Thorough mathematical modelling and analysis of Uniswap v3

Mnacho Echenim, Emmanuel Gobet, Anne-Claire Maurice

Thorough mathematical modeling and analysis of Uniswap v3*

Mnacho ECHENIM† Emmanuel GOBET‡ Anne-Claire MAURICE§

September 30, 2023

Abstract

Automated Market Makers have emerged quite recently, and Uniswap is one of the most widely used platforms (it covers 96% of the available pools as of today). This protocol is challenging from a quantitative point of view, as it allows participants to choose where they wish to concentrate liquidity. There has been an increasing number of research papers on Uniswap v3 but often, these articles use heuristics or approximations that can be far from reality: for instance, the liquidity in the pool is sometimes assumed to be constant over time, which contradicts the mechanism of the protocol. The objectives of this work are fourfold: first, to revisit Uniswap v3's principles in detail (starting from the open source code) to build an unambiguous knowledge base. Second, to analyze the Impermanent Loss of a liquidity provider by detailing its evolution, with no assumption on the swap trades or liquidity events that occur over the time period. Third, we introduce the notion of a liquidity curve. For each curve, we can construct a payoff at a given maturity, net of fees. Conversely, we show how any concave payoff can be synthetized by an initial liquidity curve and some tokens outside the pool; this paves the way for using Uniswap v3 to create options. Fourth, we analyze the behavior of collected fees without any simplifying hypothesis (like a constant liquidity or zero Spot-Pool spread) under the mild assumption that the pool price follows a general Ito price dynamic. The value of the collected fees then coincides with an integral of call and put prices. Our derivations are supported by graphical illustrations and experiments.

KEYWORDS: Automated Market Makers; modeling mechanisms; profit and loss analysis

1 Introduction

1.1 DeFi and Automated market makers

As DeFi increased in popularity – see [Gobet and Melachrinos, 2023] and [Capponi et al., 2023] for an overview –, it quickly became necessary to find tools that could play the same role as Limit Order

*This work was presented at the Blockchain@X-OMI Workshop on Blockchain and Decentralized Finance, Paris, September 20-21 2023. The authors would like to thank participants for their feedback.

†Laboratoire d'Informatique de Grenoble (LIG), CNRS, Grenoble INP, UGA. 700 avenue Centrale, Domaine Universitaire, 38401 Saint Martin d'Hères, France. Email: MNACHO.ECHENIM@UNIV-GRENOBLE-ALPES.FR

‡Centre de Mathématiques Appliquées (CMAP), CNRS, Ecole Polytechnique, Institut Polytechnique de Paris. Route de Saclay, 91128 Palaiseau Cedex, France. Email: EMMANUEL.GOBET@POLYTECHNIQUE.EDU

§Kaiko - Quantitative Data. 2 rue de Choiseul 75002 Paris, France. Email: ANNE-CLAIRE.MAURICE@KAIKO.COM

Books in traditional finance, so that actors could easily exchange crypto assets. This has led to the design of *Automated Market Makers*, or AMMs, which are protocols that permit the automated execution of buy and sell orders in a blockchain. The principle of an AMM is simple: any user can deposit their tokens in a so-called liquidity pool, where they can then be used by other actors for their trading activities. These users are called *Liquidity Providers*, or LPs, and they are rewarded for making their tokens available by the trading operations on the latter that require the payment of a fee. More specifically, consider a liquidity pool consisting of two crypto assets, X and Y. Although this setting can be generalized, it is quite standard and permits to explain the way AMMs work in a simple manner. An LP deposits respective amounts x and y of these tokens, which, for example, can be used by an actor swapping some tokens X for tokens Y. At any time, the LP can redeem their position and recover amounts x' and y' of both tokens, corresponding to the values initially deposited plus fees.

A key feature of any AMM is the way the value of one token in terms of the other is automatically derived. This value controls the amount of tokens provided and received in a swap operation, as well as the number of tokens an LP will recover when they redeem their position. A large number of AMMs are called *Constant Function Market Makers* (CFMMs) and rely on the constant function paradigm to determine this value [Angeris and Chitra, 2020]. Formally, a CFMM is described by the respective reserves R_X and R_Y of tokens X and Y that are available in the liquidity pool, and an invariant function $\mathscr{I}$ that determines the swap operations on the pool that are permitted. Given incremental positions $(\Delta x, \Delta y) \in \mathbb{R}^+ \times \mathbb{R}^+$ in tokens X and Y respectively and a direction $d \in \{-1, 1\}$, a swap consists in trading Δx tokens X for Δy tokens Y when $d = -1$ (resp. Δy tokens Y for Δx tokens X when $d = 1$). Such a swap is permitted exactly when

$$\mathscr{I}(R_X - d \cdot \Delta x, R_Y + d \cdot \Delta y) \geq \mathscr{I}(R_X, R_Y).$$

Note that rational traders will target the best number of tokens, i.e. an equality in the above. CFMMs can be classified depending on the form of their invariant function. A common invariant function involves the product of the reserves: $\mathscr{I}(R_X, R_Y) = R_X \cdot R_Y$. AMMs with such an invariant function are called *Constant Product Market Makers (CPMMs)*, and Uniswap is one of those.

1.2 The Uniswap protocols

Uniswap has released two popular AMM protocols, Uniswap v2 [Adams et al., 2020] and Uniswap v3 [Adams et al., 2021]. Both are CPMMs, with the main difference that the reserves used in the invariant function are *real* reserves in Uniswap v2 and *virtual* reserves in Uniswap v3. In order to stick with more common notations, from now on we will denote the reserves of tokens X and Y by x and y respectively, instead of R_X and R_Y. The constant product invariance rule writes as

$$\mathscr{I}(x, y) = x \cdot y = L^2,$$

where L is called the *liquidity*. It is also standard to work with the marginal price of one token X in units of Y (as for usual FX markets) and its square root. We denote these quantities by p (the $X - Y$

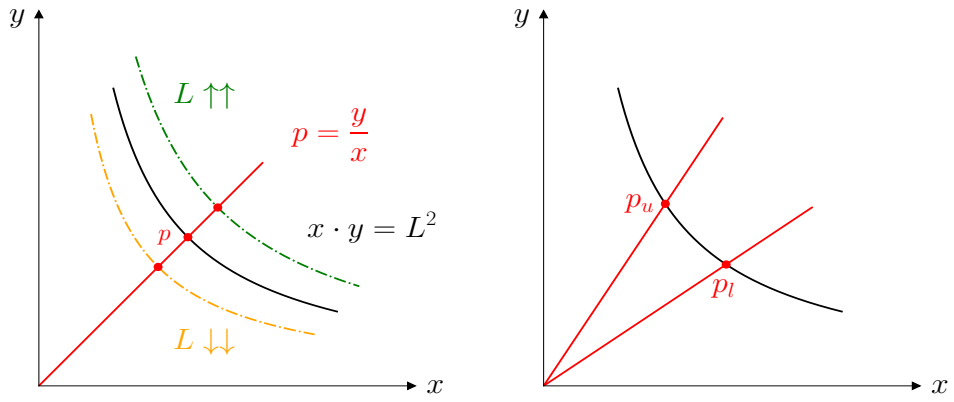

Figure 1: On the left: Representation of how prices and quantities evolve in a Uniswap protocol. On the right: representation of a range of liquidity for prices between p_ℓ and p_u in Uniswap v3.

exchange rate) and π respectively, they are defined by

$$p = \frac{y}{x} \quad \text{and} \quad \pi = \sqrt{\frac{y}{x}}.$$

The constant product rule is represented on the left-hand side of Figure 1. The black hyperbola represents the possible quantities of tokens X and Y that can be available in the pool for a given amount of liquidity. Traders swapping these tokens are constrained to making quantities move along this hyperbola. The corresponding price for some given quantities (x, y) is the slope of the line linking $(0,0)$ to (x, y).

The main feature that distinguishes the Uniswap v3 protocol from Uniswap v2 is that LPs can specify a price range on which they provide liquidity. In other words, contrary to Uniswap v2 where the liquidity provided by an LP can be used for any swap in the price range $(0,\infty)$, an LP providing liquidity to a Uniswap v3 pool can specify a lower-bound price p_ℓ and an upper-bound price p_u (where $p_\ell < p_u$) such that their liquidity can only be used on swaps within the price range $[p_\ell, p_u)$ (or, equivalently, on the square root price range $[\pi_\ell, \pi_u)$). Thus, an analysis of swap fees in Uniswap v3 is more involved.

1.3 Our contributions

Our main contributions are the following.

- In Theorem 3.2, we prove that if a LP provides the liquidity ΔL on a unitary square root price range $R = [\pi_\ell, \pi_u)$, then the Y-value $V_P(t)$ of their position in the pool (net of swap fees) at a future date t is

$$V_P(t) = \Delta L \cdot \left(\left(\frac{1}{\pi_t^R} - \frac{1}{\pi_u} \right) \cdot \pi_t^2 + (\pi_t^R - \pi_\ell) \right),$$

given as a function of the square root price π_t. The definition of unitary range is given in Section 2. The notation $\pi \mapsto \pi^R$ consists of projecting the square root price π onto the square root price range:

$$\pi^R = \begin{cases} \pi_u \text{ if } \pi_u < \pi, \\ \pi_\ell \text{ if } \pi < \pi_\ell, \\ \pi \text{ otherwise.} \end{cases} \tag{1}$$

- The result can be extended to the case where the LP adds a liquidity curve $(\Delta L_\pi)_\pi$ to the pool (Theorem 3.3). In this case, the Y-value $V_P(t)$ of their position (net of swap fees) at a future date t is

$$V_P(t) = p_t \cdot \int_{\pi_t}^{+\infty} \frac{\Delta L_\pi}{\pi^2} \mathrm{d}\pi + \int_0^{\pi_t} \Delta L_\pi \mathrm{d}\pi.$$

  This allows to easily analyze the Greeks of the position (Corollary 3.4). This formula holds regardless of the actions swap traders or other LPs perform.

- Conversely, any concave payoff (as a function of p_t) can be replicated by a providing liquidity with an explicit liquidity curve $(\Delta L_\pi)_\pi$. The accuracy of replication depends on the regularity of the payoff and the size of unitary ranges. See Theorem 3.5.

- In Theorem 4.1 we prove that the amount of fees in tokens X and Y collected over the period $[0, T]$ by an LP that provided a liquidity curve $(\Delta L_\pi)_\pi$ at time 0 is approximated by the following formulas:

$$\mathrm{Fees}^X_{0\to T} \approx \frac{\phi}{(1-\phi)\cdot(\beta_\mathrm{p}-1)} \cdot \int_0^{+\infty} \Delta L_{b^{\frac{1}{2}}} \frac{A^b_T(p)}{4\cdot b^{5/2}} \mathrm{d}b,$$
$$\mathrm{Fees}^Y_{0\to T} \approx \frac{\phi}{(1-\phi)\cdot(\beta_\mathrm{p}-1)} \cdot \int_0^{+\infty} \Delta L_{b^{\frac{1}{2}}} \frac{A^b_T(p)}{4\cdot b^{3/2}} \mathrm{d}b,$$

  in the limit of a tick base $\beta_\mathrm{p} \downarrow 1$. Here ϕ is the swap fee rate; $A^b_T(p)$ is the local time at time T and level b of the price process p (which we assume to be a general ItÃt' semimartingale), it measures the amount of time spent around the level b by p. The resulting formula is quite intuitive: the larger the liquidity provided on a range and the longer time of the price in that range, the larger the collected fees.

- As a consequence (Corollary 4.1), we connect the values of fees to option prices: namely, we establish a relation between the risk-neutral value of the total collected fees and an integral of call/put option prices across different strikes. Investigations around these relations on real data are left to future research.

Some of our results rely on the hypothesis $\mathbf{H}_0$ that the tokens kept outside the pool can be valued using the pool price p [Milionis et al., 2022, Tangri et al., 2023]. This is a standard practice, although such a valuation does not take transaction costs or price slippage into account.

1.4 Comparison with the literature

The authors of [Jaimungal et al., 2023] study optimal execution in pools and numerically solve a stochastic control problem using deep neural networks. In [Cartea et al., 2023a], the issue of optimal liquidity provision is investigated and the authors assume some CIR dynamics for the pool fee rate. The authors also model the dynamics of Impermanent Loss under some specific price assumptions, whereas we obtain this Impermanent Loss in its full generality. Other lines of research on liquidity provision in Uniswap pools and in other protocols are considered in [Fan et al., 2023] and [Cartea et al., 2023b] respectively. In particular, the authors of [Fan et al., 2022] give an expression of a liquidity provider's profit and loss in Uniswap v2 and v3 protocols, depending on the sequence of price changes. Their expression is not easily tractable since it depends on the path of the price. As a main difference, our analysis leads to closed-form approximations that are simpler to handle.

The authors in [Loesch et al., 2021] provide an empirical investigation of how Uniswap v3 pools behave. The probabilistic dynamics of Impermanent Loss is briefly studied in [Boueri, 2022] under some simplified assumptions (the liquidity is assumed to be constant for instance).

The work in [Cartea et al., 2022] gives rise to a new class of trading problems about how to optimally trade a large position and execute statistical arbitrages based on market signals. They design some strategies using stochastic optimal tools in the context of Uniswap v2; data from Uniswap v3 are also used. The work closest in its goals to ours is [Bichuch and Feinstein, 2023], where the authors announce an asymptotic analysis of Uniswap v2 and v3 fees, assuming the price follows a geometric random walk with exponential time stepping. In the current work, we derive an asymptotic formula for an arbitrary continuous price process and the theoretical result is confirmed by numerical experiments. At the time of writing this article, the work from [Bichuch and Feinstein, 2023] is not available.

To summarize, the results presented in this paper are quite novel compared to existing approaches, both on the considered issues and on the level of generality at which they are obtained. In the community, there is some consensus that deriving tractable formulas for Uniswap v3 is not possible without strong assumptions on, e.g., the Pool-Spot spread (supposed to be equal to 0) or the behavior of other LPs. Our analysis shows that we do not need these assumptions, which is a significant progress in the quantitative understanding of Uniswap v3.

1.5 Organisation of the paper

The outline of the paper is as follows. In Section 2 we review the exact mechanisms of the Uniswap v3 protocol, with an analysis based on the source code of Uniswap v3[1]. Our goal is to precisely define the way the protocol behaves, so that both practitioners and academic researchers can agree on the same set of rules when working on Uniswap v3. Section 3 is dedicated to an analysis of *Impermanent Loss*, which corresponds to the potential loss incurred by an LP providing liquidity on a price range. We give rigorous derivation of several known formulas and extend some of them. The most enlightening formula is that of Theorem 3.3, which gives the value of the LP position (net of fees) as a function of the provided liquidity curve $(\Delta L_\pi)_\pi$, showing that it is a concave payoff.

[1] The source code is available at https://github.com/Uniswap/v3-core/.

Conversely, any concave payoff can be replicated by providing some liquidity curve $(\Delta L_\pi)_\pi$. Then, in Section 4 we investigate the fees collected by an LP: we obtain explicit formulas in terms of the time spent by the pool price in different ranges (through a local times based formula) and make the connection with call/put pricing thanks to the occupation time formula. Some of the proofs are postponed to the Appendix.

2 Core operations in a Uniswap v3 pool

2.1 Real and virtual token reserves

The main feature that distinguishes the Uniswap v3 protocol from Uniswap v2 is that LPs can specify a price range on which they provide liquidity. In other words, contrary to Uniswap v2 where the liquidity provided by an LP can be used for any swap in the price range $(0,\infty)$, an LP providing liquidity to a Uniswap v3 pool can specify a lower-bound price p_ℓ and an upper-bound price p_u (where $p_\ell < p_u$) such that their liquidity can only be used on swaps within the price range $[p_\ell, p_u)$. After several LPs have provided liquidity to the pool, each time with a specific price range, the liquidity distribution on the entire price space can have an arbitrary form (see Figure 3-III in [Adams et al., 2021] and Figure 3 for an illustration).
The principle of a Uniswap v3 pool is the following. An LP providing an amount of liquidity L on a price range $[p_\ell, p_u)$ deposits respective quantities x_r and y_r of tokens X and Y into the pool. These tokens are used by swap operations as long as the price is in the range $[p_\ell, p_u)$. When the price reaches p_ℓ, all reserves of tokens Y will have been depleted, and when the price reaches p_u, all reserves of token X will have been depleted. On the price range $[p_\ell, p_u)$, the constant product rule $x \cdot y = L^2$ applies[2], with the main difference that x and y denote *virtual* token reserves instead of real reserves. The virtual quantities x and y can be decomposed as the sum of the real number of tokens in the pool and other quantities that we call offsets:

$$\begin{cases} x = x_r + x_{\text{offset}}, \\ y = y_r + y_{\text{offset}}. \end{cases}$$

The graph in Figure 2 depicts a new coordinate system in blue which describes the two extreme cases where the reserves of real tokens X and Y have been depleted (red points with respective coordinates (x_ℓ, y_ℓ) and (x_u, y_u) in the Figure). At these points we have

$$x_u = 0 + x_{\text{offset}} = x_{\text{offset}} \qquad \text{and} \qquad y_\ell = 0 + y_{\text{offset}} = y_{\text{offset}}.$$

Because the constant product rule applies at both points, we have $x_\ell \cdot y_\ell = x_u \cdot y_u = L^2$, $p_\ell = \frac{y_\ell}{x_\ell}$ and $p_u = \frac{y_u}{x_u}$. This permits to deduce the values of x_{offset} and y_{offset}:

$$x_{\text{offset}} = \frac{L}{\sqrt{p_u}} = \frac{L}{\pi_u} \quad \text{and} \quad y_{\text{offset}} = L\sqrt{p_\ell} = L \cdot \pi_\ell.$$

[2] Note that it would be more accurate to write $L(p_\ell, p_u)$ instead of L to reflect the fact that liquidity depends on a price range.

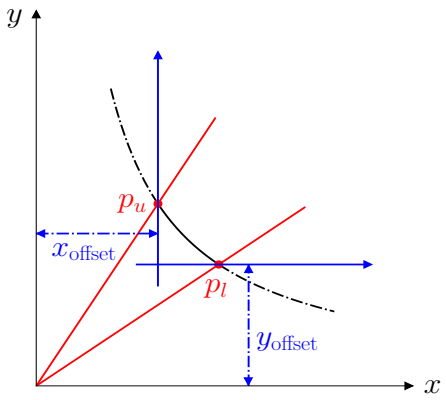

Figure 2: Representation of the *virtual* numbers (x, y) and the *offset* numbers $(x_{\text{offset}}, y_{\text{offset}})$ of tokens.

Thus, on the price range $[p_\ell, p_u)$, the real reserves x_r, y_r of tokens X and Y and the liquidity L are related by the equation

$$\left(x_r + \frac{L}{\pi_u}\right) \cdot (y_r + L \cdot \pi_\ell) = L^2. \tag{2}$$

The price that is induced by the amounts x_r and y_r is given by

$$\pi^2 = p = \frac{y_r + y_{\text{offset}}}{x_r + x_{\text{offset}}} = \frac{y_r + L \cdot \pi_\ell}{x_r + \frac{L}{\pi_u}}. \tag{3}$$

We can derive expressions of the amounts of real tokens available as follows. We have

$$\pi^2 \cdot \left(x_r + \frac{L}{\pi_u}\right) \overset{(3)}{=} y_r + L \cdot \pi_\ell \overset{(2)}{=} \frac{L^2}{x_r + \frac{L}{\pi_u}} \overset{(2)}{=} \pi^2 \cdot \frac{L^2}{y_r + L \cdot \pi_\ell},$$

so that $\pi \cdot \left(x_r + \frac{L}{\pi_u}\right) = L$ and $y_r + L \cdot \pi_\ell = L \cdot \pi$. Therefore,

$$x_r + \frac{L}{\pi_u} = \frac{L}{\pi} \quad \text{and} \quad y_r + L \cdot \pi_\ell = L \cdot \pi. \tag{4}$$

Note that since $x_r \geq 0$ and $y_r \geq 0$, necessarily, $\pi_\ell \leq \pi \leq \pi_u$. The maximum quantities of tokens X and Y within a range for a fixed liquidity L are also entailed by these formulas, they are respectively $L \cdot \left(\frac{1}{\pi_\ell} - \frac{1}{\pi_u}\right)$ and $L \cdot (\pi_u - \pi_\ell)$.

Prices and ranges in a Uniswap v3 pool. In Uniswap v3 pools, the pool price can evolve with arbitrary values (see Equation (20) for a modeling example), but the fees owed to each LP depend on all price ranges in which this price evolves. The bounds of these ranges are a discrete set of price values that are called *ticks*. A tick is an integer $\tau \in \mathbb{Z}$ to which is associated the price $p(\tau) = 1.0001^\tau$.

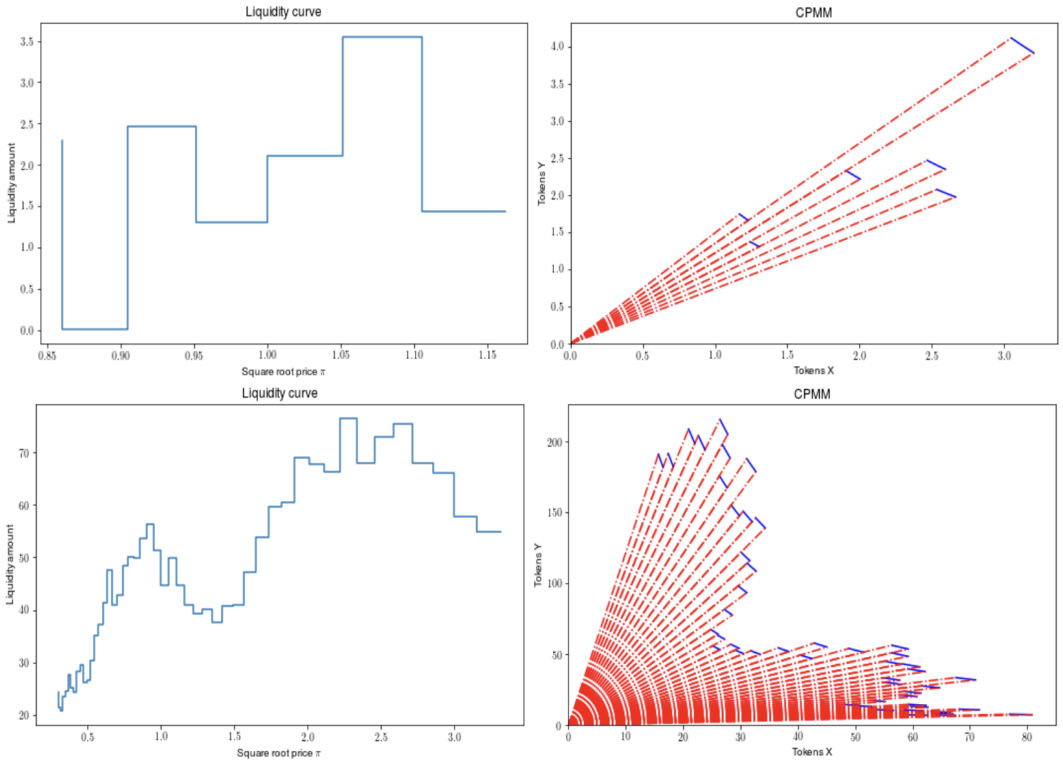

Figure 3: Different liquidity quantities deposited across several consecutive square root price ranges (on the left). On the right, the representation of the CPMM formula for each range (with its specific liquidity value). Top and bottom: pools with different liquidity distributions.

In other words, a tick can be viewed as the logarithm in base $\beta_p = 1.0001$ of a price. In practice, a Liquidity Provider does not actually specify a price range on which liquidity is to be added, but rather a tick range. Not all tick ranges can actually be selected to add liquidity: the ranges are a multiple of a fixed number of ticks δ_π which is determined at the setting of the pool, depending on the swap fees ϕ. In the default setting, we have $\delta_\pi = 10, 60, 200$, depending on the value of swap fees $\phi = 0.05\%, 0.3\%, 1\%$ (see the constructor and `createPool` method in UNISWAPV3FACTORY.SOL), but it is possible to use the smart contract to create pools with other tick spacings and swap fees. The lower and upper ticks of a tick range have to be tick indices $i \cdot \delta_\pi$ for $i \in \mathbb{Z}$. When a range is defined by two consecutive ticks $i \cdot \delta_\pi$ and $(i+1) \cdot \delta_\pi$, we refer to it as a *unitary range*.
Figure 3 depicts a possible landscape of liquidity at some time in the pool, and its Constant Product Market Making formula range by range (pieces of hyperbola). This is a generalization of Figure 1.
In what follows, for the sake of clarity, we disregard the tick spacings and values of ticks, and we focus on ranges defined by square root prices. This choice will permit to simplify several mathematical expressions.

2.2 Updating liquidity in the pool

Assume an LP wishes to add or remove liquidity to the pool on the square root price range $[\pi_\ell, \pi_u)$ at a time where the square root price in the pool is π_0. We first assume that this liquidity event occurs on a unitary range, which entails that the available liquidity on the range is constant. The results will be generalized to arbitrary ranges afterwards. The existing liquidity and real token quantities are denoted by L, x_r, y_r on the given square root price range; the updates in liquidity and token quantities are denoted by $\Delta L, \Delta x_r$ and Δy_r.

When there is no liquidity available on the price range. In this case we have $L = 0$ and $x_r = y_r = 0$. The only possible operation thus consists in the addition of liquidity to the pool, which is also referred to as a *mint* operation. The principle of this mint operation is that it is supposed to keep the current square root price unchanged. Depending on the current square root price π_0, we have the following cases:

(1) If $\pi_u < \pi_0$, then there are no reserves of token X to be added ($\Delta x_r = 0$), and Equation (2) writes as $\Delta L^2 = \frac{\Delta L}{\pi_u} \cdot (\Delta y_r + \Delta L \cdot \pi_\ell)$, from which we deduce that $\Delta y_r = \Delta L \cdot (\pi_u - \pi_\ell)$.

(2) If $\pi_0 < \pi_\ell$ then there are no reserves of token Y to be added ($\Delta y_r = 0$), and Equation (2) writes as $\Delta L^2 = \left(\Delta x_r + \frac{\Delta L}{\pi_u}\right) \cdot \Delta L \cdot \pi_\ell$, from which we deduce that $\Delta x_r = \Delta L \cdot \left(\frac{1}{\pi_\ell} - \frac{1}{\pi_u}\right)$.

(3) In the remaining case, using Equation (4), we directly obtain $\Delta x_r = \Delta L \cdot \left(\frac{1}{\pi_0} - \frac{1}{\pi_u}\right)$ and $\Delta y_r = \Delta L \cdot (\pi_0 - \pi_\ell)$.

These equations can be summarized into a single equation as follows. Given a unitary square root price range $R = [\pi_\ell, \pi_u)$ and a square root price π_0, consider the square root price π_0^R defined in Equation (1). We have

$$\Delta x_r \;=\; \Delta L \cdot \left(\frac{1}{\pi_0^R} - \frac{1}{\pi_u}\right) \quad \text{and} \quad \Delta y_r \;=\; \Delta L \cdot (\pi_0^R - \pi_\ell). \tag{5}$$

Example 2.1 (Adding liquidity to an empty Uniswap v3 pool)**.** *Consider a newly initialized liquidity pool on base asset X and quote asset Y, with a tick spacing set at $\delta_\pi = 60$, and an initial price set at $p_0 = 3019$. Assume a liquidity provider is about to deposit an amount $\Delta L_1 = 150\,000$ of liquidity on the tick range $[80100, 80160)$, which corresponds to the price range $[3009.71\cdots, 3027.82\cdots)$. The number of tokens to deposit can be deduced from Equation (5): the LP will deposit $\Delta x_r = 3.98\cdots$ tokens X and $\Delta y_r = 12\,688.39\cdots$ tokens Y.*

When there is already liquidity on the square root price range. Assume there is already an amount of liquidity $L > 0$ that is available on a given square root price range $[\pi_\ell, \pi_u)$, corresponding to the (real) reserves x_r and y_r, and that a new liquidity amount ΔL is to be added/removed to the same range, corresponding to (real) quantities Δx_r and Δy_r of tokens X and Y. It may be the case that $\Delta L < 0$ but regardless, we have $L + \Delta L \geq 0$. We first consider a unitary price range, so that the liquidity L already available on the range is constant. The quantities Δx_r and Δy_r are determined as follows.

(1) If $\pi_u < \pi_0$, then we have $x_r = \Delta x_r = 0$, and $y_r \overset{(2)}{=} L \cdot (\pi_u - \pi_\ell)$. Again, Equation (2) entails that $(L+\Delta L)^2 = \frac{L+\Delta L}{\pi_u} \cdot \left(y_r + \Delta y_r + (L+\Delta L) \cdot \pi_\ell\right)$, from which we deduce that $y_r + \Delta y_r + (L+\Delta L) \cdot \pi_\ell = (L+\Delta L) \cdot \pi_u$. Thus, replacing y_r by its value, we obtain

$$\Delta y_r = (L+\Delta L) \cdot (\pi_u - \pi_\ell) - y_r = \Delta L \cdot (\pi_u - \pi_\ell).$$

(2) If $\pi_0 < \pi_\ell$, then we have $y_r = \Delta y_r = 0$ and $x_r \overset{(2)}{=} L \cdot (\frac{1}{\pi_\ell} - \frac{1}{\pi_u})$. Using Equation (2), we have $(L+\Delta L)^2 = \left(x_r + \Delta x_r + \frac{L+\Delta L}{\pi_u}\right) \cdot (L+\Delta L) \cdot \pi_\ell$. After simplifying by $L+\Delta L$, we obtain

$$\Delta x_r = (L+\Delta L) \cdot \left(\frac{1}{\pi_\ell} - \frac{1}{\pi_u}\right) - x_r = \Delta L \cdot \left(\frac{1}{\pi_\ell} - \frac{1}{\pi_u}\right).$$

(3) Otherwise we have $x_r \overset{(4)}{=} L \cdot \left(\frac{1}{\pi_0} - \frac{1}{\pi_u}\right)$ and $y_r \overset{(4)}{=} L \cdot (\pi_0 - \pi_\ell)$. Because the square root price is meant to remain constant after the mint or burn operation, the same proportionality relations hold between $x_r + \Delta x_r$ and $L + \Delta L$, and between $y_r + \Delta y_r$ and $L + \Delta L$. This permits to deduce that

$$\Delta x_r = \Delta L \cdot \left(\frac{1}{\pi_0} - \frac{1}{\pi_u}\right) \text{ and } \Delta y_r = \Delta L \cdot (\pi_0 - \pi_\ell).$$

Remark 2.2. The computations above show that Equation (5) always holds, regardless of whether or not there is already liquidity available on the considered square root price range and regardless of the sign of ΔL, as long as $L + \Delta L \geq 0$.

When a liquidity provider deposits liquidity on several ranges, the total number of tokens required is deduced by additivity. More specifically, if a liquidity provider deposits ΔL_i on range $R_i \overset{\text{def}}{=} [\pi_{\ell_i}, \pi_{u_i})$ for $i = 1, \ldots, n$, then total amounts of tokens X and Y that are required are given by

$$\Delta x_r = \sum_{i=1}^{n} \Delta L_i \cdot \left(\frac{1}{\pi_0^{R_i}} - \frac{1}{\pi_{u_i}}\right) \text{ and } \Delta y_r = \sum_{i=1}^{n} \Delta L_i \cdot \left(\pi_0^{R_i} - \pi_{\ell_i}\right). \tag{6}$$

In the particular case where the provided liquidity is the same on all ranges and the latter are consecutive, Equation (6) can be simplified further as follows:

Proposition 2.3. *Consider a square root price range $R = [\pi_\ell, \pi_u)$ that is of the form $R_1 \cup \ldots \cup R_n$ for $n \geq 1$, where for $i \leq n$, $R_i = [\pi_{\ell_i}, \pi_{u_i})$ contains an amount L_i of liquidity, and for $i < n$, $\pi_{u_i} = \pi_{\ell_{i+1}}$. A liquidity provider who deposited (resp. withdrew) a same amount of liquidity ΔL on each range R_i, where $L_i + \Delta L \geq 0$ for all i, will deposit (resp. withdraw) the following amounts of tokens into the pool:*

$$\Delta x_r = \Delta L \cdot \left(\frac{1}{\pi_0^R} - \frac{1}{\pi_u}\right) \text{ and } \Delta y_r = \Delta L \cdot \left(\pi_0^R - \pi_\ell\right). \tag{7}$$

When this is the case, we can therefore say that the liquidity provider deposited an amount of liquidity ΔL on range R.

The proof of this result is postponed to Appendix A.1. This proposition provides the justification why the Uniswap v3 contracts track so-called *positions* for each LP. When an LP with address[3] α deposits liquidity on a (not necessarily unitary) range R, the protocol creates a position $\langle \alpha, R \rangle$ to which is associated a state consisting of: i) the liquidity that the LP owns on the range; ii) a tracker of the amount of tokens X owed to the LP due to fees and iii) a tracker of the amount of tokens Y owed to the LP due to fees (see the contract POSITION.SOL). The amounts of tokens to deposit or withdraw do not depend on the liquidity that is available on each unitary range but only on the bounds of the range and the price in the pool when the update is performed; this is translated in the code of the `_modifyPosition` method in UNISWAPV3POOL.SOL.

Example 2.4 (Updating the liquidity in a pool)**.** *Following Example 2.1, assume another liquidity provider is about to deposit the same amount of liquidity* $\Delta L_2 = 75\,000$ *on the consecutive tick ranges* $[80100, 80160)$ *and* $[80160, 80220)$*. The amount of tokens to deposit on each range is given by Equation (7): The liquidity provider will deposit*

- $1.99\cdots$ *tokens* X *and* $6344.19\cdots$ *tokens* Y *on the tick range* $[80100, 80160)$*, and*
- $4.08\cdots$ *tokens* X *and no token* Y *on the tick range* $[80160, 80220)$*.*

Note that the total amount of tokens to deposit could also have been derived by applying Equation (7) on the tick range $[80100, 80220)$*.*

2.3 Swapping tokens

The swapping process in a Uniswap v3 pool is best understood by considering the relationship between the price evolution in the pool, the available liquidity and the amounts of tokens – real or virtual – that are available in the pool. This process is more involved than that of a v2 pool because the available liquidity depends on the considered price range and when swapping tokens, the current price may *cross* from one price range to another one with a different amount of available liquidity. The principle of the algorithm is based on the following observations.

- Assume that the current square root price π is on a range with available liquidity L. Assume further that a swap operation, trading quantities Δx of tokens X and Δy of tokens Y, is performed with the guarantee that this operation does not make the current square root price cross to a range with a different amount of liquidity. This operation causes the virtual reserves of token X (resp. token Y) to become $x' = x + \Delta x$ (resp. $y' = y + \Delta y$), and the square root price to become $\pi' = \frac{y'}{x'}$. The following relationships can be derived using Equation (4):

$$\Delta x = x' - x = \frac{L}{\pi'} - \frac{L}{\pi}, \tag{8}$$

$$\Delta y = y' - y = L \cdot \pi' - L \cdot \pi. \tag{9}$$

- Assume the current price is within the range $[\pi_\ell, \pi_u)$. Then the maximum amounts of tokens that can be traded before the price crosses into another range are given by the following:

[3] In this context, the address of an LP is the (unique) identifier of the wallet that is used to transfer tokens to and from the Uniswap v3 pool.

(1) The maximum amount of tokens X that can be traded in the range is $\Delta x_m \overset{(8)}{=} L\cdot\left(\frac{1}{\pi_\ell}-\frac{1}{\pi}\right)$, and afterwards the current square root price is π_ℓ.

(2) The maximum amount of tokens Y that can be traded in the range is $\Delta y_m \overset{(9)}{=} L\cdot(\pi_u-\pi)$, and afterwards the current square root price is π_u.

We provide a high-level overview of the swap operation in Algorithm 1. Some of the features of the actual algorithm are left out for the sake of simplicity. For example, the actual algorithm allows users to specify whether the quantity provided as an input is an exact amount of tokens traded into the pool, or an exact amount of tokens traded out of the pool; it also allows users to specify a limit slippage price that, if reached, interrupts the transaction. The actual algorithm is optimized to iterate through so-called *initialized* square root prices that correspond to range bounds on which liquidity has been deposited and thus potentially changes. We assume that these initialized square root prices are all separated by the corresponding tick spacing δ_π. This is without loss of generality since depositing the same amount of liquidity on consecutive ranges is equivalent to depositing this liquidity on the union of these ranges, and this assumption simplifies the algorithm description because it is guaranteed that liquidity is constant between consecutive initialized square root prices. We denote this sequence of square root prices by $\pi_0 < \pi_1 < \cdots < \pi_m < \cdots$ and we denote by L_i the liquidity that is available between π_{i-1} and π_i. We assume that there is enough liquidity available in the entire pool for the swap to take place; otherwise, the transaction fails.
The swap operation can be represented by a function that takes as inputs a quantity of tokens to swap and a direction $d \in \{-1,1\}$, denoting which token is traded in and which token is traded out (token Y is traded in when $d=1$, causing the price in the pool to increase, and token X is traded in when $d=-1$). The function also transfers the required amounts of tokens in and out of the pool. We define a function Liq that computes the available liquidity for a given square root price. This function depends on a direction that is used when a liquidity change occurs at the considered square root price. The function is defined as follows:

$$\text{Liq}(\pi,d) = \begin{cases} L_{i+1} & \text{if } \pi_i < \pi < \pi_{i+1}, \\ L_{i+1} & \text{if } \pi = \pi_i \text{ and } d = 1, \\ L_i & \text{if } \pi = \pi_i \text{ and } d = -1. \end{cases}$$

Fees bookkeeping

Contrarily to Uniswap v2 pools, fees are not considered as additional tokens in the reserves of a Uniswap v3 pool and thus, they do not increase the liquidity in the pool. Given a range R, the *fees per unit of liquidity* in the pool are tracked by two accumulators Φ_R^X and Φ_R^Y that are updated at every transaction. More precisely, let $\Phi_R^d = \Phi_R^Y$ if $d=1$ in a swap operation (meaning that tokens Y are traded into the pool in exchange for tokens X), and $\Phi_R^d = \Phi_R^X$ if $d=-1$. The value of Φ_d is updated immediately after Line 9 in Algorithm 1 by the instruction

$$\Phi_R^d \longleftarrow \Phi_R^d + \frac{a_i}{1-\phi}\cdot\frac{\phi}{L}. \tag{10}$$

Algorithm 1: A high-level overview of the swap algorithm in a Uniswap v3 pool.

input : $q \geq 0$: an amount of tokens
d: the direction of the swap

$q_i \longleftarrow 0$; $q_o \longleftarrow 0$ // initialize the token amounts to receive and transfer
while $q \neq 0$ **do**
  $\pi' \longleftarrow \min\left\{\pi_j \mid \pi_j^d > \pi^d\right\}$ // get the next square root price in the sequence
  $L \longleftarrow \text{Liq}(\pi, d)$ // get the liquidity in the current range for the given direction
  if $L \neq 0$ **then**
    $a_i \longleftarrow \min\left\{q \cdot (1-\phi), L \cdot \left((\pi')^d - \pi^d\right)\right\}$ // compute the amount of input tokens to use in the swap on the current range
    $\pi_q \longleftarrow \left(\pi^d + \frac{a_i}{L}\right)^d$ // compute the price (using (4)) that is reached when a_i input tokens have been swapped
    $a_o \longleftarrow L \cdot \left(\pi^{-d} - (\pi_q)^{-d}\right)$ // compute the corresponding amount of output tokens obtained from the swap on the current range
    $q \longleftarrow q - \frac{a_i}{1-\phi}$ // update the amount of input tokens in the while loop
    $q_o \longleftarrow q_o + a_o$ // update the total amounts of tokens to be traded out
    $\pi \longleftarrow \pi_q$ // update the current square root price
  end
  else
    $\pi \longleftarrow \pi'$ // move directly to the next square root price
  end
end
receive q from trader and transfer q_o

Table 1: First swap operation in Example 2.5: 4 tokens X are transferred into the pool.

Tick range	$[80100, 80160)$	$[80160, 80220)$
Tokens X swapped in	4	0
Tokens Y swapped out	$12028.05\cdots$	0
Fees per liq. (in X tokens)	$5.33\cdots 10^{-8}$	0

Table 2: Second swap operation in Example 2.5: 40 000 tokens Y are transferred into the pool.

Tick range	$[80100, 80160)$	$[80160, 80220)$
Tokens Y swapped in	$30\,170.78\cdots$	$9\,829.21\cdots$
Tokens X swapped out	$9.95\cdots$	$6.54\cdots$
Fees per liq. (in Y tokens)	$4.02\cdots 10^{-4}$	$3.93\cdots 10^{-4}$

Recall that L represents the available liquidity on the considered range and $\frac{a_i}{1-\phi}$ represents the amount of input tokens on the considered range; hence $\frac{a_i \cdot \phi}{1-\phi}$ represents the fees that are accumulated and $\frac{a_i}{1-\phi} \cdot \frac{\phi}{L}$ is the amount of accumulated fees per unit of liquidity on the range. This point is important for our subsequent analysis of fees in Section 4.

The actual implementation of the Uniswap v3 contract does not store the accumulators Φ_R^X and Φ_R^Y for each range R on which liquidity was deposited. Instead, it stores global accumulators Φ^X (`feeGrowthGlobal0X128` in the source code) and Φ^Y (`feeGrowthGlobal1X128` in the source code) for the entire pool; along with state variables for both tokens `feeGrowthOutside0X128` and `feeGrowthOutside0X128` at each square root price π that is a range boundary. These state variables can be used to compute the quantities $\varphi_{\mathrm{a}}^X(\pi)$ and $\varphi_{\mathrm{a}}^Y(\pi)$ (resp. $\varphi_{\mathrm{b}}^X(\pi)$ and $\varphi_{\mathrm{b}}^Y(\pi)$) that represent the amounts of fees per unit of liquidity earned in ranges above (resp. below) π starting at the time this square root price was initialized as a range boundary. The accumulators described above for range $R = [\pi_\ell, \pi_u)$ corresponding to a position are recovered by the equations

$$\Phi_R^X = \Phi^X - \varphi_{\mathrm{b}}^X(\pi_\ell) - \varphi_{\mathrm{a}}^X(\pi_u) \quad \text{and} \quad \Phi_R^Y = \Phi^Y - \varphi_{\mathrm{b}}^Y(\pi_\ell) - \varphi_{\mathrm{a}}^Y(\pi_u),$$

which are implemented in the `getFeeGrowthInside` method of TICK.SOL and invoked when the position is updated (such as in the `_updatePosition` method of UNISWAPV3POOL.SOL).

Example 2.5. *Swap operations Consider the pool from Example 2.4, when the current price is* 3019 *(the current tick is* 80130*) and the liquidity on the current tick range* $[80100, 80160)$ *is* 225 000*, and assume a trader transfers* 4 *tokens X into the pool. This will cause the price in the pool to diminish, and it is straightforward to verify that the tick after the swap remains in the same range, the other range is thus unaffected by the swap operation. The main quantities involved in the swap operation are summarized in Table 1, the current tick after the swap operation is* 80111.

Assume a second trader now transfers 40 000 *tokens Y into the pool, causing the price in the pool to increase. This swap operation will consume all tokens X in the current tick range and move on to the following range* [80160, 80220) *where the available liquidity is* 75 000. *The amounts of tokens input and output during the swap operation along with the fees per unit of liquidity that are accumulated are summarized in Table 2, the current tick after the swap is* 80 207.

2.4 Withdrawing tokens from the pool

An LP who deposited liquidity into the pool at time t_0 on a range R is owed at time t certain amounts of tokens X and Y. These amounts are accumulated every time a swap operation occurs within the range between times t_0 and t, as described above. The total amount of fees (per unit of liquidity) accumulated between times t_0 and t are respectively $\Phi_R^X(t) - \Phi_R^X(t_0)$ and $\Phi_R^Y(t) - \Phi_R^Y(t_0)$. If the LP wishes to *burn* an amount ΔL of liquidity on the range R, then as discussed in Subsection 2.2, the amounts of tokens X and Y retrieved from the pool are given by Equation (7). The LP will also withdraw tokens that were earned as fees during swap operations on range R; the amounts of such tokens that are withdrawn are given by:

$$\Delta x_{\text{fee}} = \Delta L \cdot \left(\Phi_R^X(t) - \Phi_R^X(t_0)\right) \quad \text{and} \quad \Delta y_{\text{fee}} = \Delta L \cdot \left(\Phi_R^Y(t) - \Phi_R^Y(t_0)\right).$$

In the smart contract, the amounts of tokens owed due to swap fees are actually updated every time the considered position is updated by a liquidity event. Two variables, `feeGrowthInside0LastX128` and `feeGrowthInside1LastX128`, permit to keep track of the new tokens that are owed.

Example 2.6. *Token withdrawal Following example 2.4 assume the liquidity provider who had deposited* 75 000 *units of liquidity on ranges* [80100, 80160) *and* [80160, 80220) *wishes to burn* 60 000 *units of liquidity from range* [80100, 80160) *immediately after the swaps of Example 2.5. The respective amounts* Δx_{fee} *and* Δy_{fee} *of tokens X and Y that they recover are given by*

$$\Delta x_{\text{fee}} = 60\,000 \cdot 5.33\cdots 10^{-8} = 3.2 \cdot 10^{-2},$$
$$\Delta y_{\text{fee}} = 60\,000 \cdot 4.02\cdots 10^{-4} = 24.13\cdots.$$

Afterward the liquidity provider still owns 15 000 *units of liquidity on range* [80100, 80160) *and* 75 000 *units of liquidity on range* [80160, 80220).

All these descriptions are valid in full generality on the occurrence of swap trades and LP events.

3 Impermanent loss revisited

3.1 Impermanent Loss

Recall that liquidity providers are actors on DEXes (Decentralized Exchanges) who transfer tokens into liquidity pools and are rewarded by the fees paid by traders who use the pool to swap tokens. Clearly, this trading activity causes the price in the pool to evolve. As we will see, if a liquidity provider withdraws their tokens at a time where the pool price is significantly different from the one at deposit time, the value of their retrieved tokens net of fees will be lower than the value of

their original tokens if they had not been deposited into the pool. This loss is only materialized when the LPs withdraw their tokens from the pool, it is called the *Impermanent Loss*, or *Divergence Loss* (see [Pintail, 2020] for comments) and in what follows, we may use IL as a shorthand for this loss. The Impermanent Loss is formalized by comparing the value of two strategies: the first one consists in depositing a given number of tokens into a pool (the Liquidity providing strategy), and the second one consists in simply keeping the tokens (historically known as the *HODL* strategy). The Impermanent Loss is defined by comparing the value in Y tokens of both strategies (token Y thus plays the role of *reference numéraire*). More precisely, we denote by

- V_P the value in token Y of the portfolio in the case where tokens are transferred to a liquidity pool (Liquidity providing strategy),
- V_H the value in token Y of the portfolio in the case where all tokens are withheld (HODL strategy)

and we define the *absolute Impermanent Loss*, or simply Impermanent Loss, as the quantity

$$V_P - V_H. \tag{11}$$

Remark 3.1. It is common in the literature to find the following definition of an Impermanent Loss: $\text{IL} = \frac{V_P - V_H}{V_H}$. Some articles also define an Impermanent Loss using the equation $\frac{V_P - V_H}{V_H^0}$, where V_H^0 is the value of the initial investment (which is the same for both strategies). Let us also mention the work in [Milionis et al., 2022] where the authors introduce a variant of Impermanent Loss called "loss-versus-rebalancing" (LVR), associated with a rebalancing strategy that replicates the pool trades at market prices. This strategy is studied in the context of Uniswap v2. Throughout this paper, we will stick with the definition in Equation (11), which is more convenient from a mathematical point of view.

The results we derive on the Impermanent Loss rely on the following hypothesis:

H$_0$: *Tokens X and Y outside the pool can be valued using the pool price.*

As mentioned in the Introducion, the assumption is a standard practice.

3.2 Focus on a unitary price range

We consider a liquidity provider who added an amount ΔL of liquidity on a square root price range $R = [\pi_\ell, \pi_u)$, when the price in the pool[4] was p_0 (hence the square root price was π_0), and assume the current price is p_1 (hence the current square root price is π_1). The relative positions of the initial and current square root prices with respect to the range R are arbitrary. First we assume that R is unitary, the extension to an arbitrary range is given in Subsection 3.3.
In the statement below, we make an intensive use of the notation π^R, which denotes the projection of the square root price π onto the range R, see Equation (1) for a precise definition.

[4] In the whitepaper [Adams et al., 2021], the initial price p_0 is named the current price and denoted by p_c. Since several dates occur in our analysis of Impermanent Loss, we believe the notations p_0, π_0 are clearer.

Theorem 3.2. *Assume $\boldsymbol{H}_0$. Consider a unitary square root price range $R = [\pi_\ell, \pi_u)$ and consider the following two strategies.*

- Liquidity providing strategy. *An amount of liquidity ΔL, corresponding to token quantities Δx_r and Δy_r is added to the range R at time $t = 0$. Then at time t_1, the Y-value of the tokens that can be recovered from the pool is*

$$V_P = \Delta L \cdot \left(\left(\frac{1}{\pi_1^R} - \frac{1}{\pi_u} \right) \cdot \pi_1^2 + \left(\pi_1^R - \pi_\ell \right) \right), \tag{12}$$

  given as a function of the square root price π_1.

- HODL strategy. *The token quantities Δx_r and Δy_r are held outside of the pool. Then at time t_1, the Y-value of these tokens is*

$$V_H = \Delta L \cdot \left(\left(\frac{1}{\pi_0^R} - \frac{1}{\pi_u} \right) \cdot \pi_1^2 + \left(\pi_0^R - \pi_\ell \right) \right). \tag{13}$$

Furthermore, the Impermanent Loss at time t_1 is given by

$$V_P - V_H = -\Delta L \cdot \left(\pi_0^R - \pi_1^R \right) \cdot \left(1 - \frac{\pi_1^2}{\pi_0^R \cdot \pi_1^R} \right), \tag{14}$$

and this quantity is never positive.

The proof is postponed to Appendix A.2.
Note that these formulas are valid regardless of whether or not there have been swap trades or liquidity events between the time at which the LP deposited liquidity and the time at which their position value is computed. Note also that the value V_P does not directly depend on the initial price, although the amounts of tokens to add to the pool can be related to this initial price depending its relative position to the range under consideration (see the proof). The value V_P essentially depends on the initial deposited liquidity ΔL.
In Figure 4, we plot each of the components V_P, V_H and $V_P - V_H$. The value from Formula (12), written as a function of $p_1 = \pi_1^2$, is depicted in Figure 4 (top plots, in blue). It has the shape of covered call with a smoothing around the strike, as was observed by Guillaume Lambert on his blog [Lambert, 2021]. As can be observed in Figure 4 (bottom plots), the Impermanent Loss $V_P - V_H$ is always non-positive, and it is zero when the current price p_1 coincides with p_0. Except on the (small) range where the liquidity was added, the IL behaves as the opposite (up to a constant factor) of a call or put payoff, depending on whether $\pi_0 > \pi_u$ or $\pi_0 < \pi_\ell$, respectively, with a strike equal to $\pi_\ell \cdot \pi_u$ (i.e. equal to the geometric mean of p_l and p_u).

3.3 A concise formula for the Uniswap v3 strategy with a full liquidity curve

The valuations from Theorem 3.2 can be extended to a full liquidity curve $(\Delta L_i)_i$ spread over the possible unitary ranges $[\pi_{\ell_i}, \pi_{u_i})$, by a straightforward summation of the above formulas. This is summarized in the following statement, where we adopt the following notation:

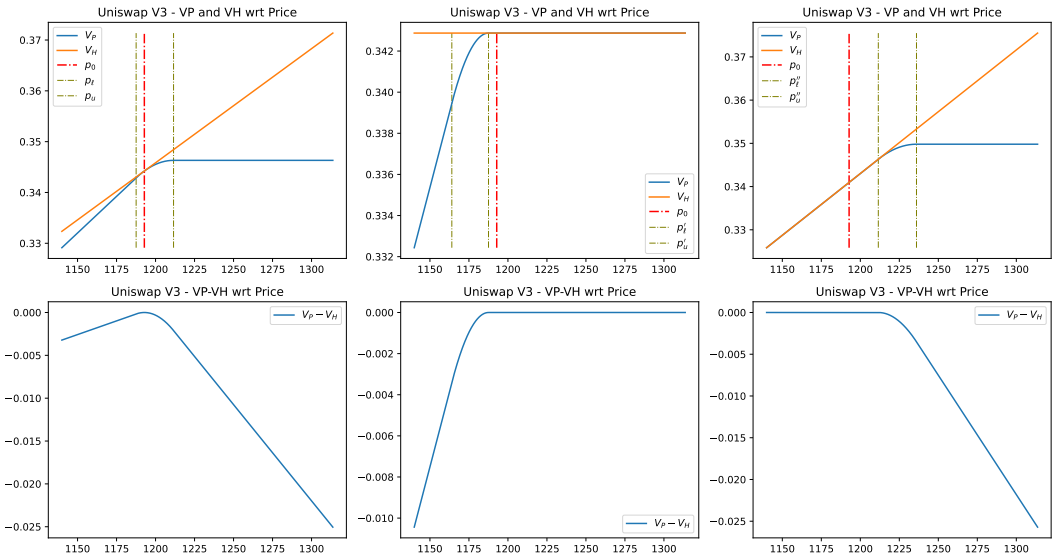

Figure 4: Values at time t_1 of V_P, V_H (top) and $V_P - V_H$ (bottom) for 3 different price ranges. The x-axis corresponds to the price value p_1 at time t_1. The dashed vertical lines in olive color represent the considered unitary ranges. At the initial price p_0, the values V_P and V_H coincide.

- $\underline{\pi}$ and $\overline{\pi}$ denote the square root prices π_ℓ and π_u defined by the unique unitary range $[\pi_\ell, \pi_u) \ni \pi$,
- ΔL_π denotes the liquidity added on the unitary range $[\underline{\pi}, \overline{\pi})$: note that $\pi \mapsto \Delta L_\pi$ is a piecewise-constant[5] function.

An example of liquidity curve $(L_\pi)_\pi$ is represented in Figure 3.

Theorem 3.3. *Assume $\boldsymbol{H}_0$. Consider a liquidity provider adding a liquidity curve $(\Delta L_\pi)_\pi$ to the pool while keeping x_0 tokens X and y_0 tokens Y outside the pool. At time $t_0 = 0$, when the price in the pool is p_0, their tokens admit the following Y-value:*

$$V_P(t=0) = x_0 \cdot p_0 + y_0 + p_0 \cdot \int_{\pi_0}^{+\infty} \frac{\Delta L_\pi}{\pi^2} \mathrm{d}\pi + \int_0^{\pi_0} \Delta L_\pi \mathrm{d}\pi.$$

At time $t_1 > t_0$, when the price in the pool is p_1, their Y-value net of swap fees is

$$V_P(t=t_1) = x_0 \cdot p_1 + y_0 + p_1 \cdot \int_{\pi_1}^{+\infty} \frac{\Delta L_\pi}{\pi^2} \mathrm{d}\pi + \int_0^{\pi_1} \Delta L_\pi \mathrm{d}\pi. \tag{15}$$

Proof. We denote by $(R_i)_{i\geq 0}$, where $R_i = [\pi_{\ell_i}, \pi_{u_i})$, the sequence of unitary square root price ranges in the pool. The exact Y-value of the tokens at any time t, at which the square root price is π_t, is

[5] It is also rcll (right-continuous with left limit), which is suitable for an integration wrt π.

given by Equation (12):

$$V_P(t) = x_0 \cdot p_t + y_0 + \sum_{i\geq 0} \Delta L_i \cdot \left(p_t \cdot \left(\frac{1}{\pi_t^{R_i}} - \frac{1}{\pi_{u_i}} \right) + \left(\pi_t^{R_i} - \pi_{\ell_i} \right) \right). \tag{16}$$

Consider the first term in the generalized summation, which is is defined by

$$S_1^\star = p_t \cdot \sum_{i\geq 0} \Delta L_i \cdot \left(\frac{1}{\pi_t^{R_i}} - \frac{1}{\pi_{u_i}} \right).$$

Note that for ranges R_i such that $\pi_{u_i} \leq \pi_t$, we have $\pi_t^{R_i} = \pi_{u_i}$ and the corresponding terms in the sum are all equal to 0. For ranges R_i such that $\pi_{\ell_i} \geq \pi_t$, we have $\frac{1}{\pi_t^{R_i}} - \frac{1}{\pi_{u_i}} = \frac{1}{\pi_{\ell_i}} - \frac{1}{\pi_{u_i}}$, and when $\pi_t \in R_i$, we have $\frac{1}{\pi_t^{R_i}} - \frac{1}{\pi_{u_i}} = \frac{1}{\pi_t} - \frac{1}{\pi_{u_i}}$. Writing the sum as an integral and using the convention on ΔL_π, we get

$$S_1^\star = -p_t \cdot \int_{\pi_t}^{+\infty} \Delta L_\pi \mathrm{d}\left(\frac{1}{\pi}\right) = p_t \cdot \int_{\pi_t}^{+\infty} \frac{\Delta L_\pi}{\pi^2} \mathrm{d}\pi. \tag{17}$$

The second term in the summation is handled similarly:

$$S_2^\star \overset{\text{def}}{=} \sum_{i\geq 0} \Delta L_i \cdot \left(\pi_t^{R_i} - \pi_{l_i} \right) = \int_0^{\pi_t} \Delta L_\pi \mathrm{d}\pi. \tag{18}$$

Plugging Equations (17) and (18) into (16) yields the stated result, both when $t = t_0$ and when $t = t_1$. □

The following result gives the first (Delta) and second (Gamma) order sensitivity of the global position with respect to the $X - Y$ exchange rate.

Corollary 3.4. *The Delta and Gamma of the Y-value of the position at $t = 0$ as defined in Theorem 3.3 (net of swap fees) is given by*

$$\Delta_P(t=0) = \frac{\partial V_P(t=0)}{\partial p_0} = x_0 + \int_{\pi_0}^{+\infty} \frac{\Delta L_\pi}{\pi^2} \mathrm{d}\pi,$$
$$\Gamma_P(t=0) = \frac{\partial^2 V_P(t=0)}{\partial p_0^2} = -\frac{\Delta L_{\pi_0}}{2\pi_0^3}.$$

In particular, such a position is always concave in the spot rate (negative Gamma).

These formulas follow from a direct differentiation of (15), details are left to the reader.

3.4 Synthetizing a concave option with an ad hoc liquidity curve

We have seen that depositing a liquidity curve implies a payoff (net of fees) which is concave as a function of the price. We now show the converse, i.e. that any smooth concave payoff can be generated by an LP strategy for some liquidity curve. This is based on the Carr-Madan result that calls and puts form a generating system of any convex payoffs (see Appendix A.4): the connection with Uniswap v3 is then possible since the Impermanent Loss has the same shape as the opposite of the payoff for a put (for ranges below p_0) or a call (for ranges above p_0).

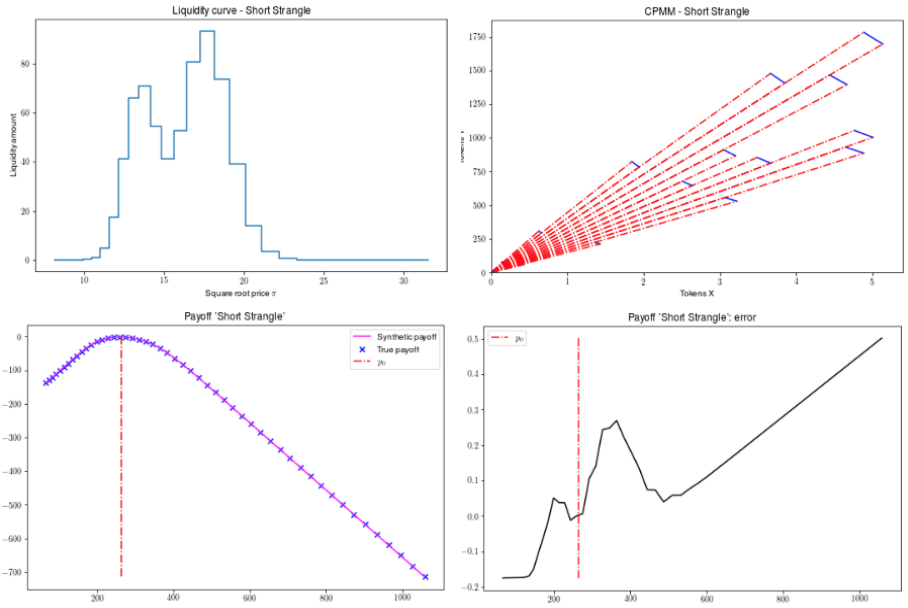

Figure 5: Illustration of Theorem 3.5 when h corresponds to a short strangle (minus a put with strike $K = p_0/1.3$ and minus a call with strike $K = 1.3 \cdot p_0$. Both options are considered with a maturity $\tau = 0.1$ and a Black-Scholes volatility equal to 50%). Top left: the liquidity curve ΔL_R from Equation (19). Top right: the CPMM representation. Bottom left: the payoff and its replication using Theorem 3.5. Bottom right: the reconstruction error. In view of the ranges of y-axis, this error is small.

Theorem 3.5. *Assume $\boldsymbol{H}_0$. Consider a concave payoff function $h : \mathbb{R}^+ \mapsto \mathbb{R}$ which we assume to be C^3 and linear for small and large values. Then, consider a strategy depositing the liquidity curve*

$$\Delta L_R := \left(-h''(\pi_\ell \cdot \pi_u)\right) \cdot (\pi_u + \pi_\ell) \cdot \pi_\ell \cdot \pi_u \tag{19}$$

at time 0 on each range $R = [\pi_\ell, \pi_u)$. In addition, add to the position quantities x_0 of tokens X and y_0 of tokens Y outside the pool, with

$$\begin{aligned} x_0 &= h'(p_0) - \sum_{R=[\pi_\ell,\pi_u)} \Delta L_R \cdot \left(\frac{1}{\pi_0^R} - \frac{1}{\pi_u}\right), \\ y_0 &= h(p_0) - h'(p_0) \cdot p_0 + \sum_{R=[\pi_\ell,\pi_u)} \Delta L_R \cdot \left(\pi_0^R - \pi_\ell\right). \end{aligned}$$

Then the total value $V_P(T)$ of the tokens in the pool (net of swap fees) and of the tokens outside the pool is such that, for any $T > 0$,

$$\left|h(p_T) - (V_P(T) + x_0 \cdot p_T + y_0)\right| \leq C \cdot \delta_\pi \cdot (\beta_{\mathrm{p}} - 1), \quad a.s.,$$

for a constant C that depends only on the payoff function h and its derivatives.

Figure 5 provides an illustration of the above theorem. The following points are important to note:

- First, the concavity assumption ensures that the liquidity to deposit in Equation (19) is indeed non-negative.
- Second, the assumption that $p \mapsto h(p)$ is linear for large and small values of p (hence $h''(p) = 0$ for these values) implies that the number of ranges affected by the liquidity deposit is finite. This is a purely technical condition to simplify the statement.
- Third, as in the case of the Carr-Madan formula, smoothness can be partly relaxed but concavity plays a crucial role. In other words, the assumptions on the smoothness of h and its asymptotic behavior could be relaxed, but they were made here for the sake of having a simplified proof avoiding technicalities that could obfuscate the main messages.
- As far as a practical implementation is concerned, it may happen that this strategy leads to considering negative quantities of tokens X or Y: this is possible by using lending/borrowing protocols, up to paying some extra fees (that we neglect here). The above result states that the value of the payoff $h(p_T)$ is equal, up to a small range width $\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)$, to the value of pool (with the above liquidity curve), minus the Y-value of swap fees in the pool (see Theorem 4.1 for an estimation), plus $x_0 \cdot p_0 + y_0$ for the value of tokens outside the pool.

Proof of Theorem 3.5. We start from the Carr-Madan formula (Appendix A.4) which gives

$$h(p_T) = h(p_0) + h'(p_0) \cdot (p_T - p_0) + \int_{p_0}^{+\infty} h''(K)(p_T - K)_+ \mathrm{d}K + \int_0^{p_0} h''(K)(K - p_T)_+ \mathrm{d}K.$$

The assumption on h ensures that the integral is restricted to a compact set of $(0,\infty)$, which we denote $[\varepsilon, 1/\varepsilon]$ for some $\varepsilon > 0$. We can then replace the integral by a sum on a finite number of ranges $R = [\pi_\ell, \pi_u)$ and evaluate the function at the geometric mean $\pi_\ell \cdot \pi_u$. Note that the width of a price range satisfies $\frac{p_u - p_\ell}{p_\ell} = O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big)$. Therefore, it is easy to verify that

$$\begin{aligned} h(p_T) = h(p_0) + h'(p_0) \cdot (p_T - p_0) &+ \sum_{R=[\pi_\ell,\pi_u) \subset [\pi_0,\infty)} h''(\pi_\ell \cdot \pi_u) \cdot \big(\pi_T^2 - \pi_\ell \cdot \pi_u\big)_+ \cdot \big(\pi_u^2 - \pi_\ell^2\big) \\ &+ \sum_{R=[\pi_\ell,\pi_u) \subset [0,\pi_0)} h''(\pi_\ell \cdot \pi_u) \cdot \big(\pi_\ell \cdot \pi_u - \pi_T^2\big)_+ \cdot \big(\pi_u^2 - \pi_\ell^2\big) \\ &+ O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big) \end{aligned}$$

where the $O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big)$ is uniform in p_T, p_0 and T. Note that in the sum above, the range containing π_0 (in the case where π_0 is not the boundary of such a range) was discarded. This does not significantly modify the magnitude of the error term and it is a slight simplification for the subsequent analysis that could be adjusted in case it is necessary to account for this contribution too.
Since the Impermanent Loss $V_P - V_H$ has the same profile as a call for $p_T > p_0$ and a put for $p_T < p_0$ (Theorem 3.2), we get the approximation:

$$h(p_T) = h(p_0) + h'(p_0) \cdot (p_T - p_0) + \sum_{R=[\pi_\ell,\pi_u)} h''(\pi_\ell \cdot \pi_u) \cdot \frac{(\pi_u^2 - \pi_\ell^2)}{\left(\frac{1}{\pi_\ell} - \frac{1}{\pi_u}\right)} \cdot \left| \frac{1}{\pi_T^R} - \frac{1}{\pi_0^R} \right| \cdot \left| \pi_T^2 - \pi_T^R \cdot \pi_0^R \right|$$

$$
\begin{aligned}
&\quad + O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big) \\
&= h(p_0) + h'(p_0) \cdot (p_T - p_0) + \sum_{R=[\pi_\ell, \pi_u)} \big(-h''(\pi_\ell \cdot \pi_u)\big) \cdot (\pi_u + \pi_\ell) \cdot \pi_\ell \cdot \pi_u \cdot \big(V_P^R(\pi_T) - V_H^R(\pi_T)\big) \\
&\quad + O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big),
\end{aligned}
$$

where $V_P^R(\pi_T) - V_H^R(\pi_T)$ (given in (14)) denotes the Impermanent Loss incurred by adding an amount $\Delta L = 1$ of liquidity to range R at time $t = 0$. Assuming that h is concave, the entire sum reads as the global Loss $V_P - V_H$ with a liquidity curve $\Delta L_R := \big(-h''(\pi_\ell \cdot \pi_u)\big) \cdot (\pi_u + \pi_\ell) \cdot \pi_\ell \cdot \pi_u$, which is Equation (19).

Putting aside the term $\sum_{R=[\pi_\ell, \pi_u)} \Delta L_R \cdot V_P^R(\pi_T)$, and considering the value V_H from Equation (13), we can re-interpret the remaining term $h(p_0) + h'(p_0) \cdot (p_T - p_0) - \sum_{R=[\pi_\ell, \pi_u)} \Delta L_R \cdot V_H^R(\pi_T)$ using the quantities x_0 and y_0 of tokens X and Y outside the pool: this corresponds to

$$
\begin{aligned}
x_0 &= h'(p_0) - \sum_{R=[\pi_\ell, \pi_u)} \Delta L_R \cdot \left(\frac{1}{\pi_0^R} - \frac{1}{\pi_u} \right), \\
y_0 &= h(p_0) - h'(p_0) \cdot p_0 + \sum_{R=[\pi_\ell, \pi_u)} \Delta L_R \cdot \big(\pi_0^R - \pi_\ell\big),
\end{aligned}
$$

which completes the proof.

As a side remark, we observe that the summations in the expressions of the quantities x_0 and y_0 can be simplified using integrals:

$$
\begin{aligned}
x_0 &= h'(p_0) + \int_0^{p_0} h''(p) \cdot 2\sqrt{p} \cdot p \cdot \mathrm{d}\left(-\frac{1}{\sqrt{p}} \right) + O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big) \\
&= h'(p_0) + \int_0^{p_0} h''(p) \mathrm{d}p = 2 \cdot h'(p_0) - h'(0) + O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big), \\
y_0 &= h(p_0) - h'(p_0) \cdot p_0 + \int_{p_0}^{+\infty} (-h''(p)) \cdot 2\sqrt{p} \cdot p \cdot \mathrm{d}\big(\sqrt{p}\big) + O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big) \\
&= h(p_0) - h'(p_0) \cdot p_0 + \int_{p_0}^{+\infty} (-h''(p)) \cdot p \cdot \mathrm{d}p + O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big) \\
&= h(p_0) - h'(p_0) \cdot p_0 + \Big[(-h'(p)) \cdot p \Big]_{p=p_0}^{p=1/\varepsilon} - \int_{p_0}^{+1/\varepsilon} (-h'(p)) \cdot \mathrm{d}p + O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big) \\
&= -h'(1/\varepsilon) \cdot 1/\varepsilon + h(1/\varepsilon) + O\big(\delta_\pi \cdot (\beta_{\mathrm{p}} - 1)\big).
\end{aligned}
$$

□

Example 3.6. *Consider the log payoff $h(p) = \log(p/p_0)$. This payoff can be approximated by depositing the following liquidity profile into the pool*

$$
\Delta L_R = \frac{\pi_u + \pi_\ell}{\pi_\ell \cdot \pi_u} \text{ for all unitary ranges } R = [\pi_\ell, \pi_u).
$$

This payoff plays an important role for designing volatility index [Carr and Wu, 2006]: we leave these investigations for further research. Note however that the payoff h does not satisfy the regularity and asymptotic properties required by Theorem 3.5 and it may cause difficulties in applying arguments in the previous proof. A safe alternative is to linearize h for small and large values of p.

4 Estimation of swap fees

4.1 A general approximation formula

Our final goal is to analyze the distributions of swap fees, for an arbitrary liquidity curve $(\Delta L_\pi)_\pi$. Intuitively, the more the rate $X-Y$ oscillates, the more swap fees are collected: we represent them by the local time of the pool price process p, in a general Itô dynamics. For a proper probabilistic analysis, the price process is defined on a filtered probability space $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{0\le t\le T}, \mathbb{P})$ that supports a Brownian motion W, and with a filtration that satisfies the "usual conditions".
We recall ([Revuz and Yor, 1999, Corollary 1.9 p.227]) that the local time of the price process p (as in Equation (20) below) at level a is a measure of how much time p spends around a over some period of time :

$$A_T^a(p) = \lim_{\varepsilon\downarrow 0} \frac{1}{\varepsilon}\int_0^T \mathbb{1}_{p_t\in[a,a+\varepsilon)} \mathrm{d}\langle p\rangle_t = \lim_{\varepsilon\downarrow 0} \frac{1}{\varepsilon}\int_0^T \mathbb{1}_{p_t\in[a,a+\varepsilon)} a^2\sigma_t^2 \mathrm{d}t.$$

To avoid any confusion with liquidity, the local time is denoted by A (and not by L as it is usually done). In the subsequent analysis, we assume that there exists a risk-neutral measure $\mathbb{P}^\star$ used for the pricing, under which the price process p is a martingale (here the interest rate is set to zero).

Theorem 4.1. *Consider a liquidity provider depositing a liquidity curve $(\Delta L_\pi)_\pi$ at time 0, and assume $(\Delta L_\pi)_\pi$ has a finite support (it is equal to 0 outside a bounded set of square root prices). We analyze the amount of fees in X and Y that were accumulated over the period $[0,T]$ for a given $T>0$. We assume that the swap trades cause the price process $(p_t)_t$ to move from one tick to another.*[6] *We also assume that pool price process $(p_t)_t$ follows an Itô dynamics of the form*

$$\frac{\mathrm{d}p_t}{p_t} = \mu_t \mathrm{d}t + \sigma_t \mathrm{d}W_t, \tag{20}$$

with a stochastic drift $(\mu_t)_t$ and a stochastic volatility $(\sigma_t)_t$: both μ and σ are bounded and uniformly Hölder continuous in time, the volatility σ is positive and bounded away from 0.
Then the amount of fees in X and Y accumulated over the period $[0,T]$ is defined as

$$\mathrm{Fees}^X_{0\to T} = \sum_{R=[\pi_\ell,\pi_u)} \Delta L_\pi \cdot \left(\Phi^X_R(T) - \Phi^X_R(0)\right), \qquad \mathrm{Fees}^Y_{0\to T} = \sum_{R=[\pi_\ell,\pi_u)} \Delta L_\pi \cdot \left(\Phi^Y_R(T) - \Phi^Y_R(0)\right),$$

and approximated as follows:

$$\lim_{\beta_\mathrm{p}\downarrow 1} (\beta_\mathrm{p}-1)\cdot \mathrm{Fees}^X_{0\to T} \overset{\mathbb{P}}{=} \frac{\phi}{1-\phi}\cdot\int_0^{+\infty} \Delta L_{b^{\frac{1}{2}}} \frac{A^b_T(p)}{4\cdot b^{5/2}}\mathrm{d}b, \tag{21}$$

$$\lim_{\beta_\mathrm{p}\downarrow 1} (\beta_\mathrm{p}-1)\cdot \mathrm{Fees}^Y_{0\to T} \overset{\mathbb{P}}{=} \frac{\phi}{1-\phi}\cdot\int_0^{+\infty} \Delta L_{b^{\frac{1}{2}}} \frac{A^b_T(p)}{4\cdot b^{3/2}}\mathrm{d}b, \tag{22}$$

where the limits hold in probability.

[6] In other words, we assume that large swaps can be split into smaller ones for which the price moves from one tick to the consecutive one, and that smaller swaps that do not cause the price to move to another tick can be aggregated until a large enough swap is obtained.

Consider the function

$$\mathfrak{F}\big((\Delta L_\pi)_\pi,(\sigma_t)_t\big)\overset{\text{def}}{=}\frac{1}{(\beta_{\mathrm{p}}-1)}\mathbb{E}^\star\left[\lim_{\beta_{\mathrm{p}}\downarrow 1}(\beta_{\mathrm{p}}-1)\cdot\mathrm{Fees}^X_{0\to T}\cdot p_T+\lim_{\beta_{\mathrm{p}}\downarrow 1}(\beta_{\mathrm{p}}-1)\cdot\mathrm{Fees}^Y_{0\to T}\right],$$

which represents the Y-value of the approximated collected swap fees as a function of the liquidity distribution and the local time of the exchange rate process, assuming a discounting factor equal to 1. We have

$$\mathfrak{F}\big((\Delta L_\pi)_\pi,(\sigma_t)_t\big)=\frac{\phi}{(1-\phi)\cdot(\beta_{\mathrm{p}}-1)}\cdot\int_0^{+\infty}\Delta L_{b^{\frac{1}{2}}}\frac{\mathbb{E}^\star\left[A^b_T(p)\right]}{2\cdot b^{3/2}}\mathrm{d}b. \tag{23}$$

Note that since $(\Delta L_\pi)_\pi$ has a finite support, there are no integrability issues in the above integral representations.
Note also that we refer to approximated collected swap fees for two reasons: first, because we consider the limits in (21)-(22); second because to effectively recover the fees, they have to be withdrawn from the pool, which induces a valuation price that may be different from the pool price, and possibly cost additional transaction fees.
The assumption of an Ito process in Equation (20) is quite standard in the litterature: see for instance [Cartea et al., 2022, Cartea et al., 2023a, Cohen et al., 2023] among others. We shall consider the assumption on the price process p and the one on the swap trades combined; this combination is quite close in the spirit to [Robert and Rosenbaum, 2011]. However this global assumption is questionable, since p has an infinite variation and this could lead to infinite amount of fees (without the condition on swap trades). On the other hand, this setting allows for getting a proxy for fees, and allows for flexibility in the pool price model (the volatility can be stochastic). An alternative modelling could be a jump process with tiny jumps to account for small changes in the pool, but the fees proxy analysis in this model is out of reach so far.

Proof. By Equation (10), the amount of fees in tokens X and Y accumulated over the period per unit of liquidity on the range $R=[\pi_\ell,\pi_u)$ is

$$\Delta\Phi^X_R=\Phi^X_R(T)-\Phi^X_R(0)=\sum_i\frac{a^X_i}{1-\phi}\cdot\frac{\phi}{L_i},\qquad\Delta\Phi^Y_R=\Phi^Y_R(T)-\Phi^Y_R(0)=\sum_j\frac{a^Y_j}{1-\phi}\cdot\frac{\phi}{L_j},$$

where L_i is the liquidity available on the considered range at a time $0\le t_i\le T$ at which a^X_i (resp. a^Y_i) tokens were deposited into the pool on range R to retrieve tokens Y (resp. X). We have assumed that the above amounts of tokens a^X_i and a^Y_i exactly match the quantities that shift the price from one tick to the next one. Thus, using Equations (8) and (9), the sums above can be rewritten as

$$\begin{aligned}\Delta\Phi^X_R&=\sum_{\text{swap } X \text{ for } Y \text{ at time } t_i\in[0,T]}\mathbb{1}_{\pi_{t_i}\in R}\cdot\left(\frac{1}{\pi_{t_i}\cdot\beta_{\mathrm{p}}^{-1/2}}-\frac{1}{\pi_{t_i}}\right)\cdot\frac{\phi}{1-\phi},\\ \Delta\Phi^Y_R&=\sum_{\text{swap } Y \text{ for } X \text{ at time } t_j\in[0,T]}\mathbb{1}_{\pi_{t_j}\in R}\cdot\left(\pi_{t_j}\cdot\beta_{\mathrm{p}}^{1/2}-\pi_{t_j}\right)\cdot\frac{\phi}{1-\phi}.\end{aligned}$$

Here the times t_i are the successive hitting times of the tick grid by the price process p, as it decreases when tokens X are swapped in for tokens Y, or increases in the other case. Using a result from [Gobet and Landon, 2014, Proof of Theorem 2.3] in the context of the general Itô model (20), we know that the total number of hitting times is of order $(\beta_{\mathrm{p}}-1)^{-2}$ in probability and that the timestep $\sup_i |t_{i+1}-t_i|$ is of order $(\beta_{\mathrm{p}}-1)^{-(2-\eta)}$ in probability (for any $\eta>0$). In addition, standard stochastic calculus algebra shows that

$$\mathbb{P}\left(\frac{p_{t_{i+1}}}{p_{t_i}}=\beta_{\mathrm{p}}^{\pm 1}\middle|\mathscr{F}_{t_i}\right)=\frac{1}{2}+O(\beta_{\mathrm{p}}-1) \tag{24}$$

with a remainder uniform in ω; in other words, local price changes are almost symmetric and the bias can be neglected. Combining Equation (24) with a standard convergence result of triangular arrays of random variables (see [Genon-Catalot and Jacod, 1993, Lemma 9]) allows to get

$$\begin{aligned}
\Delta\Phi_R^X &= \frac{(\beta_{\mathrm{p}}^{1/2}-1)\cdot\phi}{(1-\phi)}\cdot\sum_{\text{swap at time } t_i\in[0,T]}\mathbb{1}_{\text{decreasing price between } t_i \text{ and } t_{i+1}}\cdot\mathbb{1}_{\pi_{t_i}\in R}\cdot\frac{1}{\pi_{t_i}}\\
&= \frac{(\beta_{\mathrm{p}}^{1/2}-1)\cdot\phi}{2\cdot(1-\phi)}\cdot\sum_{\text{swap at time } t_i\in[0,T]}\mathbb{1}_{\pi_{t_i}\in R}\cdot\frac{1}{\pi_{t_i}}+O_{\mathbb{P}}(1),\\
\Delta\Phi_R^Y &= \frac{(\beta_{\mathrm{p}}^{1/2}-1)\cdot\phi}{2\cdot(1-\phi)}\cdot\sum_{\text{swap at time } t_i\in[0,T]}\mathbb{1}_{\pi_{t_i}\in R}\cdot\pi_{t_i}+O_{\mathbb{P}}(1).
\end{aligned}$$

Write $(\log(\beta_{\mathrm{p}}))^2=(\log(p_{t_{i+1}}/p_{t_i}))^2=\int_{t_i}^{t_{i+1}}\sigma_t^2\mathrm{d}t+$residual: using martingale convergence results (like [Gobet and Landon, 2014, Proposition 1.5 and Proposition A.1]) we derive

$$\begin{aligned}
\Delta\Phi_R^X &= \frac{(\beta_{\mathrm{p}}^{1/2}-1)\cdot\phi}{2\cdot(1-\phi)}\cdot\sum_{\text{swap at time } t_i\in[0,T]}\mathbb{1}_{\pi_{t_i}\in R}\cdot\frac{1}{\pi_{t_i}}\cdot\frac{\int_{t_i}^{t_{i+1}}\sigma_t^2\mathrm{d}t}{(\log(\beta_{\mathrm{p}}))^2}+o_{\mathbb{P}}((\beta_{\mathrm{p}}-1)^{-1}),\\
\Delta\Phi_R^Y &= \frac{(\beta_{\mathrm{p}}^{1/2}-1)\cdot\phi}{2\cdot(1-\phi)}\cdot\sum_{\text{swap at time } t_i\in[0,T]}\mathbb{1}_{\pi_{t_i}\in R}\cdot\pi_{t_i}\cdot\frac{\int_{t_i}^{t_{i+1}}\sigma_t^2\mathrm{d}t}{(\log(\beta_{\mathrm{p}}))^2}+o_{\mathbb{P}}((\beta_{\mathrm{p}}-1)^{-1}).
\end{aligned}$$

We now replace the summation of terms computed at times t_i by an integral: this can be done using the regularity of the coefficients μ and σ. More specifically, the indicator function at discrete times can also be replaced by its continuous version, taking advantage of the fact that the process π does not spend too much time close to the boundaries of R (the singular points for the indicator function). The computations involve standard routines (like in [Gobet and Menozzi, 2010]), details are left to the reader. Globally, using $\frac{\beta_{\mathrm{p}}^{1/2}-1}{(\log(\beta_{\mathrm{p}}))^2}\approx\frac{1}{2\cdot(\beta_{\mathrm{p}}-1)}$ as $\beta_{\mathrm{p}}\to 1$, we obtain

$$\begin{aligned}
(\beta_{\mathrm{p}}-1)\cdot\Delta\Phi_R^X &= \frac{\phi}{4\cdot(1-\phi)}\cdot\int_0^T\mathbb{1}_{\pi_t\in R}\frac{1}{\pi_t}\cdot\sigma_t^2\mathrm{d}t+o_{\mathbb{P}}(1),\\
(\beta_{\mathrm{p}}-1)\cdot\Delta\Phi_R^Y &= \frac{\phi}{4\cdot(1-\phi)}\cdot\int_0^T\mathbb{1}_{\pi_t\in R}\pi_t\cdot\sigma_t^2\mathrm{d}t+o_{\mathbb{P}}(1).
\end{aligned} \tag{25}$$

We now use the occupation time formula, see [Revuz and Yor, 1999, Corollary 1.6, p.224]: for any Itô process X and any non-negative measurable function Ψ, we have

$$\int_0^T \Psi(X_t)\mathrm{d}\langle X\rangle_t = \int_{\mathbb{R}} \Psi(a) A_T^a(X)\mathrm{d}a. \tag{26}$$

Since $\frac{\mathrm{d}\pi_t}{\pi_t} = \frac{\sigma_t}{2}\mathrm{d}W_t + \left(\frac{\mu_t}{2} - \frac{\sigma_t^2}{8}\right)\mathrm{d}t$, the bracket of π is $\mathrm{d}\langle\pi\rangle_t = \frac{1}{4}\cdot\pi_t^2\cdot\sigma_t^2\mathrm{d}t$, therefore the occupation formula with $\Psi^X(\pi) \overset{\text{def}}{=} \mathbb{1}_{\pi\in R}\cdot\pi^{-3}$ and $\Psi^Y(\pi) \overset{\text{def}}{=} \mathbb{1}_{\pi\in R}\cdot\pi^{-1}$ yields

$$(\beta_{\mathrm{p}}-1)\cdot\Delta\Phi_R^X = \frac{\phi}{(1-\phi)}\cdot\int_R \frac{A_T^a(\pi)}{a^3}\mathrm{d}a + o_{\mathbb{P}}(1), \qquad (\beta_{\mathrm{p}}-1)\cdot\Delta\Phi_R^Y = \frac{\phi}{(1-\phi)}\cdot\int_R \frac{A_T^a(\pi)}{a}\mathrm{d}a + o_{\mathbb{P}}(1).$$

The actual fees retrieved by the liquidity provider are obtained by multiplying the computed quantities by the liquidity added to each range and summing the results, yielding $\mathrm{Fees}_{0\to T}^X = \sum_R \Delta L_\pi \cdot \Delta\Phi_R^X$ and $\mathrm{Fees}_{0\to T}^Y = \sum_R \Delta L_\pi\cdot\Delta\Phi_R^Y$. We have

$$(\beta_{\mathrm{p}}-1)\cdot\mathrm{Fees}_{0\to T}^X = \frac{\phi}{(1-\phi)}\cdot\int_0^{+\infty}\Delta L_a\frac{A_T^a(\pi)}{a^3}\mathrm{d}a + o_{\mathbb{P}}(1),$$
$$(\beta_{\mathrm{p}}-1)\cdot\mathrm{Fees}_{0\to T}^Y = \frac{\phi}{(1-\phi)}\cdot\int_0^{+\infty}\Delta L_a\frac{A_T^a(\pi)}{a}\mathrm{d}a + o_{\mathbb{P}}(1).$$

Now, using the change of variables for local time [Revuz and Yor, 1999, Exercice 1.23 p.234],

$$2a\cdot A_T^a(\pi) = A_T^{a^2}(p), \qquad \forall a>0,$$

plugging this expression into the above integrals and making use of the new variable $b \overset{\text{def}}{=} a^2$ yields Equations (21) and (22).
We now prove that Equation (23) holds. More specifically, using Equation (25), we compute

$$\mathbb{E}^\star\left[\sum_R \Delta L_\pi\cdot\left(\lim_{\beta_{\mathrm{p}}\downarrow 1}(\beta_{\mathrm{p}}-1)\cdot\Delta\Phi_R^X\cdot p_T + \lim_{\beta_{\mathrm{p}}\downarrow 1}(\beta_{\mathrm{p}}-1)\cdot\Delta\Phi_R^Y\right)\right]$$
$$= \sum_R \Delta L_\pi\cdot\frac{\phi}{4\cdot(1-\phi)}\cdot\mathbb{E}^\star\left[\int_0^T \mathbb{1}_{\pi_t\in R}\frac{p_T}{\pi_t}\cdot\sigma_t^2\mathrm{d}t + \int_0^T \mathbb{1}_{\pi_t\in R}\pi_t\cdot\sigma_t^2\mathrm{d}t\right].$$

Under the risk-neutral valuation rule, we have $\mathbb{E}^\star\left[\frac{p_T}{\pi_t}\middle|\mathscr{F}_t\right] = \pi_t$, meaning that in total value, this expression is equal to twice the Y-fees. Thus, the end of the computation is the same as Equation (22), with an extra factor 2 and with the risk-neutral expectation. □

Corollary 4.2 (Expected fees using CEX call/put prices)**.** *Assume that the price within the pool coincides with the one outside the pool, and that the risk-neutral pricing rule* $\mathrm{Put}_{t=0}(T,b) = \mathbb{E}^\star\left[(b-p_T)_+\right]$ *and* $\mathrm{Call}_{t=0}(T,b) = \mathbb{E}^\star\left[(p_T-b)_+\right]$ *holds for the call/put options traded outside the pool*[7]*. In the setting of Theorem 4.1, we have*

$$\mathbb{E}^\star\left[\lim_{\beta_{\mathrm{p}}\downarrow 1}(\beta_{\mathrm{p}}-1)\cdot\mathrm{Fees}_{0\to T}^X\cdot p_T + \lim_{\beta_{\mathrm{p}}\downarrow 1}(\beta_{\mathrm{p}}-1)\cdot\mathrm{Fees}_{0\to T}^Y\right]$$

[7] Typically on CEXes (Centralized Exchanges)

$$= \frac{\phi}{(1-\phi)} \cdot \left(\int_0^{p_0} \Delta L_{b^{\frac{1}{2}}} \cdot \frac{\mathrm{Put}_{t=0}(T,b)}{b^{3/2}} \mathrm{d}b + \int_{p_0}^{+\infty} \Delta L_{b^{\frac{1}{2}}} \cdot \frac{\mathrm{Call}_{t=0}(T,b)}{b^{3/2}} \mathrm{d}b \right).$$

This formula shows that in a situation where the price within the pool coincides with the one outside the pool, the value of fees (for a general liquidity curve) should be aligned with the values of calls/puts written on the same rate $X-Y$. This formula has a similar shape to the Carr-Madan formula (recalled in Appendix A.4), and it is interesting to notice that, similarly to a volatility index [Carr and Wu, 2006], it can be evaluated from call/put market prices.
We mention that in many pools, the coincidence of prices within and outside the pool does not hold perfectly: see [Jaimungal et al., 2023] for a recent study of the Spot-Pool spread. Hence, in these situations, the validity of Corollary 4.2 is debatable.

Proof of Corollary 4.2. The Tanaka formula [Revuz and Yor, 1999, Theorem 2.1, p. 222] gives

$$(p_T - K)_+ = (p_0 - K)_+ + \int_0^T \mathbb{1}_{p_t > K} \mathrm{d}p_t + \frac{1}{2} A_T^K(p),$$
$$(K - p_T)_+ = (K - p_0)_+ - \int_0^T \mathbb{1}_{p_t \leq K} \mathrm{d}p_t + \frac{1}{2} A_T^K(p).$$

Therefore, for OTM options ($p_0 < K$ for a Call and $p_0 > K$ for a Put), we have

$$\mathrm{Call}_{t=0}(T,K) = \frac{1}{2}\mathbb{E}^\star\left[A_T^K(p)\right], \qquad \mathrm{Put}_{t=0}(T,K) = \frac{1}{2}\mathbb{E}^\star\left[A_T^K(p)\right].$$

Plugging these values into Equation (23) and splitting the integral depending on whether $K > p_0$ or $K < p_0$ yields the stated result. □

4.2 Estimation of expected swap fees in Black-Scholes model

Our goal is to derive an analytical formula of

$$\mathbb{E}^\star\left[\lim_{\beta_{\mathrm{p}}\downarrow 1}(\beta_{\mathrm{p}}-1)\cdot\mathrm{Fees}^X_{0\to T}\cdot p_T + \lim_{\beta_{\mathrm{p}}\downarrow 1}(\beta_{\mathrm{p}}-1)\cdot\mathrm{Fees}^Y_{0\to T}\right]$$

in the case where liquidity is concentrated on the range $R = [\pi_\ell, \pi_u)$ and when the price evolves like a Geometric Brownian motion.

Theorem 4.3. *Assume the conditions and notations of Theorem 4.1 with constant volatility $\sigma_t = \sigma$, and with $\Delta L_\pi = 1$ for $\pi = \pi_\ell$ and 0 otherwise. Then the risk-neutral Y-value of the renormalized approximated collected swap fees (as defined in (23)) is given by*

$$\mathbb{E}^\star\left[\lim_{\beta_{\mathrm{p}}\downarrow 1}(\beta_{\mathrm{p}}-1)\cdot\mathrm{Fees}^X_{0\to T}\cdot p_T + \lim_{\beta_{\mathrm{p}}\downarrow 1}(\beta_{\mathrm{p}}-1)\cdot\mathrm{Fees}^Y_{0\to T}\right]$$
$$= \frac{\phi\cdot\sigma^2}{2\cdot(1-\phi)} \cdot \int_0^T \pi_0 e^{-\frac{\sigma^2}{8}t} \cdot \left(\mathcal{N}\left(\frac{1}{\sigma\sqrt{t}}\cdot\ln\left(\frac{p_0}{p_l}\right) - \frac{1}{2}\cdot\sigma\sqrt{t}\right) - \mathcal{N}\left(\frac{1}{\sigma\sqrt{t}}\cdot\ln\left(\frac{p_0}{p_u}\right) - \frac{1}{2}\cdot\sigma\sqrt{t}\right)\right)\mathrm{d}t. \quad (27)$$

The proof is postponed to Subsection A.3.

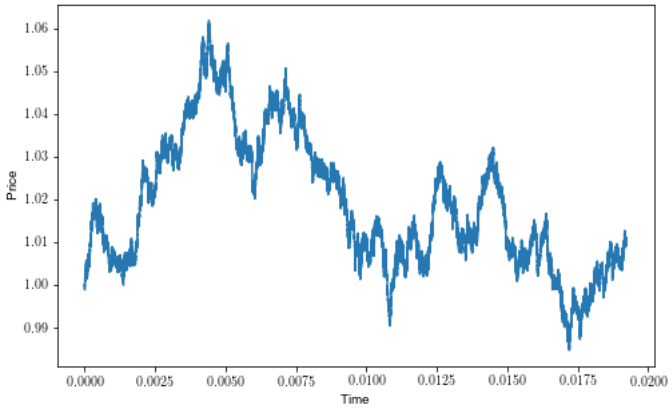

Figure 6: Sample path of p, with $p_0 = 1$, $\sigma = 40\%$ and drift $\mu = 5\%$

4.3 Numerical experiments

This subsection is devoted to the illustration of previous results about fees. The experimentation is performed on synthetic data, using a Geometric Brownian motion for p with constant volatility $\sigma = 40\%$ and drift $\mu = 5\%$, over the period $[0, T]$ with $T = 1/52$ (1 week). Given a path of p, we compute the fees depending on different assumptions on the tick spacing δ_π. We recall that tick spacings and swap fees are related: $\delta_\pi = 10, 60, 200$ corresponds to $\phi = 0.05\%, 0.3\%, 1\%$. For our test we add the values $\delta_\pi = 2, \phi = 0.01\%$.
The path sampling of p has required to draw the hitting times by p of the ticks grid with $\beta_p = 1.0001$. Fortunately, the distributions of hitting times and positions are explicit (see [Borodin and Salminen, 2002, Section II.2]) and thus can be sampled on a computer. Figure 6 represents the sample path that was obtained for this test.

4.3.1 Illustrations of Theorem 4.1

In Figure 7 we report the exact fees collected in tokens X and Y, depending on different price ranges, and report on a second axis the occupation density of p. We observe that fees and occupation density are strongly linked, which is coherent with (25) and Theorem 4.1.
In Figure 8, we verify the validity of the approximation in Equation (25). For this purpose, we depict the scatter plots of the exact fees versus their limits, for various values of δ_π. We observe that points are located on the diagonal showing an excellent accuracy of the formulas.

4.3.2 Illustrations of Theorem 4.3

Thanks to the analytical formula (27), it is straightforward to compute the expected fees across different ranges. We proceed by direct numerical integration of the time integral in (27). The fees values obviously increase as the time horizon gets larger; the fees values also increase as the volatility gets larger, which is coherent with our assumption which considers that swap trades occur as

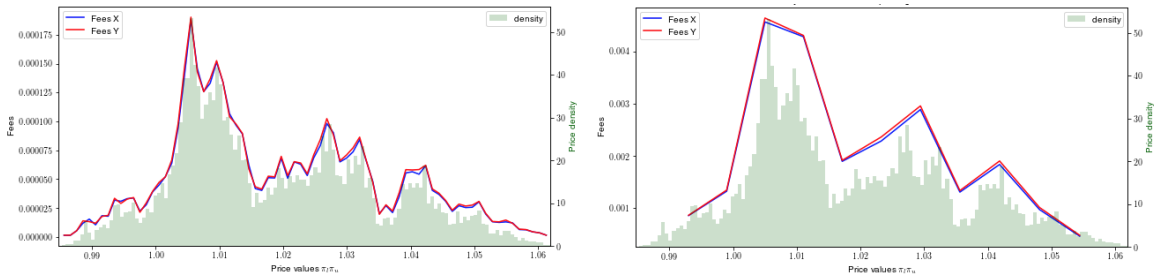

Figure 7: Left axis: the exact fees collected in tokens X and Y. Right axis: occupation density of p. Left: when $\delta_\pi = 10$. Right: when $\delta_\pi = 60$.

frequently as prices change from one tick to another.

5 Conclusion

In this work, we have revisited in detail the core mechanisms of Uniswap v3, anchoring our analysis on the source code of the protocol. We exhibit some new simplified formulas of the Impermanent Loss for general liquidity curves. This gives the opportunity to replicate any concave payoff (under mild regularity and growth assumptions). The analysis is undertaken in the full generality on the occurrence of swap trades and burn/mint liquidity events. We establish a new convergence result for the renormalized collected fees in each token, in the limit of a small tick spacing. This highlights the importance of the time spent by the process in each range, which translates into an integral with respect to the local time of the price process. This enables us to easily compute the expected fees as an integral of call/puts, with a quite general exchange rate process.
This work opens interesting perspectives: 1) comparing the fees predicted by our formula with those actually observed in the pool; 2) deriving an analytical formula for fees in a jump model; 3) performing a quantitative study of arbitrage opportunities between Calls/Puts and Uniswap v3 pools. This would allow to complete the analyses done in [Capponi and Jia, 2021] who argue that arbitrageurs extract profits from liquidity providers. All these topics are left for future research.

A Technical results

A.1 Proof of Proposition 2.3

We prove the result when the initial price $\pi_0 \in R$, the cases where $\pi_0 < \pi_\ell$ or $\pi_0 > \pi_u$ are proved in a similar way. Note that by construction we have $\pi_\ell = \pi_{\ell_1}$ and $\pi_u = \pi_{u_n}$. For $i = 1, \dots, n$, the amounts of tokens deposited on (or withdrawn from) range R_i are given by Equation (5) (see also Remark 2.2):

$$\Delta x_{r_i} = \Delta L \cdot \left(\frac{1}{\pi_0^{R_i}} - \frac{1}{\pi_{u_i}} \right) \text{ and } \Delta y_{r_i} = \Delta L \cdot \left(\pi_0^{R_i} - \pi_{\ell_i} \right).$$

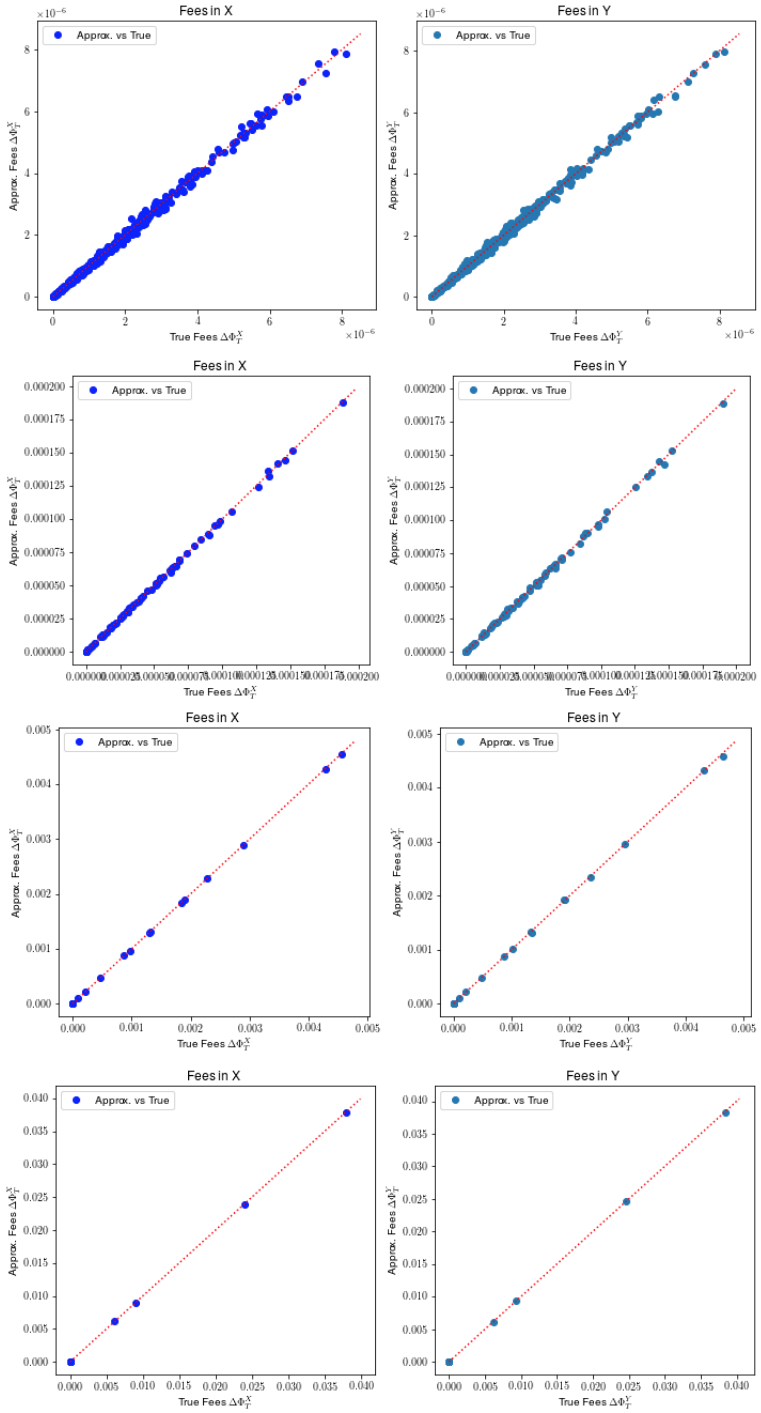

Figure 8: True collected fees versus their approximations. From top to bottom: $\delta_\pi = 2, 10, 60, 200$. Each point corresponds to the amount of swap fee on a range; smaller values of δ_π have more ranges hence more points.

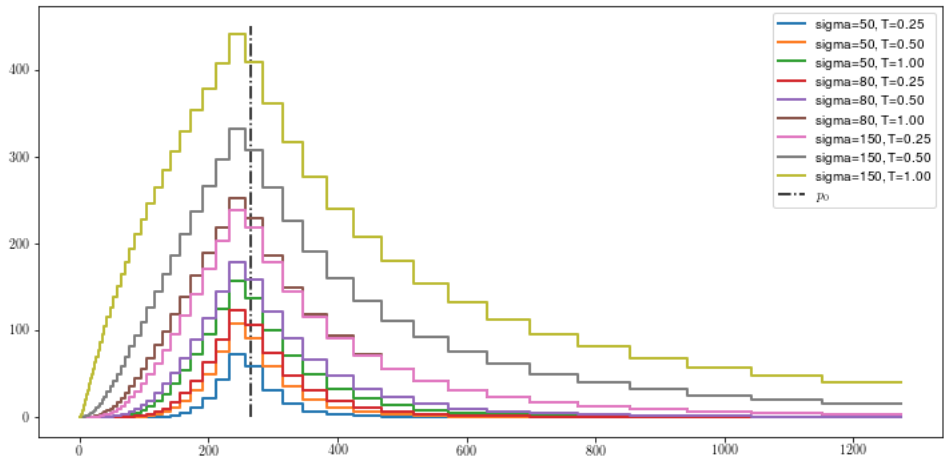

Figure 9: In the Black-Scholes model, representation of the expected fees (Equation (27)) as a function of square root price ranges, for different volatilities (σ =50%, 80%, 150%) and different time horizons ($T = 0.25, 0.5, 1$).

Let j denote the index such that $\pi_0 \in R_j$. By definition if $i < j$ then $\pi_0^{R_i} = \pi_{u_i}$, and if $i > j$ then $\pi_0^{R_i} = \pi_{\ell_i}$. The total amount of tokens X deposited by the liquidity provider is

$$\begin{aligned}
\Delta x_r &= \sum_{i=1}^{n} \Delta L \cdot \left(\frac{1}{\pi_0^{R_i}} - \frac{1}{\pi_{u_i}} \right) \\
&= \sum_{i<j} \Delta L \cdot \left(\frac{1}{\pi_0^{R_i}} - \frac{1}{\pi_{u_i}} \right) + \Delta L \cdot \left(\frac{1}{\pi_0} - \frac{1}{\pi_{u_j}} \right) + \sum_{j<i\leq n} \Delta L \cdot \left(\frac{1}{\pi_0^{R_i}} - \frac{1}{\pi_{u_i}} \right) \\
&= \Delta L \cdot \left(\frac{1}{\pi_0} - \frac{1}{\pi_{u_j}} \right) + \sum_{j<i\leq n} \Delta L \cdot \left(\frac{1}{\pi_{u_{i-1}}} - \frac{1}{\pi_{u_i}} \right) = \Delta L \cdot \left(\frac{1}{\pi_0} - \frac{1}{\pi_u} \right) = \Delta L \cdot \left(\frac{1}{\pi_0^R} - \frac{1}{\pi_u} \right).
\end{aligned}$$

In a similar way we can prove that $\Delta y_r = \Delta L \cdot \left(\pi_0^R - \pi_\ell \right)$, hence the result. □

A.2 Proof of Theorem 3.2

The amounts of tokens that are deposited or withdrawn from the pool are given by Equation (7) from Proposition 2.3. At time $t = 0$, the square root price in the pool is π_0 and we have

$$\Delta x_r(t=0) \;=\; \Delta L \cdot \left(\frac{1}{\pi_0^R} - \frac{1}{\pi_u} \right) \quad \text{and} \quad \Delta y_r(t=0) \;=\; \Delta L \cdot (\pi_0^R - \pi_\ell).$$

At time $t = t_1$, when the square root price in the pool is π_1, we have

$$\Delta x_r(t=t_1) \;=\; \Delta L \cdot \left(\frac{1}{\pi_1^R} - \frac{1}{\pi_u} \right) \quad \text{and} \quad \Delta y_r(t=t_1) \;=\; \Delta L \cdot (\pi_1^R - \pi_\ell).$$

Under hypothesis $\mathbf{H}_0$ we have $V_P = \Delta x_r(t=t_1)\cdot p_1 + \Delta y_r(t=t_1)$ and $V_H = \Delta x_r(t=0)\cdot p_1 + \Delta y_r(t=0)$, which yields Equations (12) and (13).
The Impermanent Loss is derived by combining both equations above, we have

$$\begin{aligned}\mathrm{IL} &= V_P - V_H = \Delta L\cdot\left(\left(\frac{1}{\pi_1^R}-\frac{1}{\pi_0^R}\right)\cdot\pi_1^2 + \left(\pi_1^R-\pi_0^R\right)\right)\\ &= -\Delta L\cdot\left(\pi_0^R-\pi_1^R\right)\cdot\left(1-\frac{\pi_1^2}{\pi_0^R\cdot\pi_1^R}\right).\end{aligned}$$

There remains to verify that the Impermanent Loss is always non-positive. It is clearly 0 when $\pi_0^R = \pi_1^R$, we consider the two remaining cases.

When $\pi_0^R > \pi_1^R$. This entails that $\pi_u > \pi_1^R \geq \pi_1$. We deduce that $\pi_0^R - \pi_1^R > 0$ and $\frac{\pi_1^2}{\pi_0^R\cdot\pi_1^R} < 1$, hence the result.

When $\pi_0^R < \pi_1^R$. This entails that $\pi_\ell < \pi_1^R \leq \pi_1$. We deduce that $\pi_0^R - \pi_1^R < 0$ and $\frac{\pi_1^2}{\pi_0^R\cdot\pi_1^R} > 1$, hence the result.

□

A.3 Proof of Theorem 4.3

Starting from Equation (23) and using the occupation time formula (26) with p, we have:

$$\begin{aligned}\mathfrak{F}\left((\Delta L_\pi)_\pi, (\sigma_t)_t\right) &= \frac{\phi}{(1-\phi)\cdot(\beta_{\mathrm{p}}-1)}\cdot\mathbb{E}^\star\left[\int_{p_l}^{p_u}\frac{A_T^b(p)}{2\cdot b^{3/2}}\mathrm{d}b\right]\\ &= \frac{\sigma^2\cdot\phi}{2\cdot(1-\phi)\cdot(\beta_{\mathrm{p}}-1)}\cdot\mathbb{E}^\star\left[\int_0^T \mathbb{1}_{\pi_t\in[\pi_\ell,\pi_u)}\cdot\pi_t\mathrm{d}t\right].\end{aligned}$$

In the Black-Scholes model with $\sigma_t = \sigma$, we have

$$\pi_t = \pi_0 e^{\frac{\sigma}{2}W_t - \frac{\sigma^2}{4}t} = \pi_0 e^{\frac{\sigma}{2}W_t - \frac{1}{2}\left(\frac{\sigma}{2}\right)^2 t - \frac{\sigma^2}{8}t},$$

i.e, π_t can be seen as a risk-neutral Geometric Brownian Motion with interest rate $-\frac{\sigma^2}{8}$ and volatility $\frac{\sigma}{2}$. Therefore, the computations leading to the Black-Scholes formula give

$$\mathbb{E}\left[\mathbb{1}_{\pi_t\geq b}\cdot\pi_t\right] = \pi_0 e^{-\frac{\sigma^2}{8}t}\cdot\mathcal{N}\left(\frac{2}{\sigma\sqrt{t}}\ln\left(\frac{\pi_0\cdot e^{-\frac{\sigma^2}{8}t}}{b}\right) - \frac{1}{4}\sigma\sqrt{t}\right) = \pi_0 e^{-\frac{\sigma^2}{8}t}\cdot\mathcal{N}\left(\frac{1}{\sigma\sqrt{t}}\ln\left(\frac{p_0}{b^2}\right) - \frac{1}{2}\sigma\sqrt{t}\right).$$

Computing the difference between the above values with $b \overset{\mathrm{def}}{=} \pi_\ell$ and $b \overset{\mathrm{def}}{=} \pi_u$ leads to the result. □

A.4 Carr-Madan formula

Theorem A.1 ([Carr and Madan, 2001, Appendix 1])**.** *The system of Call and Put payoffs with maturity* T *:* $\left((p_T-K)_+,(K-p_T)_+\right)_{K\geq 0}$ *allows to statically replicate any vanilla payoff* $h(p_T)$*, where* h *can be any regular function or difference of convex functions: for any* p_T *and any* $p_0 \geq 0$*,*

$$h(p_T) = h(p_0) + h'(p_0)\cdot(p_T - p_0) + \int_{p_0}^{+\infty} h''(K)(p_T-K)_+ \mathrm{d}K + \int_0^{p_0} h''(K)(K-p_T)_+ \mathrm{d}K.$$

References

[Adams et al., 2020] Adams, H., Zinsmeister, N., and Robinson, D. (2020). Uniswap v2 Core. Technical report. https://uniswap.org/whitepaper.pdf.

[Adams et al., 2021] Adams, H., Zinsmeister, N., Salem, M., Keefer, R., and Robinson, D. (2021). Uniswap v3 core. *Uniswap, Tech. Rep.* https://uniswap.org/whitepaper-v3.pdf.

[Angeris and Chitra, 2020] Angeris, G. and Chitra, T. (2020). Improved price oracles: Constant function market makers. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pages 80–91.

[Bichuch and Feinstein, 2023] Bichuch, M. and Feinstein, Z. (2023). Implied volatility in decentralized finance from automated market makers. Presentation at SIAM Conference on Financial Mathematics and Engineering, Philadelphia, 6-9 June 2023.

[Borodin and Salminen, 2002] Borodin, A. N. and Salminen, P. (2002). *Handbook of Brownian Motion-Facts and Formulae*. Birkhäuser Basel.

[Boueri, 2022] Boueri, N. (2022). G3M impermanent loss dynamics. *arXiv preprint arXiv:2108.06593*.

[Capponi et al., 2023] Capponi, A., Iyengar, G., and Sethuraman, J. (2023). Decentralized finance: Protocols, risks, and governance. *Foundations and Trends in Privacy and Security*, 5(3):144–188.

[Capponi and Jia, 2021] Capponi, A. and Jia, R. (2021). The adoption of blockchain-based decentralized exchanges. *arXiv preprint arXiv:2103.08842*.

[Carr and Madan, 2001] Carr, P. and Madan, D. (2001). Towards a theory of volatility trading. *Option Pricing, Interest Rates and Risk Management, Handbook in Mathematical Finance*, pages 458–476.

[Carr and Wu, 2006] Carr, P. and Wu, L. (2006). A tale of two indices. *The Journal of Derivatives*, 13(3):13–29.

[Cartea et al., 2022] Cartea, Á., Drissi, F., and Monga, M. (2022). Decentralised finance and automated market making: Execution and speculation. *Available at SSRN 4144743*.

[Cartea et al., 2023a] Cartea, Á., Drissi, F., and Monga, M. (2023a). Decentralised finance and automated market making: Predictable loss and optimal liquidity provision. *arXiv preprint arXiv:2309.08431*.

[Cartea et al., 2023b] Cartea, Á., Drissi, F., Sánchez-Betancourt, L., Siska, D., and Szpruch, L. (2023b). Automated market makers designs beyond constant functions. *Available at SSRN 4459177*.

[Cohen et al., 2023] Cohen, S., Vidales, M. S., Siska, D., and Szpruch, L. (2023). Inefficiency of CFMs: hedging perspective and agent-based simulations. *arXiv preprint arXiv:2302.04345*.

[Fan et al., 2023] Fan, Z., Marmolejo-Cossio, F., Moroz, D. J., Neuder, M., Rao, R., and Parkes, D. C. (2023). Strategic liquidity provision in Uniswap v3. *arXiv preprint arXiv:2106.12033*.

[Fan et al., 2022] Fan, Z., Marmolejo-Cossiŋo, F., Altschuler, B., Sun, H., Wang, X., and Parkes, D. C. (2022). Differential liquidity provision in Uniswap v3 and implications for contract design. *arXiv preprint arXiv:2204.00464*.

[Genon-Catalot and Jacod, 1993] Genon-Catalot, V. and Jacod, J. (1993). On the estimation of the diffusion coefficient for multidimensional diffusion processes. *Ann. Inst. H. Poincaré (Probab. Statist.)*, 29:119–151.

[Gobet and Landon, 2014] Gobet, E. and Landon, N. (2014). Optimization of joint p-variations of Brownian semimartingales. *Electronic Journal of Probability*, 19(36).

[Gobet and Melachrinos, 2023] Gobet, E. and Melachrinos, A. (2023). Decentralized finance & blockchain technology. *hal preprint 04131680*. Tutorial at SIAM Financial Mathematics and Engineering 2023, Jun 2023, Philadelphia, United States.

[Gobet and Menozzi, 2010] Gobet, E. and Menozzi, S. (2010). Stopped diffusion processes: boundary corrections and overshoot. *Stochastic Processes and Their Applications*, 120:130–162.

[Jaimungal et al., 2023] Jaimungal, S., Saporito, Y. F., Souza, M. O., and Thamsten, Y. (2023). Optimal trading in automatic market makers with deep learning. *arXiv preprint arXiv:2304.02180*.

[Lambert, 2021] Lambert, G. (2021). Uniswap v3 LP tokens as perpetual put and call options. *Medium*. https://medium.com/@lambert-guillaume/uniswap-v3-lp-tokens-as-perpetual-put-and-call-options-5b66219db827.

[Loesch et al., 2021] Loesch, S., Hindman, N., Richardson, M. B., and Welch, N. (2021). Impermanent loss in Uniswap v3. *arXiv preprint arXiv:2111.09192*.

[Milionis et al., 2022] Milionis, J., Moallemi, C. C., Roughgarden, T., and Zhang, A. L. (2022). Automated market making and loss-versus-rebalancing. *arXiv preprint arXiv:2208.06046*.

[Pintail, 2020] Pintail (2020). Uniswap: A good deal for liquidity providers? https://pintail.medium.com/uniswap-a-good-deal-for-liquidity-providers-104c0b6816f2.

[Revuz and Yor, 1999] Revuz, D. and Yor, M. (1999). *Continuous martingales and Brownian motion*. Comprehensive Studies in Mathematics. Berlin: Springer, third edition.

[Robert and Rosenbaum, 2011] Robert, C. Y. and Rosenbaum, M. (2011). A new approach for the dynamics of ultra-high-frequency data: The model with uncertainty zones. *Journal of Financial Econometrics*, 9(2):344–366.

[Tangri et al., 2023] Tangri, R., Yatsyshin, P., Duijnstee, E. A., and Mandic, D. (2023). Generalizing impermanent loss on decentralized exchanges with constant function market makers. *arXiv preprint arXiv:2301.06831*.