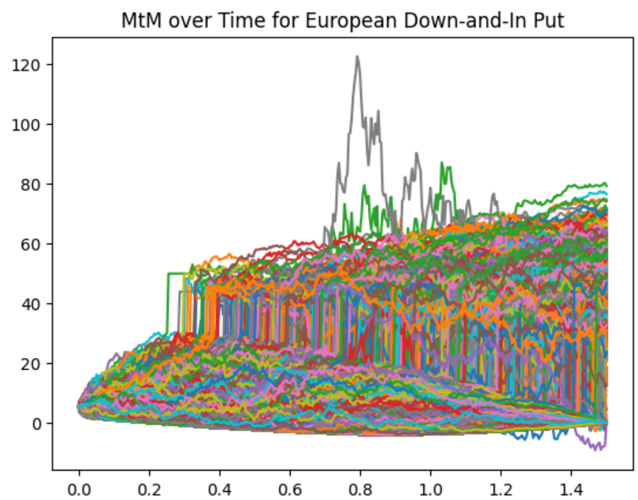
Below the MtM simulation graphs are reported, illustrating the scenario-by-scenario MtM paths for the European Down-and-In Put. Their atypical "kinks" and discontinuities arise from the inclusion of the barrier-activation indicator in the LSMC basis. You may also notice some slightly negative values; these are numerical artifacts of the least squares regression that occasionally overfit in extreme scenarios. It is important to remember that LSMC is an approximation of the true nested-Monte-Carlo MtM. In expectation it converges to the correct MtM with high accuracy, even though individual paths may exhibit local errors. As the overlaid average MtM curve confirms, the method delivers a very precise estimate of the true fair value across the full time grid.



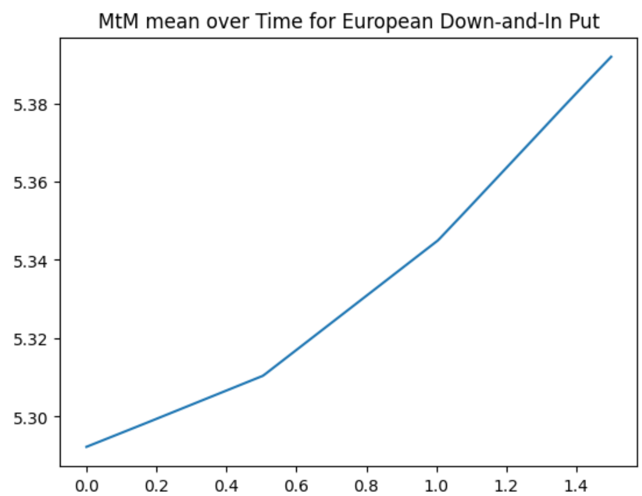
(a) MtM Barrier 50



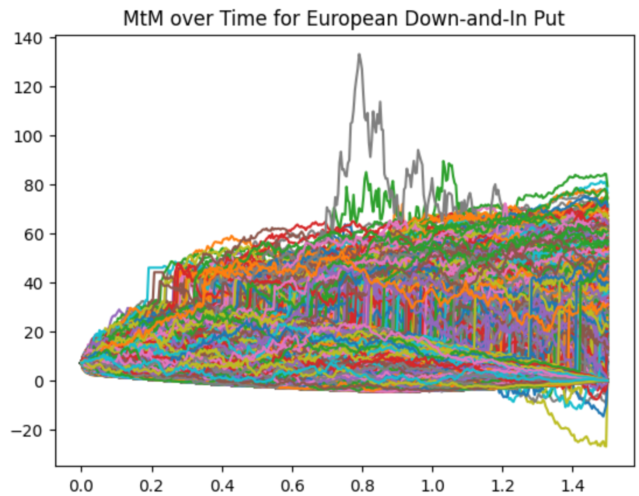
(b) MtM mean



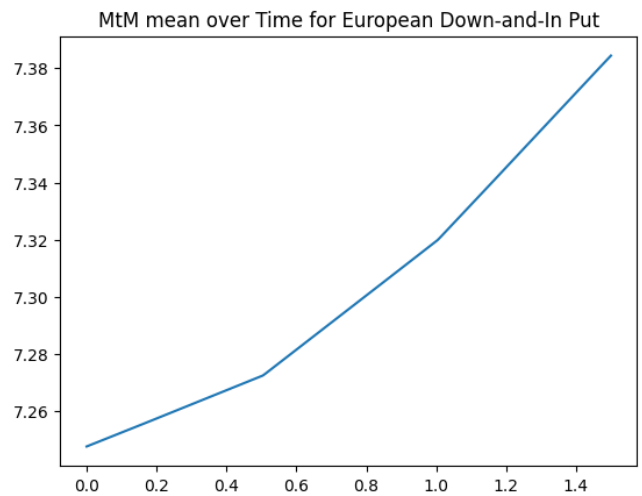
(a) MtM Barrier 55



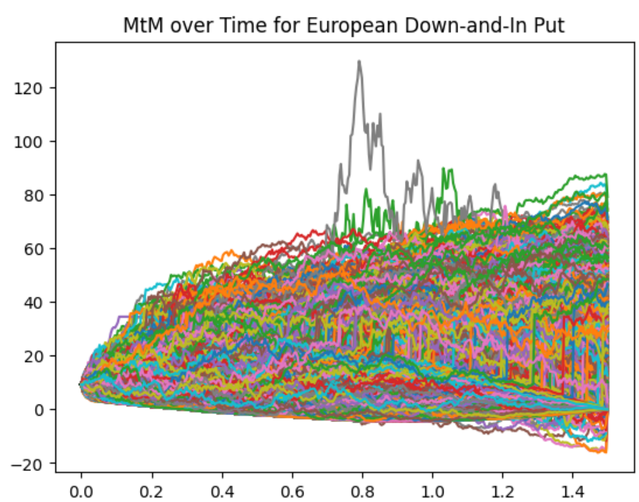
(b) MtM mean



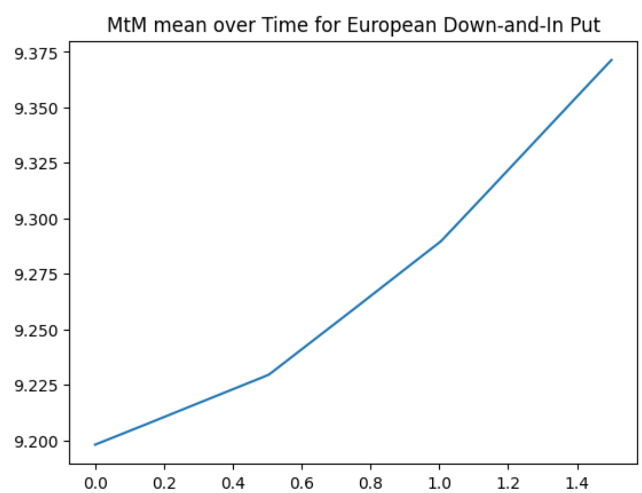
(a) MtM Barrier 60



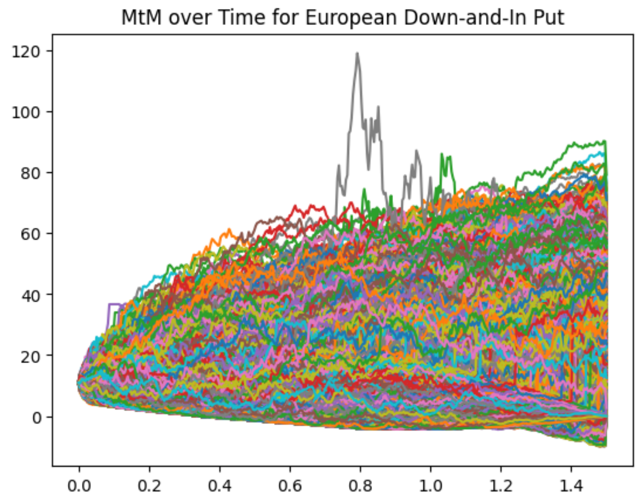
(b) MtM mean



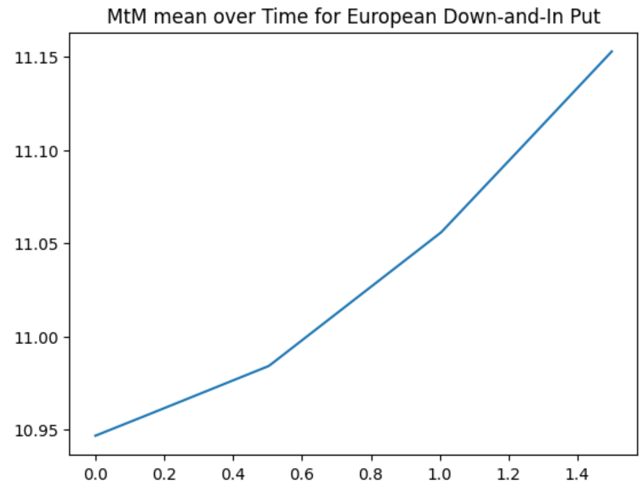
(a) MtM Barrier 65



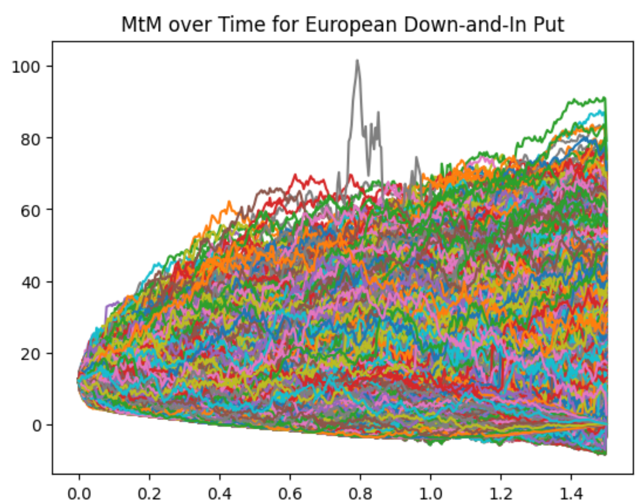
(b) MtM mean



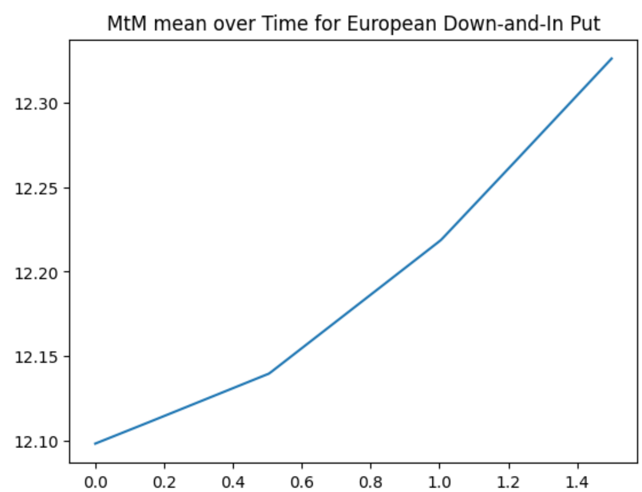
(a) MtM Barrier 70



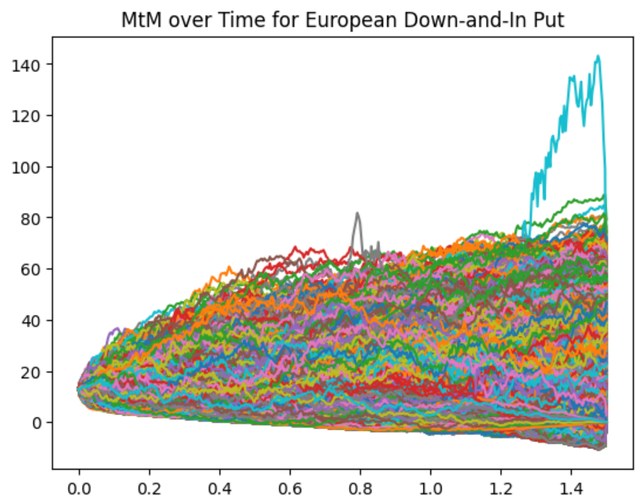
(b) MtM mean



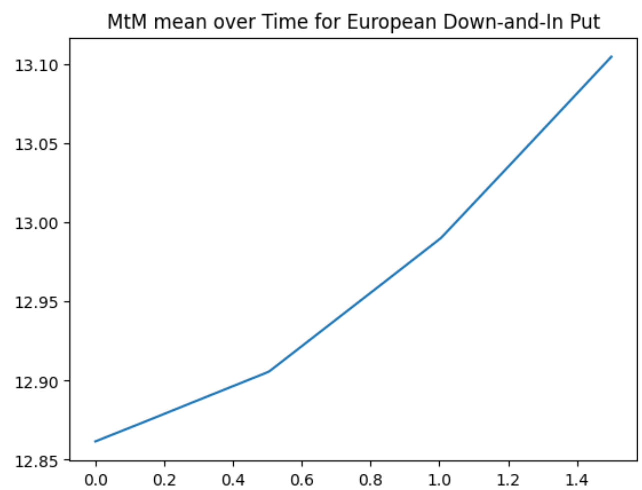
(a) MtM Barrier 75



(b) MtM mean



(a) MtM Barrier 80



(b) MtM mean