Optimal execution on Uniswap v2/v3 under transient price impact

Bastien Baude[*] Damien Challet[†] Ioane Muni Toke[‡]

Université Paris-Saclay, CentraleSupélec, Laboratoire MICS
91192 Gif-sur-Yvette, France

December 2, 2025

Abstract

We study the optimal liquidation of a large position on Uniswap v2 and Uniswap v3 in discrete time. The instantaneous price impact is derived from the AMM pricing rule. Transient impact is modeled to capture either exponential or approximately power-law decay, together with a permanent component. In the Uniswap v2 setting, we obtain closed-form solutions under general price dynamics. For Uniswap v3, we consider a two-layer liquidity framework, which naturally extends to multiple layers. We address the problem using dynamic programming under geometric Brownian motion dynamics and approximate the solution numerically using a discretization scheme.

Keywords – Decentralized finance, decentralized exchange, automated market makers, Uniswap, market microstructure, optimal execution, optimal scheduling problem.

Contents

[*] bastien.baude@centralesupelec.fr [†] damien.challet@centralesupelec.fr [‡] ioane.muni-toke@centralesupelec.fr

1 Introduction

Since the emergence of Decentralized Finance (DeFi), Decentralized Exchanges (DEXs) have rapidly evolved as leading venues for trading crypto-assets on blockchain networks. Whereas Centralized Exchanges (CEXs) rely on Limit Order Books (LOBs) to match buyers and sellers, DEXs facilitate transactions via liquidity pools. Liquidity Providers (LPs) deposit crypto-assets into these pools, while Liquidity Takers (LTs) trade against them. Transaction terms are set by algorithmic pricing rules known as Automated Market Makers (AMMs). Fees are charged on trades and shared between LPs and DEXs.

Uniswap has played a central role in defining the architecture of DEXs. Uniswap v1, launched in 2018, allows users to trade crypto-assets against Ethereum (ETH) and introduced the Constant Product AMM (CPAMM). This mechanism was formalized and generalized in Uniswap v2 (Adams et al., 2020), released in 2020, which supports arbitrary crypto-asset pairs. Uniswap v3, introduced in 2021 (Adams et al., 2021), extends the CPAMM by introducing the concept of Concentrated Liquidity AMM (CLAMM). LPs are no longer required to allocate capital across the entire price range; instead, they can specify custom price intervals in which their liquidity is active. Beyond Uniswap, alternative AMMs have emerged to address specific types of crypto-asset pairs; see Curve (Egorov, 2019) and Balancer (Martinelli and Mushegian, 2019).

The Total Value Locked (TVL) refers to the value of the crypto-assets, generally expressed in dollars, held in liquidity pools. As of September 2025, TVL across all DEXs is $25b,[1] with a daily trading volume of $12.5b. Uniswap accounts for $6b in TVL and processes a daily volume of $4.5b.

The challenge faced by LTs when executing large trades on LOBs has been extensively studied in the literature. The Almgren-Chriss framework (Almgren and Chriss, 1999, 2001) pioneers optimal execution modelling by formulating the trade-off between execution costs and the risk of adverse price movements during execution. The model, written in discrete time, combines linear instantaneous and permanent price impacts with a Bachelier dynamic for the fundamental price; see Almgren (2003) for the continuous-time version with nonlinear instantaneous price impact. Several extensions are proposed in Guéant (2016) and Cartea et al. (2015). While the Almgren-Chriss model accounts for instantaneous and permanent price impacts, it assumes no gradual recovery of the price following a trade. This intermediate regime, known as transient market impact, is first modeled in Obizhaeva and Wang (2013) through an exponential decay kernel. The model has been extended in various directions; see Alfonsi et al. (2010, 2008) and Gatheral et al. (2012); Curato et al. (2017).

Few works have addressed the optimal execution problem on AMMs. Cartea et al. (2025) develop a continuous-time model in the spirit of Almgren-Chriss, applied to CPAMMs. A first-order approximation is used for the instantaneous price impact, while the permanent impact is assumed to be linear. Regarding prices dynamics, three models are introduced: (i) the fundamental price from the CEX follows a geometric Brownian motion, while the AMM spot price mean-reverts around it; (ii) only the AMM spot price is modeled, with stochastic liquidity; (iii) spot prices from CEX and AMM are assumed to be cointegrated. In all three models, piecewise constant instantaneous price impacts are used to approximate the solution, and the results are illustrated using market data. Cartea et al. (2023) extend the model to account for execution across multiple AMMs. However, to the best of our knowledge, the literature has not yet addressed the optimal execution problem on AMMs under transient price impact.

In this paper, we study the optimal execution problem, first on CPAMMs and then CLAMMs, under transient price impact. The spot price is modeled as the combination of three components: (i) a fundamental price process; (ii) the cumulative price impact induced by past trades; and (iii) a transient impact model. The instantaneous price impact is derived from the AMM pricing rule. Following Cartea et al. (2025), we consider a first-order expansion, under which the price impact depends on the square root of the spot price and is therefore stochastic. In the spirit of Obizhaeva and Wang (2013), transient price impact is modeled using a combination of exponential kernels in order to capture either exponential or approximately power-law decay, together with a permanent component. The execution problem is formulated in discrete time and

[1] Data sourced from DefiLlama, available at `https://defillama.com/protocols/dexs`.

consists in maximizing the expected total cash-flows resulting from the liquidation. For CPAMMs, we obtain closed-form solutions under general fundamental price dynamics. For CLAMMs, we formulate the problem in a dynamic programming framework under geometric Brownian motion and propose a numerical scheme to approximate the solution.

In the literature, several works have addressed the problem of optimal order routing across networks of AMMs. Danos et al. (2021) formalize this problem as a convex optimization program and provide closed-form bounds of the solution for networks of CPAMMs. Angeris et al. (2022) extend this framework by allowing multiple trades within the optimization problem. Diamandis et al. (2023) present an algorithm for optimal routing through constant function market makers (CFMMs), using a dual decomposition approach. Danos et al. (2024) focus on networks of CLAMMs and propose a simple and efficient "push-and-solve" algorithm implementable on blockchain networks. Finally, optimal routing over CFMM networks in the presence of hooks is discussed in Chitra et al. (2025).

The remainder of the paper is organized as follows. Section 2 introduces the optimal execution problem on CPAMMs under transient price impact and provides closed-form solutions under general fundamental price dynamics. Anticipating that no closed-form solution is available in the CLAMMs case, the problem is also reformulated and solved within a dynamic programming framework under geometric Brownian motion assumption. Several scenarios are then considered to illustrate the results. Section 3 turns to the CLAMMs case and presents a numerical scheme to approximate the solution. Results are illustrated within a two-layer liquidity framework, which naturally extends to multiple layers.

2 The optimal scheduling problem on Uniswap v2

2.1 Mathematical framework

We consider a liquidity pool governed by a Constant Product Automated Market Maker (CPAMM), following the Uniswap v2 design. The pool consists of two tokens (i.e., crypto-assets), denoted as token a (e.g., ETH) and token b (e.g., USDT), with reserves q^a and q^b, respectively. The constant product formula defining the AMM is:

$$q^a q^b = L^2, \tag{1}$$

where L is the liquidity of the pool. The spot price of token a in terms of token b (e.g., the price of ETH in USDT) is given by:

$$p = \frac{q^b}{q^a}. \tag{2}$$

Liquidity takers execute trades against the pool. Suppose a trader wants to sell $\delta^a > 0$ units of token a. In return, the AMM determines the amount $\delta^b > 0$ of token b to be received by solving:

$$(q^a + \delta^a)(q^b - \delta^b) = L^2, \tag{3}$$

assuming no fees. Fees are discussed in Section 2.4. The trade is settled in such a way that the constant product formula (1) remains satisfied after the swap. Solving (3) for δ^b yields:

$$\delta^b = q^b - \frac{L^2}{q^a + \delta^a}. \tag{4}$$

From (1) and (2), the reserves can be expressed in terms of liquidity and spot price:

$$q^a = \frac{L}{\sqrt{p}}, \quad q^b = L\sqrt{p}. \tag{5}$$

Substituting into (4), the amount δ^b received by the trader becomes:

$$\delta^b = \frac{\delta^a p}{1 + \frac{\delta^a \sqrt{p}}{L}} \approx \delta^a p\Big(1 - \frac{\delta^a \sqrt{p}}{L}\Big), \tag{6}$$

where the approximation corresponds to a first-order Taylor expansion in the dimensionless term $\frac{\delta^a \sqrt{p}}{L} = \frac{\delta^a}{q^a}$. We now define the execution price of the trade, denoted by $\overline{p}$, as the average exchange rate obtained over the entire trade:

$$\overline{p} = \frac{\delta^b}{\delta^a} = \frac{p}{1 + \frac{\delta^a \sqrt{p}}{L}} \approx p\Big(1 - \frac{\delta^a \sqrt{p}}{L}\Big), \tag{7}$$

where the approximation corresponds to a first-order expansion in $\frac{\delta^a \sqrt{p}}{L}$. This represents the effective price paid by the trader per unit of token a over the full trade. In contrast to the spot price p, which reflects the marginal price before the trade, the execution price accounts for the cumulative effect of moving along the AMM curve. As emphasized in Cartea et al. (2025), Equation (7) is explicitly derived from the AMM pricing rule, highlighting a structural difference between AMMs and LOBs. Indeed, in the LOB literature, execution costs are typically modeled as linear in trade size, with a constant slope estimated from market data. Next, the slippage is defined as the relative deviation of the execution price from the spot price. From the first-order expansion in (7), it is approximated by $-\frac{\delta^a \sqrt{p}}{L}$. The slope of the slippage with respect to trade size depends on the spot price and therefore becomes stochastic under stochastic spot price dynamics.

The post-swap spot price is given by:

$$p^+ = \frac{q^b - \delta^b}{q^a + \delta^a} = \frac{p}{\big(1 + \frac{\delta^a \sqrt{p}}{L}\big)^2} \approx p\Big(1 - \frac{2\delta^a \sqrt{p}}{L}\Big), \tag{8}$$

where the approximation also corresponds to a first-order expansion in $\frac{\delta^a \sqrt{p}}{L}$. This expression reflects the marginal price immediately after the trade and thus incorporates the impact induced by the trade size. The relative price impact is defined as the relative difference between the post-swap spot price and the pre-swap spot price. From the first-order expansion in (8), it is approximated by $-\frac{2\delta^a \sqrt{p}}{L}$. The relative price impact is twice the slippage.

2.2 Optimal execution strategy

2.2.1 Problem formulation

We consider the problem of a trader willing to execute a large sell order of size $\xi > 0$ of token a over a fixed time horizon T. The time interval $[0, T]$ is divided into a regular partition: $0 = t_0 < t_1 < \cdots < t_N = T$, with constant time step $\Delta = \frac{T}{N}$. At each time t_n, the trader sells an amount δ_n (for the sake of readability, the superscript a is omitted), subject to the volume constraint:

$$\sum_{n=0}^{N} \delta_n = \xi. \tag{9}$$

We model the spot price as the combination of three components: (i) a fundamental price process, treated as an exogenous and stochastic process capturing market volatility and external factors; (ii) the cumulative price impact induced by previous trades; and (iii) the resilience of the liquidity pool, reflecting the gradual recovery of the spot price following each trade. First, we denote by $(f_m)_{m=0}^{N}$ the fundamental price process, i.e., the spot prices that would have prevailed in the absence of price impact from the execution schedule, with f_0 given as a deterministic initial price. Second, the impact on the spot price at time t_m induced by the previous trades $(\delta_n)_{n=0}^{m-1}$ is derived from the price impact formula (8). Third, the resilience of the liquidity pool is modeled by a convex combination of exponential price recovery terms. We write the spot price at time t_m, for $m = 0, \dots, N$, as:

$$p_m = f_m\Big(1 - \sum_{n=0}^{m-1}\sum_{j=0}^{J} \omega_j e^{-(m-n)\Delta\rho_j} \frac{2\delta_n\sqrt{f_n}}{L}\Big), \tag{10}$$

where $(\rho_j)_{j=0}^{J}$ are resilience parameters and $(\omega_j)_{j=0}^{J}$ the associated weights, satisfying $\rho_j \geq 0$, $\omega_j \geq 0$ for $j = 0, \dots, J$ and $\sum_{j=0}^{J} \omega_j = 1$. A permanent impact component can be incorporated into the modeling by taking a resilience parameter to zero. Equation (10) is constructed by successively applying the relative price impact formula (8) induced by previous trades, combined with a first-order expansion in the dimensionless terms $(\frac{\delta_n\sqrt{f_n}}{L})_{n=0}^{m-1}$. Then, we multiply the relative price impacts by the resilience factors $(\sum_{j=0}^{J} \omega_j e^{-(m-n)\Delta\rho_j})_{n=0}^{m-1}$. Moreover, the liquidity of the pool is assumed to remain constant throughout the trading horizon, i.e., LPs are passive. This modeling choice is supported by the empirical findings of Cartea et al. (2025), who report limited LP activity relative to LTs and stable pool liquidity over the relevant execution horizons.

In the same vein, we write the cash-flow at time t_m from selling δ_m units as:

$$\mathcal{C}_m = \delta_m f_m\Big(1 - \sum_{n=0}^{m-1}\sum_{j=0}^{J} \omega_j e^{-(m-n)\Delta\rho_j} \frac{2\delta_n\sqrt{f_n}}{L} - \frac{\delta_m\sqrt{f_m}}{L}\Big). \tag{11}$$

Equation (11) is derived by substituting the spot price from (10) into the cash-flow formula (6) and is also a first-order expansion in the dimensionless terms $(\frac{\delta_n\sqrt{f_n}}{L})_{n=0}^{m-1}$. Specifically, the execution price that determines the cash-flow at time t_m combines two effects both resulting from the trader's actions: the cumulative impact of past trades on the spot price and the slippage induced by the current trade.

We now formulate the execution problem of the trader. The objective is to determine the optimal sequence of trade sizes $\delta = \big(\delta_0, \dots, \delta_N\big)^\top$ that maximizes the expected total cash-flows, corresponding to the sum over time of the contributions defined in (11), subject to the volume constraint (9):

$$\begin{aligned} \delta^* = \underset{\delta}{\operatorname{argmax}} \quad & \mathbb{E}\Big[\sum_{n=0}^{N} \mathcal{C}_n\Big] \\ \text{s.t.} \quad & \sum_{n=0}^{N} \delta_n = \xi. \end{aligned} \tag{12}$$

The constraint ensures that the full order of size ξ is executed over the time interval $[0, T]$. The trades are not *a priori* required to be non-negative, as depending on the dynamics of the fundamental price, it may be optimal to buy (or oversell relative to ξ) and later unwind the excess. An analogous formulation applies to traders seeking to execute a large buy order.

2.2.2 General solution

We now state the general closed-form solution to the execution problem (12).

Proposition 1 (General solution)**.** *The optimal execution schedule is:*

$$\delta^* = \Big(\xi - \frac{L}{2}\mathbb{1}^\top A^{-1} B\Big)\frac{A^{-1}\mathbb{1}}{\mathbb{1}^\top A^{-1}\mathbb{1}} + \Big(\frac{L}{2}\mathbb{1}^\top A^{-1} B\Big)\frac{A^{-1}B}{\mathbb{1}^\top A^{-1}B}, \tag{13}$$

where $\mathbb{1} = \big(1, \dots, 1\big)^\top$*, the components of the vector* $B \in \mathbb{R}^{N+1}$ *are defined as:*

$$B_m = \mathbb{E}[f_m], \tag{14}$$

and the entries of the matrix $A \in \mathbb{R}^{(N+1)\times(N+1)}$ are given by:

$$A_{mn} = \begin{cases} \sum_{j=0}^{J} \omega_j e^{-(m-n)\rho_j \Delta} \mathbb{E}[f_m \sqrt{f_n}] & \text{if } n \leq m \\ \sum_{j=0}^{J} \omega_j e^{-(n-m)\rho_j \Delta} \mathbb{E}[f_n \sqrt{f_m}] & \text{if } n > m. \end{cases} \tag{15}$$

Equation (13) shows that the optimal execution schedule δ^* can be expressed as a linear combination of two distinct profiles: one driven by resilience and expected price impacts ($A^{-1}\mathbb{1}$), and another that also accounts for the expected price trajectory ($A^{-1}B$). This linear combination reflects the trade-off between minimizing market impact and exploiting expected price dynamics. The respective weights depend on the total order size ξ and the liquidity level L.

Corollary 1 (Martingale case)**.** *If the fundamental price process is a martingale, the optimal solution* (13) *reduces to:*

$$\delta^* = \xi \frac{A^{-1}\mathbb{1}}{\mathbb{1}^\top A^{-1}\mathbb{1}}, \tag{16}$$

where the matrix A is defined in (15)*.*

In the martingale case, the solution is independent of the liquidity level L: although liquidity affects the overall cost of execution, it does not influence the shape of the optimal schedule in the martingale case. The execution strategy is fully determined by the structure of resilience and expected price impacts.

Corollary 2 (Geometric Brownian motion case)**.** *We assume that the fundamental price follows a geometric Brownian motion evaluated at discrete times t_m, for $m = 0, \ldots, N$:*

$$f_m = f_0 e^{(\mu - \frac{\sigma^2}{2})m\Delta + \sigma W_m}, \tag{17}$$

where f_0 is the initial fundamental price, μ the drift, σ the volatility and W_m denotes the Brownian motion at time t_m. Under the condition $\mu < \frac{3\sigma^2}{4} + 4\min_j \rho_j$, the optimal strategy δ^* is unique. Moreover, in the single-kernel setting ($J = 0$), the inverse of matrix A admits a closed-form expression:

$$A^{-1} = \frac{1}{f_0\sqrt{f_0}} \begin{bmatrix} 1 + \frac{a^2}{\gamma_1} & -\frac{a}{\gamma_1} & 0 & \cdots & 0 \\ -\frac{a}{\gamma_1} & \frac{1}{\gamma_1} + \frac{a^2}{\gamma_2} & -\frac{a}{\gamma_2} & \cdots & 0 \\ 0 & -\frac{a}{\gamma_2} & \frac{1}{\gamma_2} + \frac{a^2}{\gamma_3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -\frac{a}{\gamma_N} \\ 0 & 0 & 0 & -\frac{a}{\gamma_N} & \frac{1}{\gamma_N} \end{bmatrix}, \tag{18}$$

where $a = e^{-(\rho-\mu)\Delta}$, $b_n = e^{\left(\frac{3\mu}{2} + \frac{3\sigma^2}{8}\right)n\Delta}$ and $\gamma_n = b_n - a^2 b_{n-1}$ for $n = 1, \ldots, N$.

The proofs are provided in Appendix A. While the inverse of A can be expressed explicitly in the single-kernel case, it must be computed numerically in the general multi-kernel setting.

2.2.3 Dynamic programming

Anticipating that no general solution is available in the CLAMMs case, we reformulate the execution problem (12) within a dynamic programming framework, in both closed-loop and open-loop settings. We start with the closed-loop formulation, in which the state variables at time t_n are given by the triplet $(x_n, (I_n^j)_{j=0}^J, f_n)$, where x_n is the remaining inventory, I_n^j the cumulative price impacts induced by the j-th resilience factor and f_n the fundamental price. The remaining inventory evolves according to:

$$\begin{cases} x_0 = \xi \\ x_{n+1} = x_n - \delta_n. \end{cases} \tag{19}$$

The dynamics of the cumulative price impacts are derived from (10) and read:

$$\begin{cases} I_0^j = 0 \\ I_{n+1}^j = e^{-\Delta\rho_j}\Big(I_n^j + \frac{2\delta_n\sqrt{f_n}}{L}\Big), \end{cases} \tag{20}$$

for $j = 0, \ldots, J$.

Let $v_n(x_n, (I_n^j)_{j=0}^J, f_n)$ denote the value function, representing the maximal expected proceeds from liquidating x_n units over the periods $t_n, \ldots, t_N$, given the current cumulative price impacts and fundamental price. From (12), the associated Bellman equation reads:

$$v_n(x_n, (I_n^j)_{j=0}^J, f_n) = \sup_{\delta_n} \ \delta_n f_n\big(1 - \sum_{j=0}^J \omega_j I_n^j - \frac{\delta_n\sqrt{f_n}}{L}\big) + \mathbb{E}\Big[v_{n+1}(x_{n+1}, (I_{n+1}^j)_{j=0}^J, f_{n+1})|f_n\Big]. \tag{21}$$

To satisfy the volume constraint (9), the terminal condition enforces complete liquidation:

$$v_N(x_N, (I_N^j)_{j=0}^J, f_N) = x_N f_N\Big(1 - \sum_{j=0}^J \omega_j I_N^j - \frac{x_N\sqrt{f_N}}{L}\Big), \tag{22}$$

which corresponds to selling the entire remaining inventory x_N at t_N in a single trade.

In contrast to Proposition 1, where the dynamics of the fundamental price remain unspecified, solving the Bellman equation and deriving the value function requires specifying them. We assume that the fundamental price follows a geometric Brownian motion dynamics (17). The explicit value function and optimal control in a closed-loop setting are provided in the following proposition.

Proposition 2 (Dynamic programming: closed-loop solution)**.** *Under the geometric Brownian motion assumption* (17)*, the value function reads:*

$$v_n(x_n, (I_n^j)_{j=0}^J, f_n) = x_n f_n\Big(A_n + \sum_{j=0}^J B_n^j I_n^j + C_n x_n\sqrt{f_n}\Big) + \sqrt{f_n}\Big(D_n + \sum_{j=0}^J E_n^j I_n^j + \sum_{j_1=0}^J\sum_{j_2=0}^J F_n^{j_1,j_2} I_n^{j_1} I_n^{j_2}\Big), \tag{23}$$

and the optimal control:

$$\delta_n^*(x_n, (I_n^j)_{j=0}^J, f_n) = \frac{1}{2\phi_{n+1}\sqrt{f_n}}\Big[\theta_{n+1}^1 + \sum_{j=0}^J I_n^j \theta_{n+1}^{2,j} + x_n\sqrt{f_n}\theta_{n+1}^3\Big], \tag{24}$$

where the coefficients ϕ_{n+1}, θ_{n+1}^1, $\big(\theta_{n+1}^{2,j_1}\big)_{j_1=0}^J$ and θ_{n+1}^3 are defined as:

$$\phi_{n+1} = \frac{1}{L} + \frac{2}{L}\sum_{j=0}^J B_{n+1}^j e^{(\mu-\rho_j)\Delta} - C_{n+1}e^{(\frac{3\mu}{2}+\frac{3\sigma^2}{8})\Delta} - \frac{4}{L^2}\sum_{j_1=0}^J\sum_{j_2=0}^J F_{n+1}^{j_1,j_2} e^{(\frac{\mu}{2}-\rho_{j_1}-\rho_{j_2}-\frac{\sigma^2}{8})\Delta}, \tag{25}$$

and,

$$\begin{aligned} \theta_{n+1}^1 &= 1 - A_{n+1}e^{\mu\Delta} + \frac{2}{L}\sum_{j=0}^J E_{n+1}^j e^{(\frac{\mu}{2}-\rho_j-\frac{\sigma^2}{8})\Delta}, \\ \theta_{n+1}^{2,j_1} &= -\omega_{j_1} - B_{n+1}^{j_1}e^{(\mu-\rho_{j_1})\Delta} + \frac{4}{L}\sum_{j_2=0}^J F_{n+1}^{j_1,j_2} e^{(\frac{\mu}{2}-\rho_{j_1}-\rho_{j_2}-\frac{\sigma^2}{8})\Delta}, \\ \theta_{n+1}^3 &= \frac{2}{L}\sum_{j=0}^J B_{n+1}^j e^{(\mu-\rho_j)\Delta} - 2C_{n+1}e^{(\frac{3\mu}{2}+\frac{3\sigma^2}{8})\Delta}. \end{aligned} \tag{26}$$

The coefficients $A_n, (B_n^j)_{j=0}^J, C_n, D_n, (E_n)_{j=0}^J$ *and* $(F_n)_{j_1,j_2=0}^J$ *are determined recursively as follows:*

$$\begin{cases} A_n = A_{n+1}e^{\mu\Delta} + \frac{1}{2\phi_{n+1}}\theta_{n+1}^1\theta_{n+1}^3 \\ B_n^j = B_{n+1}^j e^{(\mu-\rho_j)\Delta} + \frac{1}{2\phi_{n+1}}\theta_{n+1}^{2,j}\theta_{n+1}^3 \\ C_n = C_{n+1}e^{(\frac{3\mu}{2}+\frac{3\sigma^2}{8})\Delta} + \frac{1}{4\phi_{n+1}}(\theta_{n+1}^3)^2 \\ D_n = D_{n+1}e^{(\frac{\mu}{2}-\frac{\sigma^2}{8})\Delta} + \frac{1}{4\phi_{n+1}}(\theta_{n+1}^1)^2 \\ E_n^j = E_{n+1}^j e^{(\frac{\mu}{2}-\rho_j-\frac{\sigma^2}{8})\Delta} + \frac{1}{2\phi_{n+1}}\theta_{n+1}^1\theta_{n+1}^{2,j} \\ F_n^{j_1,j_2} = F_{n+1}^{j_1,j_2} e^{(\frac{\mu}{2}-\rho_{j_1}-\rho_{j_2}-\frac{\sigma^2}{8})\Delta} + \frac{1}{4\phi_{n+1}}\theta_{n+1}^{2,j_1}\theta_{n+1}^{2,j_2}, \end{cases} \tag{27}$$

with terminal conditions $A_N = 1$, $B_N^j = -\omega_j$, $C_N = -\frac{1}{L}$, $D_N = E_N^j = F_N^{j_1,j_2} = 0$.

Details on the derivation are provided in Appendix A. Whereas the general solution from Proposition 1 is static, the closed-loop solution is dynamic as it adjusts its execution path in response to price feedback.

We now turn to the open-loop formulation of the execution problem (12). The fundamental price is removed from the set of state variables and the problem is reformulated in expectation under the geometric Brownian motion assumption (17). Consequently, the strategy does not incorporate feedback from the evolving fundamental price during execution. We introduce $\tilde{I}_n^j$ the cumulative mean price impact at time t_n induced by the j-th resilience factor, for $j = 0, \dots, J$. The dynamics are:

$$\begin{cases} \tilde{I}_0^j = 0 \\ \tilde{I}_{n+1}^j = e^{-\Delta\rho_j}\Big(\tilde{I}_n^j + \frac{2\delta_n\sqrt{f_0}}{L}e^{(\frac{\mu}{2}+\frac{3\sigma^2}{8})n\Delta}\Big). \end{cases} \tag{28}$$

The inventory dynamics are the same as in the closed-loop setting (19).

In the open-loop setting, we denote the value function by $\tilde{v}$, and the Bellman equation reads:

$$\tilde{v}_n(x_n, (\tilde{I}_n^j)_{j=0}^J) = \sup_{\delta_n}\ \delta_n f_0 e^{\mu n\Delta}\big(1 - \sum_{j=0}^J \omega_j \tilde{I}_n^j - \frac{\delta_n\sqrt{f_0}}{L}e^{(\frac{\mu}{2}+\frac{3\sigma^2}{8})n\Delta}\big) + \tilde{v}_{n+1}(x_{n+1}, (\tilde{I}_{n+1}^j)_{j=0}^J). \tag{29}$$

To satisfy the volume constraint (9), the terminal condition also enforces complete liquidation:

$$\tilde{v}_N(x_N, (\tilde{I}_N^j)_{j=0}^J) = x_N f_0 e^{\mu N\Delta}\big(1 - \sum_{j=0}^J \omega_j \tilde{I}_N^j - \frac{x_N\sqrt{f_0}}{L}e^{(\frac{\mu}{2}+\frac{3\sigma^2}{8})N\Delta}\big). \tag{30}$$

The explicit value function and optimal control in the open-loop setting are provided in the following proposition.

Proposition 3 (Dynamic programming: open-loop solution)**.** *Under the geometric Brownian motion assumption* (17)*, the value function reads:*

$$\tilde{v}_n(x_n, (\tilde{I}_n^j)_{j=0}^J) = x_n f_0\Big(\tilde{A}_n + \sum_{j=0}^J \tilde{B}_n^j\tilde{I}_n^j + \tilde{C}_n x_n\sqrt{f_0}\Big) + \sqrt{f_0}\Big(\tilde{D}_n + \sum_{j=0}^J \tilde{E}_n^j\tilde{I}_n^j + \sum_{j_1=0}^J\sum_{j_2=0}^J \tilde{F}_n^{j_1,j_2}\tilde{I}_n^{j_1}\tilde{I}_n^{j_2}\Big), \tag{31}$$

and the optimal control:

$$\delta_n^*(x_n, (\tilde{I}_n^j)_{j=0}^J) = \frac{1}{2\tilde{\phi}_{n+1}\sqrt{f_0}}\Big[\tilde{\theta}_{n+1}^1 + \sum_{j=0}^J \tilde{I}_n^j\tilde{\theta}_{n+1}^{2,j} + x_n\sqrt{f_0}\tilde{\theta}_{n+1}^3\Big], \tag{32}$$

where the coefficients $\tilde{\phi}_{n+1}$*,* $\tilde{\theta}^1_{n+1}$*,* $\left(\tilde{\theta}^{2,j_1}_{n+1}\right)_{j_1=0}^{J}$ *and* $\tilde{\theta}^3_{n+1}$ *are defined as:*

$$\tilde{\phi}_{n+1} = \frac{1}{L} e^{(\frac{3\mu}{2}+\frac{3\sigma^2}{8})n\Delta} + \frac{2}{L}\sum_{j=0}^{J} \tilde{B}^j_{n+1} e^{-\rho_j\Delta} e^{(\frac{\mu}{2}+\frac{3\sigma^2}{8})n\Delta} - \tilde{C}_{n+1} - \frac{4}{L^2}\sum_{j_1=0}^{J}\sum_{j_2=0}^{J} \tilde{F}^{j_1,j_2}_{n+1} e^{-(\rho_{j_1}+\rho_{j_2})\Delta} e^{(\mu+\frac{3\sigma^2}{4})n\Delta}, \tag{33}$$

and,

$$\begin{aligned}
\tilde{\theta}^1_{n+1} &= e^{\mu n\Delta} - \tilde{A}_{n+1} + \frac{2}{L}\sum_{j=0}^{J} \tilde{E}^j_{n+1} e^{-\rho_j\Delta} e^{(\frac{\mu}{2}+\frac{3\sigma^2}{8})n\Delta}, \\
\tilde{\theta}^{2,j_1}_{n+1} &= -\omega_{j_1} e^{\mu n\Delta} - \tilde{B}^{j_1}_{n+1} e^{-\rho_{j_1}\Delta} + \frac{4}{L}\sum_{j_2=0}^{J} \tilde{F}^{j_1,j_2}_{n+1} e^{-(\rho_{j_1}+\rho_{j_2})\Delta} e^{(\frac{\mu}{2}+\frac{3\sigma^2}{8})n\Delta}, \\
\tilde{\theta}^3_{n+1} &= \frac{2}{L}\sum_{j=0}^{J} \tilde{B}^j_{n+1} e^{-\rho_j\Delta} e^{(\frac{\mu}{2}+\frac{3\sigma^2}{8})n\Delta} - 2\tilde{C}_{n+1}.
\end{aligned} \tag{34}$$

The coefficients $\tilde{A}_n, (\tilde{B}^j_n)_{j=0}^J, \tilde{C}_n, \tilde{D}_n, (\tilde{E}_n)_{j=0}^J$ *and* $(\tilde{F}_n)_{j_1,j_2=0}^J$ *are determined recursively as follows:*

$$\begin{cases}
\tilde{A}_n = \tilde{A}_{n+1} + \frac{1}{2\tilde{\phi}_{n+1}}\tilde{\theta}^1_{n+1}\tilde{\theta}^3_{n+1} \\
\tilde{B}^j_n = \tilde{B}^j_{n+1} e^{-\rho_j\Delta} + \frac{1}{2\tilde{\phi}_{n+1}}\tilde{\theta}^{2,j}_{n+1}\tilde{\theta}^3_{n+1} \\
\tilde{C}_n = \tilde{C}_{n+1} + \frac{1}{4\tilde{\phi}_{n+1}}(\tilde{\theta}^3_{n+1})^2 \\
\tilde{D}_n = \tilde{D}_{n+1} + \frac{1}{4\tilde{\phi}_{n+1}}(\tilde{\theta}^1_{n+1})^2 \\
\tilde{E}^j_n = \tilde{E}^j_{n+1} e^{-\rho_j\Delta} + \frac{1}{2\tilde{\phi}_{n+1}}\tilde{\theta}^1_{n+1}\tilde{\theta}^{2,j}_{n+1} \\
\tilde{F}^{j_1,j_2}_n = \tilde{F}^{j_1,j_2}_{n+1} e^{-(\rho_{j_1}+\rho_{j_2})\Delta} + \frac{1}{4\tilde{\phi}_{n+1}}\tilde{\theta}^{2,j_1}_{n+1}\tilde{\theta}^{2,j_2}_{n+1},
\end{cases} \tag{35}$$

with terminal conditions $\tilde{A}_N = e^{\mu N\Delta}$*,* $\tilde{B}^j_N = -\omega_j e^{\mu N\Delta}$*,* $\tilde{C}_N = -\frac{1}{L}e^{(\frac{\mu}{2}+\frac{3\sigma^2}{8})N\Delta}$*,* $\tilde{D}_N = \tilde{E}^j_N = \tilde{F}^{j_1,j_2}_N = 0$*.*

The open-loop solution is static, like the general one from Proposition 1. Empirically, it coincides with the solution derived in Corollary 2.

2.3 Numerical results

2.3.1 Illustrations

We illustrate the optimal trading strategy using the closed-form expression from Corollary 2 under the geometric Brownian motion assumption (17) with the following set of parameters: $T = 1$, $N = 10$, $f_0 = 1$, $L = 1000$, $\mu = 0$ and $\sigma = 0.3$. In the single-kernel setting, the closed-form inverse of A given in (18) is used, whereas in the general multi-kernel case, it is computed numerically. Three scenarios are investigated and Figure 1 displays the corresponding optimal execution schedules for $\xi = 1$.

Scenario 1 (exponential resilience) We first examine exponential recovery of the liquidity pool with no permanent impact, setting $J = 0$ and $\rho = 3$. With a single exponential kernel (i.e., $J = 0$), the index j in (10) is omitted. The resulting optimal strategy exhibits features reminiscent of a bucket-shaped profile, characterized by identical trades at the beginning and end of the trading horizon, and smaller constant sized trades in between. The symmetric bucket-shaped profile was originally observed in Obizhaeva and Wang (2013) for block-shaped LOBs without risk aversion. However, in contrast to the symmetric bucket-shaped profile, the profile obtained here is asymmetric. Indeed, the initial trade is larger than the final one, similar profiles are reported in Alfonsi et al. (2010) with decreasing power-law shaped LOBs. Appendix B establishes this property and studies the sensitivity of the strategy to model parameters, both within a two-period framework. Moreover, the intermediate trades exhibit a slight downward slope, a feature also observed

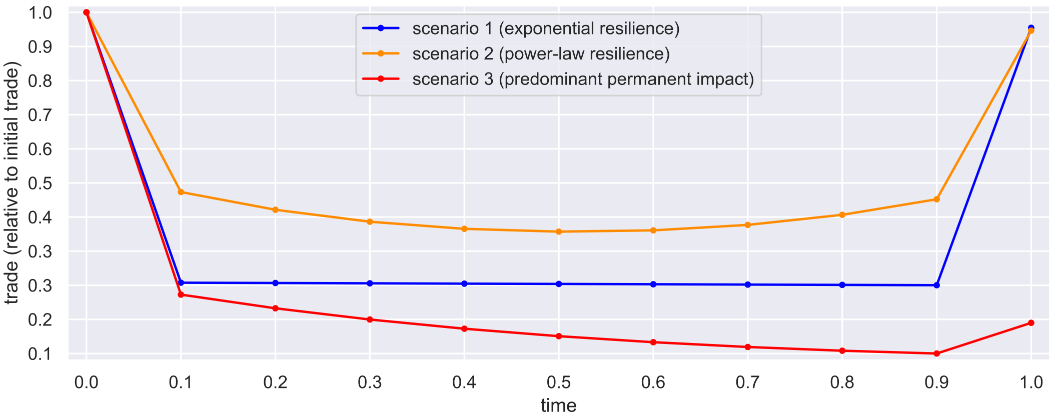


Figure 1: Optimal execution schedule under the three scenarios (relative to the initial trade for the sake of comparison); $T = 1$, $N = 10$, $L = 1000$, $f_0 = 1$, $\mu = 0$ and $\sigma = 0.3$.

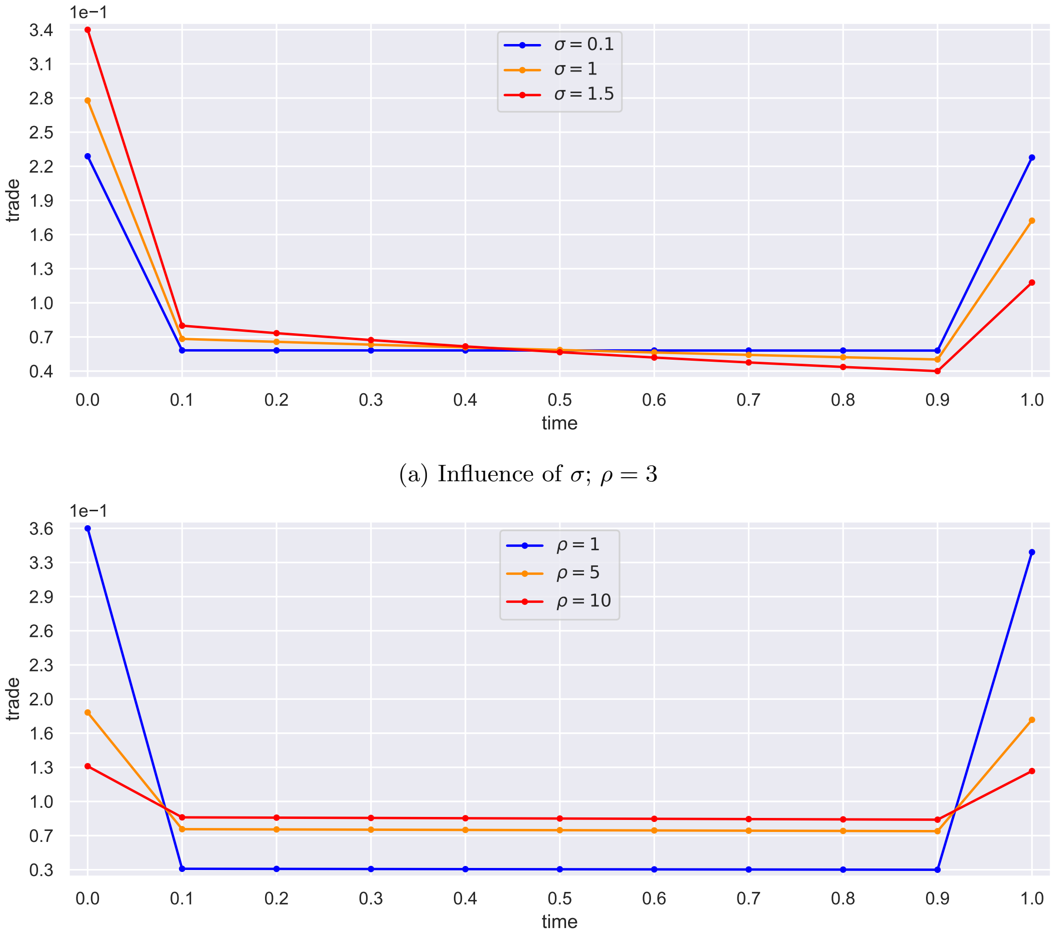
in Obizhaeva and Wang (2013) when risk aversion is included in the objective function. We recover a similar pattern here despite the absence of risk aversion. This behavior stems from the CPAMM framework, where both price impact and slippage depend on the spot price, which is stochastic and thus introduces uncertainty into the execution process. This asymmetry becomes more pronounced as the volatility parameter increases, reflecting a stronger incentive to front-load execution due to higher expected price impact and slippage of future trades (see Section 2.3.2).

Scenario 2 (power-law resilience) Next, a power-law recovery with no permanent impact is considered. The power-law kernel is approximated by a convex combinations of exponential kernels:

$$(1 + m\Delta\alpha)^{-\beta} \approx \sum_{j=0}^{J} \omega_j e^{-m\Delta\rho_j}, \tag{36}$$

where $\alpha \geq 0$ is a scaling parameter and $\beta \geq 0$ denotes the power-law exponent. Given α and β, the resilience parameters $(\rho_j)_{j=0}^J$ and the associated weights $(\omega_j)_{j=0}^J$ are obtained numerically by fitting the power-law kernel over the interval $[0, T]$. To do so, we use the SLSQP algorithm provided by the Scipy *minimize* function. We set $\alpha = 10$ and $\beta = 0.8$. In this setting, we take $J = 1$, which provides an accurate approximation of the power-law kernel. An advanced calibration methodology for larger values of J is presented in Bochud and Challet (2006). The optimal execution schedule exhibits a U-shaped profile, with trade sizes gradually increasing toward both ends of the trading interval. A symmetric U-shape profile is presented in Bouchaud et al. (2018, Section 21.2) and obtained under power-law decaying propagators without risk aversion. A more front-loaded U-shape profile is obtained in Busseti and Lillo (2012) by incorporating risk aversion into the objective function. As in Scenario 1, we obtain an asymmetric profile despite the absence of risk aversion. This asymmetry arises from the stochastic nature of market impact and slippage in our CPAMM framework.

Scenario 3 (predominant permanent impact) Finally, we consider a scenario under predominant permanent impact and limited exponential recovery, with parameters: $J = 1$, $\omega_0 = 0.99$, $\rho_0 = 0$, $\omega_1 = 0.01$ and $\rho_1 = 5$. The resulting optimal execution schedule is close to a front-loaded profile, in the spirit of the Almgren-Chriss schedule under a mean–variance objective (Almgren and Chriss, 2001). Despite the front-loaded nature of the schedule, the final trade remains significant relative to the previous ones, resulting in a hybrid between a front-loaded and a bucket-shaped profile.



(a) Influence of σ; $\rho = 3$

(b) Influence of ρ; $\sigma = 0.3$

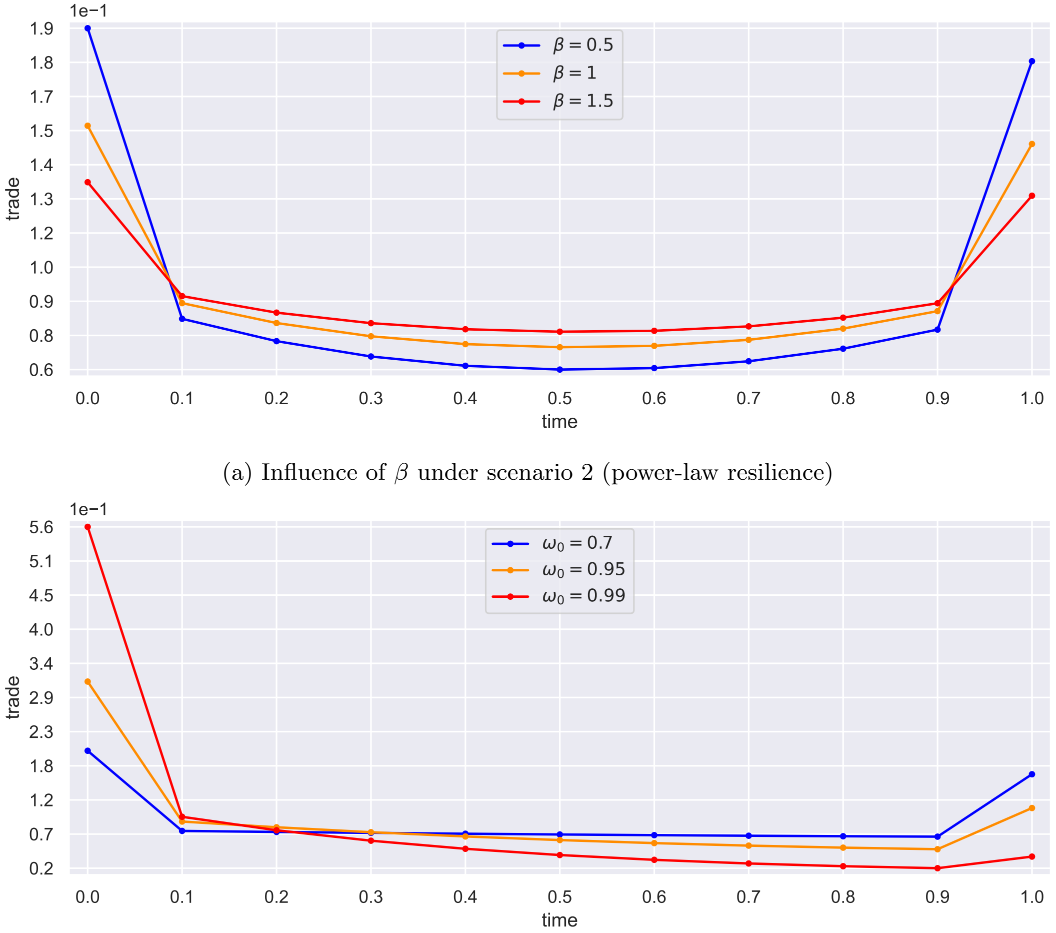
Figure 2: Influence of the parameters on the optimal execution schedule under scenario 1 (exponential resilience); $T = 1$, $N = 10$, $L = 1000$, $f_0 = 1$, $\mu = 0$ and $J = 0$.

2.3.2 Influence of the parameters

This section investigates the influence of the parameters on the optimal execution schedule.

Influence of the volatility In the single-kernel case ($J = 0$), corresponding to scenario 1 (exponential resilience), Figure 2a illustrates the influence of the volatility parameter σ. As the volatility increases, we observe a rotation in the execution profile toward the beginning of the trading schedule. The model allocates larger trade sizes at earlier stages, at the expense of the terminal ones. This behavior reflects the incentive of the trader to front-load execution due to higher expected price impact of future trades.

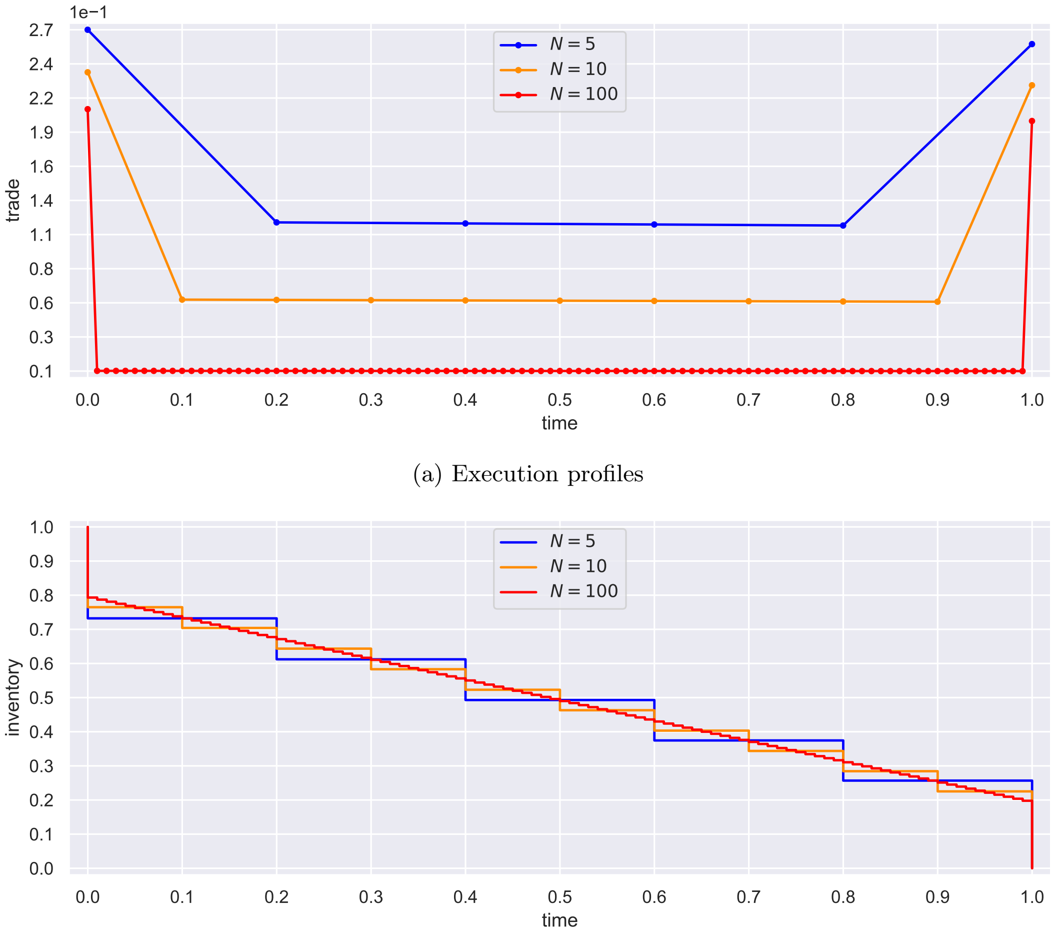
Influence of the resilience Still under scenario 1 in the single-kernel setting ($J = 0$), Figure 2b shows the influence of the resilience parameter ρ on the execution profile. As the resilience increases, the execution profile exhibits an upward parallel shift centered on the intermediate trades. Both the initial and terminal trades decrease in size, while intermediate trades are amplified. This shift enables the strategy to benefit from a stronger mean-reverting effect in prices, thereby reducing overall impact. Then, Figure 3a illustrates the influence of the power-law exponent β in the two-kernels setting ($J = 1$), corresponding to scenario 2. The



(a) Influence of β under scenario 2 (power-law resilience)

(b) Influence of ω_0 under scenario 3 (predominant permanent impact); $\rho_0 = 0$, $\rho_1 = 5$

Figure 3: Influence of the parameters on the optimal execution schedule; $T = 1$, $N = 10$, $L = 1000$, $f_0 = 1$, $\mu = 0$, $\sigma = 0.3$ and $J = 1$.



(a) Execution profiles

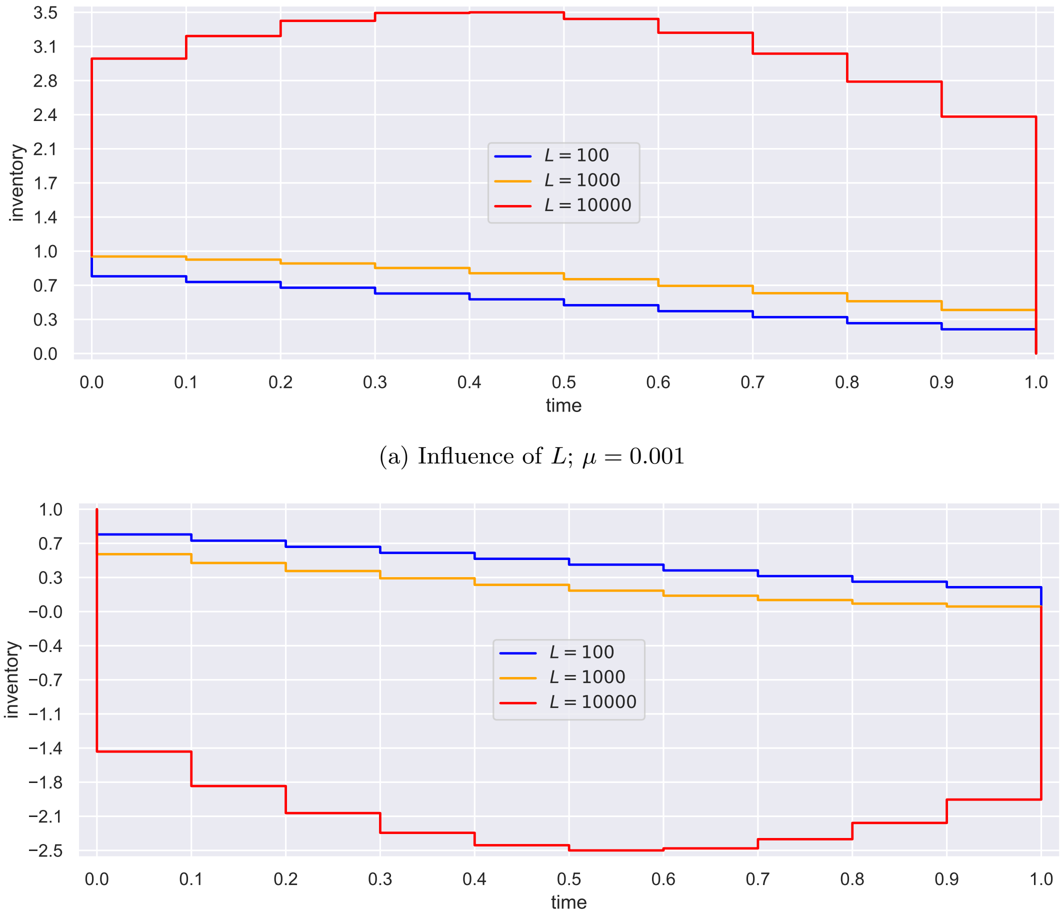
(b) Inventory paths

Figure 4: Influence of the trading frequency on the optimal execution schedule under scenario 1 (exponential resilience); $T = 1$, $L = 1000$, $f_0 = 1$, $\mu = 0$, $\sigma = 0.3$, $J = 0$ and $\rho = 3$.

resilience parameters are calibrated via the procedure described in Section 2.3.1. As β increases, the model also increases the size of the intermediary trades. Moreover, a higher β concentrates the resilience dynamics around a single dominant factor, thereby flattening the U-shaped profile. Finally, Figure 3b examines the effect of the permanent component of price impact under scenario 3. As the permanent impact becomes increasingly predominant, the execution strategy shifts toward a more front-loaded profile. When impact is fully permanent (i.e., $\omega_0 = 1$), the optimal strategy consists in liquidating the entire inventory at the first time step, as no mean-reverting effect remains to be exploited.

Influence of the trading frequency Figures 4 display the effect of the trading frequency N on both the optimal execution profile and the resulting inventory path in the scenario 1 (exponential resilience). As N increases, all intermediary trades decrease uniformly, and both the initial and final trades also decrease. Nevertheless, the overall structure of the strategy remains unchanged: the inventory paths intersect at all common time steps, indicating that the underlying execution dynamics are preserved.

Influence of the liquidity While liquidity has no impact on the optimal strategy in the martingale setting, it plays a significant role when the drift is non-zero. This effect is illustrated in Figures 5. When



(a) Influence of L; $\mu = 0.001$

(b) Influence of L; $\mu = -0.001$

Figure 5: Influence of the liquidity with non-zero drift on the optimal execution schedule under scenario 1 (exponential resilience); $T = 1$, $N = 10$, $f_0 = 1$, $\sigma = 0.3$, $J = 0$ and $\rho = 3$.

the drift is positive, the optimal strategy reduces the size of the early trades and favors liquidation closer to maturity in order to benefit from the expected price increase induced by the positive drift. As the liquidity of the pool increases, the strategy further decreases the size of the early trades, and once liquidity is sufficiently high, the strategy involves buying at the beginning of the period and unwinding the position at maturity. When the drift is negative the reasoning is reversed and the strategy tends to sell more at the beginning of the period and buy back at maturity in order to benefit from the expected price decrease.

2.3.3 Open-loop versus closed-loop

First, we compare the optimal strategy obtained in the open-loop setting, as formulated in Proposition 3, with the closed-loop counterpart from Proposition 2 evaluated along the mean fundamental price trajectory. We compute the mean and maximum absolute differences (expressed in basis points) across the three scenarios introduced in Section 2.3.1, as summarized in Table 1. The mean and maximum absolute differences are computed as:

$$\text{mean abs difference} = \frac{1}{N+1}\sum_{n=0}^{N}\left|\delta_n^*\left(x_n, (I_n^j)_{j=0}^J, f_0 e^{\mu n \Delta}\right) - \tilde{\delta}_n^*(x_n, (\tilde{I}_n^j)_{j=0}^J)\right|, \tag{37}$$

	scenario 1	scenario 2	scenario 3
mean abs difference (in basis points)	3	2	2
max abs difference (in basis points)	17	9	5

Table 1: Mean and maximum absolute differences between open-loop and closed-loop evaluated along the mean price trajectory optimal execution schedules; $T = 1$, $N = 10$, $L = 1000$, $f_0 = 1$, $\mu = 0$ and $\sigma = 0.3$.

$$\text{max abs difference} = \max_{n=0,\dots,N} \left| \delta_n^*\big(x_n, (I_n^j)_{j=0}^J, f_0 e^{\mu n \Delta}\big) - \tilde{\delta}_n^*\big(x_n, (\tilde{I}_n^j)_{j=0}^J\big) \right|. \tag{38}$$

Both the remaining inventory and the cumulative price impacts are computed forward in time according to their respective dynamics. Across all three scenarios, the open-loop and closed-loop evaluated along the mean fundamental price trajectory strategies yield similar results.

Second, under scenario 1 from Section 2.3.1, we compare the optimal strategy in the closed-loop setting evaluated along the mean fundamental price trajectory with those obtained under the following upward and downward price trajectories:

$$f_m^{\pm} = f_0 e^{(\mu - \frac{\sigma^2}{2}) m\Delta \pm 3\sigma m \sqrt{\Delta}}. \tag{39}$$

Figure 6a presents the resulting execution schedules. At the initial time, the executed trades are identical across trajectories. As prices diverge from the mean trajectory, the strategy adjusts accordingly. In the case of a persistently upward trend, it accelerates liquidation to benefit from rising prices. Conversely, under a downward trend, execution is delayed. Then, we analyze the optimal strategy under fundamental price trajectories subject to a persistent upward or downward multiplicative price bump $e^{(\mu - \frac{\sigma^2}{2})\Delta \pm 3\sigma\sqrt{\Delta}}$, applied at $t_n = 0.5$. The corresponding schedules are displayed in Figure 6b. In this case, the strategy reacts immediately at the time of the shock, then returns to the initial profile, shifted to account for changes in remaining inventory.

2.4 Incorporating fees

In this section, we discuss the integration of swap fees into the execution framework, the constant product formula (3) now reads:

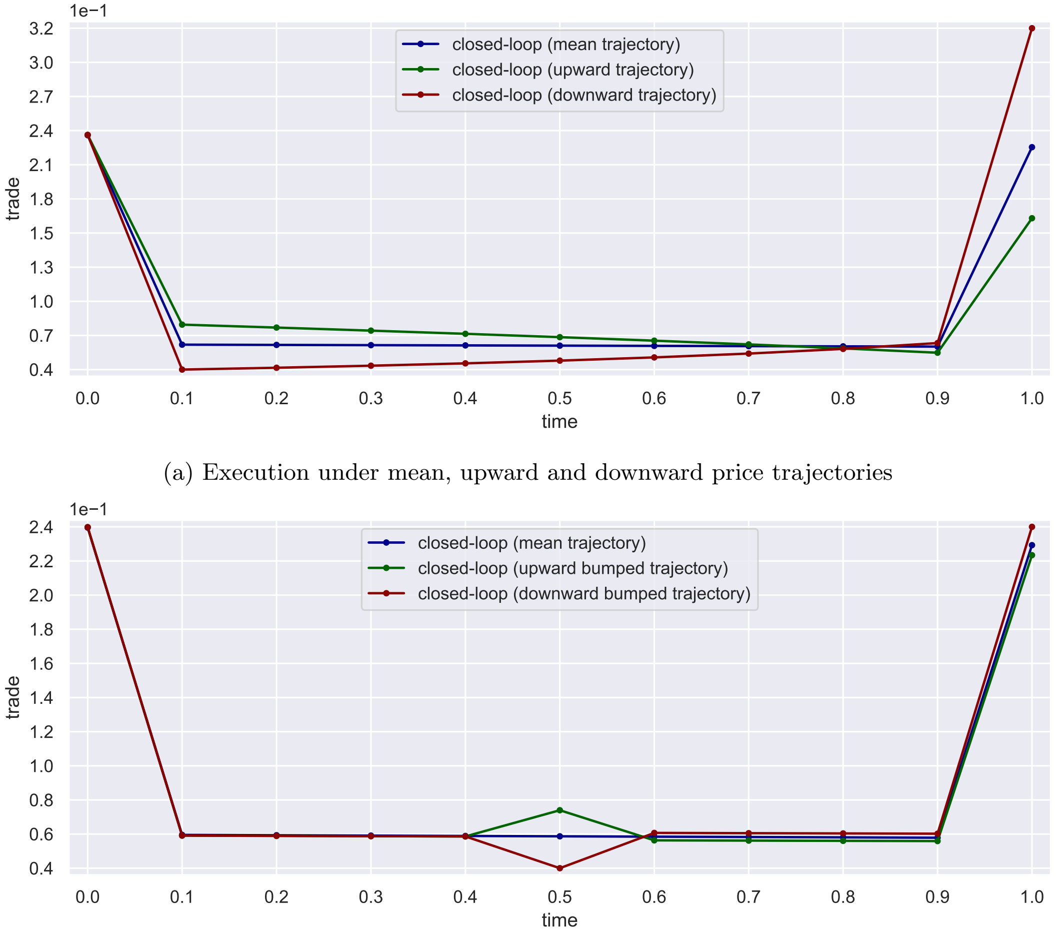
$$(q^a + (1 - \phi)\delta^a)(q^b - \delta^b) = L^2, \tag{40}$$

where ϕ denotes the swap fee. In Uniswap v2, a swap fee is applied to the input amount, and is fixed at $\phi = 0.3\%$. After the swap, liquidity is updated to $L^+ = \sqrt{(q^a + \delta^a)(q^b - \delta^b)}$ to reflect the fee-adjusted reserves. Equation (40) is valid only for positive trades (i.e., sell orders), since applying it to negative trades (i.e., buy orders) would imply that fees are received by the trader rather than paid to the pool. In this restricted setting, the optimal scheduling problem (12) still admits a closed-form solution, with appropriate adjustments to account for fees. Empirically, for the scenarios considered in Section 2.3, we observe that a fee level of $\phi = 0.3\%$ has negligible influence on the resulting optimal strategies. Extending the model to handle negative trades would require adjusting the fee structure depending on the sign of the trade, which considerably complicates the modeling framework. For these reasons, we omit fees in the paper.

3 From Uniswap v2 to v3

3.1 A two-layer liquidity framework

We consider a Uniswap v3 pool with two layers of liquidity: L_0 when the price is above the price threshold $\overline{p}$, and $L_1 < L_0$ when the price is below $\overline{p}$. Figure 7 illustrates the liquidity distribution as a function of price. We consider the same execution problem (12), reformulated within a dynamic programming framework under a closed-loop setting as in Section 2.2.3. In the two-layer liquidity framework, both cash-flows and



(a) Execution under mean, upward and downward price trajectories

(b) Execution under mean, upward and downward bumped price trajectories

Figure 6: Closed-loop optimal execution schedules evaluated along mean, upward and downward price trajectories; $T = 1$, $N = 10$, $L = 1000$, $f_0 = 1$, $\mu = 0$, $\sigma = 0.3$, $J = 0$ and $\rho = 3$.

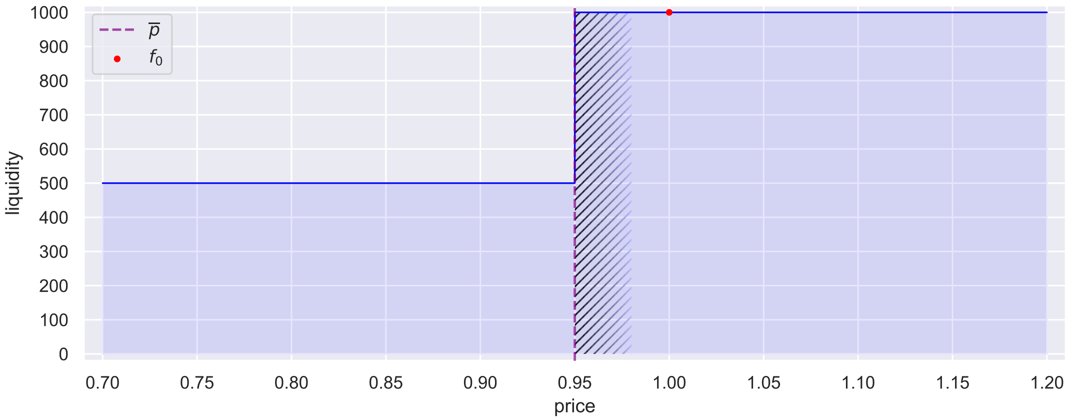


Figure 7: Illustration of the two-layer liquidity distribution. For a given trade size, the dashed zone indicates the spot price interval where the trade leads the spot price to cross from the high-liquidity layer to the low-liquidity one; $L_0 = 1000$, $L_1 = 500$, $f_0 = 1$ and $\overline{p} = 0.95$.

price impacts depend on whether the spot price is above or below the threshold $\overline{p}$. A third case arises when the spot price lies above $\overline{p}$, but the execution causes it to cross the threshold. In this case, both the total cash flows and the aggregate price impact are decomposed into two parts: one corresponding to the execution above the threshold, and the other to the execution below it. The corresponding dynamics of the cumulative price impacts read:

$$I_{n+1}^j = \begin{cases} e^{-\Delta\rho_j}\big(I_n^j + \frac{2\delta_n\sqrt{f_n}}{L_1}\big) & \text{if } p_n \le \overline{p} \\ e^{-\Delta\rho_j}\big(I_n^j + \frac{2\overline{\delta}_n\sqrt{f_n}}{L_0} + \frac{2(\delta_n - \overline{\delta}_n)\sqrt{\overline{p}}}{L_1}\big) & \text{if } p_n > \overline{p} \text{ and } \delta_n > \overline{\delta}_n \\ e^{-\Delta\rho_j}\big(I_n^j + \frac{2\delta_n\sqrt{f_n}}{L_0}\big) & \text{if } p_n > \overline{p} \text{ and } \delta_n \le \overline{\delta}_n, \end{cases} \tag{41}$$

for $j = 0, \dots, J$ where $I_0^j = 0$ and the quantity $\overline{\delta}_n$ corresponds to the trade size that brings the post-swap price down to the threshold $\overline{p}$, starting from an initial spot price above the threshold:

$$\overline{\delta}_n = \frac{L_0}{2f_n\sqrt{f_n}}(p_n - \overline{p}). \tag{42}$$

The dynamics of the inventory remain unchanged from Section 2.2.3.

In this framework, the Bellman equation reads:

$$v_n(x_n, (I_n^j)_{j=0}^J, f_n) = \sup_{\delta_n}\ \mathcal{C}_n + \mathbb{E}\Big[v_{n+1}(x_{n+1}, (I_{n+1}^j)_{j=0}^J, f_{n+1})\big|f_n\Big], \tag{43}$$

where the cash-flow at time t_n is:

$$\mathcal{C}_n = \begin{cases} \delta_n f_n\big(1 - \sum_{j=0}^J \omega_j I_n^j - \frac{\delta_n\sqrt{f_n}}{L_1}\big) & \text{if } p_n \le \overline{p} \\ \overline{\delta}_n f_n\big(1 - \sum_{j=0}^J \omega_j I_n^j - \frac{\overline{\delta}_n\sqrt{f_n}}{L_0}\big) & \\ \qquad + (\delta_n - \overline{\delta}_n)\overline{p}\big(1 - \sum_{j=0}^J \omega_j I_n^j - \frac{(\delta_n - \overline{\delta}_n)\sqrt{\overline{p}}}{L_1}\big) & \text{if } p_n > \overline{p} \text{ and } \delta_n > \overline{\delta}_n \\ \delta_n f_n\big(1 - \sum_{j=0}^J \omega_j I_n^j - \frac{\delta_n\sqrt{f_n}}{L_0}\big) & \text{if } p_n > \overline{p} \text{ and } \delta_n \le \overline{\delta}_n. \end{cases} \tag{44}$$

To satisfy the volume constraint (9), the terminal condition of the value function enforces complete liquidation using $\delta_N = x_N$ in (44). In this section, we consider a two-layer liquidity framework, which can be naturally generalized to multiple layers.

Due to the discontinuities in the model, closed-form expressions are no longer available. Therefore, in the following section, we present a numerical method to approximate the optimal strategy.

3.2 Discretization scheme

In this section, we present a numerical method to approximate the value function and the corresponding optimal control under the two-layer liquidity framework introduced in Section 3.1. We also consider the geometric Brownian motion assumption for the fundamental price, as described in Section 2.2.3. We adopt a discretization scheme, starting with the price domain. The price grid is defined for $k_f \in \{0, \dots, K_f\}$ as:

$$f^{(k_f)} = e^{Y^{(k_f)}}, \quad Y^{(k_f)} = m_T + z\sigma_T\Big(\frac{2k_f - K_f}{K_f}\Big), \tag{45}$$

where $z > 0$ controls the grid width, while $m_T = \log(f_0) + (\mu - \frac{\sigma^2}{2})T$ and $\sigma_T = \sigma\sqrt{T}$ correspond to the mean and standard deviation of the log-price at maturity. An alternative would be to use adaptive grids based on the mean and standard deviation of the log-price over time. We tested both fixed and adaptive grids, and our numerical experiments show that the methodology employing fixed grids yields more stable results.

Next, we discretize the inventory and cumulative price impact spaces. For $k_x \in \{0, \dots, K_x\}$, the inventory grid is defined by:

$$x^{(k_x)} = \frac{k_x \xi}{K_x}, \tag{46}$$

and for $k_i^j \in \{0, \dots, K_I\}$, the grid associated with the cumulative impact induced by the j-th resilience factor is given by:

$$I^{(k_i^j)} = e^{-\Delta\rho_j} \frac{2k_i^j \xi \sqrt{f^{(K_f)}}}{L_1 K_I}. \tag{47}$$

To approximate the conditional expectation in the Bellman equation, we introduce log-price midpoints between each $Y^{(k_f)}$:

$$y^{(k_f^\pm)} = \frac{1}{2}\big(Y^{(k_f)} + Y^{(k_f \pm 1)}\big), \tag{48}$$

with boundary values $y^{(0^-)} = -\infty$ and $y^{(K_f^+)} = +\infty$ to encompass the entire price domain. Given a current price $f^{(k)}$, the conditional expectation of the value function at the next time step is approximated by:

$$\mathbb{E}\big[v_{n+1}(x_{n+1}, (I_{n+1}^j)_{j=0}^J, f_{n+1}) | f_n = f^{(k)}\big] \approx \sum_{q=0}^{K_f} \omega_{k\to q} v_{n+1}(x_{n+1}, (I_{n+1}^j)_{j=0}^J, f^{(q)}), \tag{49}$$

where the transition probability weights $\omega_{k\to q}$ are computed as:

$$\omega_{k\to q} = \Phi(\frac{y^{(q+)} - \mu_k}{s}) - \Phi(\frac{y^{(q-)} - \mu_k}{s}), \tag{50}$$

with $\mu_k = \log(f^{(k)}) + (\mu - \frac{\sigma^2}{2})\Delta$, $s = \sigma\sqrt{\Delta}$ and Φ denotes the standard normal cumulative distribution function.

The algorithm proceeds backward in time. The terminal condition of the value function is given by (44) using $\delta_N = x_N$. Then, starting from this terminal condition and applying the Bellman recursion, we estimate the value function and the corresponding optimal control at each grid point of the fundamental price, inventory and cumulative impacts, using the approximation of the conditional expectation (49). Moreover, for a given inventory level indexed by k_x, the supremum in the Bellman equation (43) is evaluated over a discrete set of trade sizes drawn from the inventory grid: $x^{(k)}$ for $k \in 0, \dots, k_x$. This restriction enforces non-negativity of trades, thereby introducing no-buy constraints into the optimization. This design choice is motivated by computational efficiency and by the empirical observation that, in Section 2.3, none of the optimal profiles in the martingale case involve buy orders. We therefore restrict the use of the algorithm to the martingale case ($\mu = 0$). Allowing for buy orders would simply require extending the grid over which the supremum is computed. Appendix C provides a pseudocode of the backward algorithm. In Appendix D, we present consistency checks within the Uniswap v2 framework to empirically validate our numerical method. Specifically, we compare the numerical results in the degenerate case where $L_0 = L_1$ with the closed-form solution derived in Section 2.2.3, both evaluated along the mean price trajectory.

3.3 Numerical results

In this section, we begin by analyzing the impact of the threshold price $\overline{p}$ relative to the initial price f_0 on the optimal execution strategy evaluated along the mean price trajectory. The schedules are computed using the discretization scheme described in Section 3.2, and correspond to the solution to the dynamic programming problem under the two-layer liquidity structure introduced in Section 3.1. We adopt a grid resolution of $K_f = 500$, $K_x = 250$ and $K_I = 500$. In Appendix D, we vary the grid sizes to assess the stability of the numerical scheme. The threshold level is expressed in terms of a spread $\overline{s}$, defined by:

$$\overline{p} = f_0 + \overline{s}. \tag{51}$$

We distinguish three regimes depending on the position of the threshold $\overline{p}$ relative to the initial price f_0.

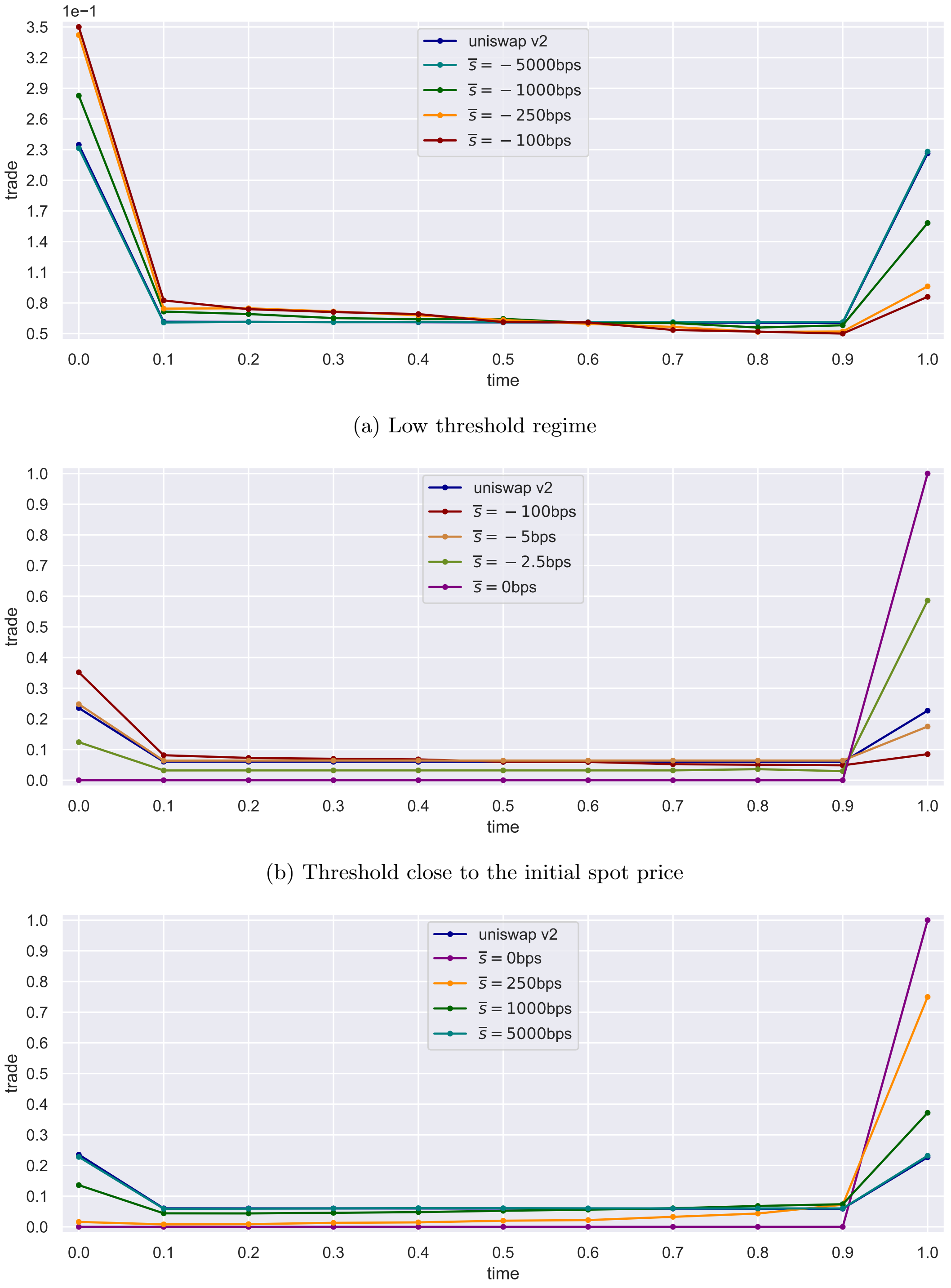
Low threshold regime In Figure 8a, we present the execution strategies for spreads ranging from -5000bps to -100bps. When the threshold price lies far below the initial price ($\overline{s} = -5000$bps), the optimal execution strategy coincides with the strategy obtained in the Uniswap v2 framework. Indeed, the probability that the price reaches the low-liquidity zone before maturity is very low. Consequently, the model disregards the discontinuity in liquidity and proceeds as if the pool exhibited constant liquidity. Then, as the threshold gradually approaches f_0, the execution strategy progressively front-loads the trades. Indeed, we observe a rotation in the execution profile, with more volume executed in the early steps of the horizon. This adjustment reflects the increasing probability, driven by price volatility, that the threshold will be crossed before maturity. In anticipation of the associated cost increase in the low-liquidity regime, the model accelerates execution to mitigate potential higher slippage and price impact.

Threshold close to the initial spot price The execution strategies for spreads ranging from -100bps to 0bps are reported in Figure 8b. When the threshold level is equal to the initial price ($\overline{s} = 0$), the optimal execution strategy consists of full liquidation at maturity. As previously mentioned, the strategy is evaluated along the mean price trajectory, which, in this case, remains at the threshold throughout the trading horizon. Consequently, the trader holds the position, waiting for a favorable price movement to benefit from lower costs in the higher-liquidity zone. Figure 9 illustrates the influence of the lower-layer liquidity parameter L_1 on the optimal strategy. As L_1 approaches L_0, the optimal strategy gradually transitions from full liquidation at maturity to the Uniswap v2 optimal execution schedule when L_1 reaches L_0.

High threshold regime Figure 8c illustrates the execution profiles for spreads ranging from 0bps to 5000bps. Strategies for spreads under 100bps are omitted due to their strong similarity. When the threshold price is set above the initial price, the execution strategy shifts again toward earlier liquidation. The probability of entering the low-liquidity region becomes lower, and the model gradually converges toward the same constant-liquidity execution profile. Indeed, when the price dynamics follow a martingale, the optimal strategy under constant liquidity is independent of the liquidity level (see Section 2.2).

4 Discussion and conclusion

In this paper, we study the optimal execution problem on AMMs, aiming to maximize the sum of expected cash-flows under transient price impact. The study covers both CPAMMs and CLAMMs. We derive closed-form solutions under general fundamental price dynamics for CPAMMs, and analyze the resulting strategies under geometric Brownian motion dynamics. We also solve the reformulated problem within a dynamic programming framework to enrich the strategy with price feedback. For CLAMMs, closed-form solutions are



(a) Low threshold regime

(b) Threshold close to the initial spot price

(c) High threshold regime

Figure 8: Influence of the spread $\overline{s}$ on the optimal execution schedule; $T = 1$, $N = 10$, $L_0 = 1000$, $L_1 = 500$, $f_0 = 1$, $\mu = 0$, $\sigma = 0.3$, $J = 0$, $\rho = 3$, $K_f = 500$, $K_x = 250$ and $K_I = 500$.

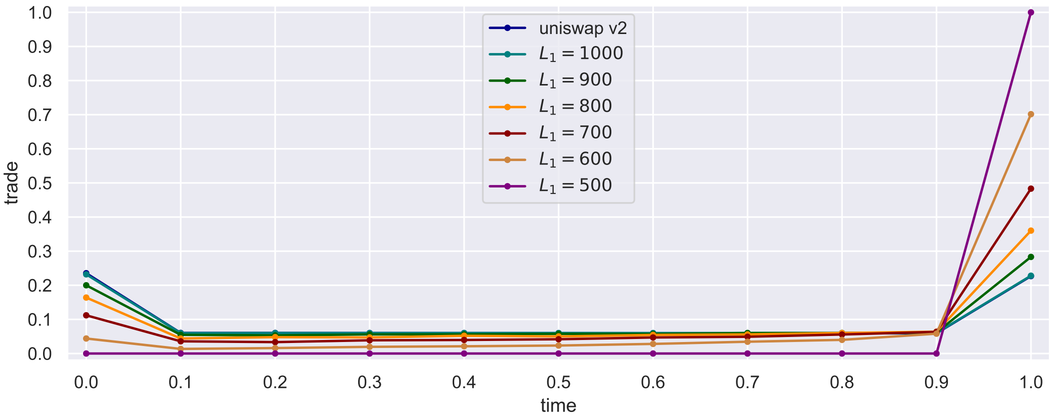


Figure 9: Influence of L_1 on the optimal execution schedule; $T = 1$, $N = 10$, $L_0 = 1000$, $f_0 = 1$, $\overline{s} = 0$, $\mu = 0$, $\sigma = 0.3$, $J = 0$, $\rho = 3$, $K_f = 500$, $K_x = 250$ and $K_I = 500$.

no longer available and the optimal strategy is then approximated numerically using a discretization scheme. The analysis is developed within a two-layer liquidity framework, which naturally extends to multiple layers. Numerical results are obtained using synthetic data. The application to real market data, including the calibration of resilience kernels, will be conducted in a companion paper.

In the CPAMMs case, exponential resilience leads to an asymmetric bucket-shaped execution profile, while power-law resilience results in an asymmetric U-shaped profile. In the literature, such asymmetries typically stem from risk aversion. In our setting, they arise from the endogenous and stochastic nature of market impact and slippage. Moreover, incorporating a mean-variance objective function following the approach of Almgren and Chriss (2001) would enrich the model from a risk-management perspective. This extension, however, presents significant mathematical challenges which we leave for future investigation.

The optimal execution strategy on CPAMMs is independent of the liquidity level of the pool in the martingale setting. In contrast, under the two-layer liquidity structure of CLAMMs, the strategy depends on both the liquidity levels and the position of the spot price relative to the price threshold at which the liquidity changes. When liquidity is more favorable, the strategy tends to front-load execution to reduce exposure to the risk of falling into the low-liquidity region following an adverse price movement, where execution costs are higher. Conversely, when the liquidity level is less favorable, the strategy postpones execution, waiting for a favorable price movement to benefit from lower costs in the higher-liquidity zone. We believe that this work may help practitioners in the DeFi ecosystem better understand execution mechanisms and reduce associated costs.

Finally, we assume that the liquidity of the pool is constant throughout the trading horizon, i.e., LPs remain passive. In the spirit of Cartea et al. (2025), a natural extension would be to incorporate stochastic liquidity dynamics. This extension will be explored in future research.

Acknowledgments

The authors would like to thank Mohamed Frihat, Vincent Danos, Hamza El Khalloufi and Alejandro Roigé Vázquez for fruitful discussions.

This work used HPC resources from the “Mésocentre” computing center of CentraleSupélec and École Normale Supérieure Paris-Saclay supported by CNRS and Région Île-de-France.

Disclosure of interest

The authors have no competing interests to declare.

References

Adams, H., Zinsmeister, N., and Robinson, D. (2020). Uniswap v2 Core. Available at `https://app.uniswap.org/whitepaper.pdf`.

Adams, H., Zinsmeister, N., Salem, M., Keefer, R., and Robinson, D. (2021). Uniswap v3 Core. Available at `https://app.uniswap.org/whitepaper-v3.pdf`.

Alfonsi, A., Fruth, A., and Schied, A. (2008). Constrained portfolio liquidation in a limit order book model. *Banach Center Publ*, 83:9–25.

Alfonsi, A., Fruth, A., and Schied, A. (2010). Optimal execution strategies in limit order books with general shape functions. *Quantitative finance*, 10(2):143–157.

Almgren, R. and Chriss, N. (1999). Value under liquidation. *Risk*, 12(12):61–63.

Almgren, R. and Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk*, 3:5–40.

Almgren, R. F. (2003). Optimal execution with nonlinear impact functions and trading-enhanced risk. *Applied mathematical finance*, 10(1):1–18.

Angeris, G., Evans, A., Chitra, T., and Boyd, S. (2022). Optimal routing for constant function market makers. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 115–128.

Bochud, T. and Challet, D. (2006). Optimal approximations of power-laws with exponentials. *arXiv preprint arXiv:physics/0605149*.

Bouchaud, J.-P., Bonart, J., Donier, J., and Gould, M. (2018). *Trades, quotes and prices: financial markets under the microscope*. Cambridge University Press.

Busseti, E. and Lillo, F. (2012). Calibration of optimal execution of financial transactions in the presence of transient market impact. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(09):P09010.

Cartea, Á., Drissi, F., and Monga, M. (2023). Execution and statistical arbitrage with signals in multiple automated market makers. In *2023 IEEE 43rd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 37–42. IEEE.

Cartea, Á., Drissi, F., and Monga, M. (2025). Decentralised finance and automated market making: Execution and speculation. *Journal of Economic Dynamics and Control*, page 105134.

Cartea, Á., Jaimungal, S., and Penalva, J. (2015). *Algorithmic and high-frequency trading*. Cambridge University Press.

Chitra, T., Kulkarni, K., and Srinivasan, K. (2025). Optimal Routing in the Presence of Hooks: Three Case Studies. *arXiv preprint arXiv:2502.02059*.

Curato, G., Gatheral, J., and Lillo, F. (2017). Optimal execution with non-linear transient market impact. *Quantitative Finance*, 17(1):41–54.

Danos, V., Khalloufi, H. E., and Prat, J. (2021). Global order routing on exchange networks. In *International Conference on Financial Cryptography and Data Security*, pages 207–226. Springer.

Danos, V., Watson, L. M., El Khalloufi, H., and Valencia, S. (2024). On-chain optimal aggregation of Uniswap v3 clones. *HAL preprint hal-04800479v2*.

Diamandis, T., Resnick, M., Chitra, T., and Angeris, G. (2023). An efficient algorithm for optimal routing through constant function market makers. In *International Conference on Financial Cryptography and Data Security*, pages 128–145. Springer.

Egorov, M. (2019). StableSwap - efficient mechanism for Stablecoin liquidity. Available at `https://classic.curve.finance/files/stableswap-paper.pdf`.

Gatheral, J., Schied, A., and Slynko, A. (2012). Transient linear price impact and fredholm integral equations. *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics*, 22(3):445–474.

Guéant, O. (2016). *The Financial Mathematics of Market Liquidity: From optimal execution to market making*. CRC Press.

Martinelli, F. and Mushegian, N. (2019). A non-custodial portfolio manager, liquidity provider, and price sensor. Available at `https://docs.balancer.fi/whitepaper.pdf`.

Obizhaeva, A. A. and Wang, J. (2013). Optimal trading strategy and supply/demand dynamics. *Journal of Financial markets*, 16(1):1–32.

Appendix

A Proofs

A.1 Proof of Proposition 1 and Corollary 1

Proof. By introducing a Lagrange multiplier $\lambda \in \mathbb{R}$ for the equality $\sum_{m=0}^{N} \delta_m = \xi$ in (12), we obtain:

$$\delta^* = \underset{\delta}{\operatorname{argmax}} \quad \mathbb{E}\Big[\sum_{n=0}^{N} \mathcal{C}_n\Big] + \lambda\big(\xi - \sum_{n=0}^{N} \delta_n\big). \tag{52}$$

For $m = 0, \dots, N$, the first-order condition from (52) reads:

$$\frac{\partial}{\partial \delta_m} \mathbb{E}\Big[\sum_{n=0}^{N} \mathcal{C}_n\Big] = \lambda, \tag{53}$$

where,

$$\begin{aligned} \frac{\partial}{\partial \delta_m} \mathbb{E}\Big[\sum_{n=0}^{N} \mathcal{C}_n\Big] = \mathbb{E}[f_m] &- \frac{2}{L} \sum_{n=0}^{m} \delta_n^* \Big(\sum_{j=0}^{J} \omega_j e^{-(m-n)\Delta \rho_j} \mathbb{E}[f_m \sqrt{f_n}] \Big) \\ &- \frac{2}{L} \sum_{n=m+1}^{N} \delta_n^* \Big(\sum_{j=0}^{J} \omega_j e^{-(n-m)\Delta \rho_j} \mathbb{E}[f_n \sqrt{f_m}] \Big). \end{aligned} \tag{54}$$

We can rewrite the $N+1$ first-order conditions (53) in compact matrix form as:

$$A\delta^* = \frac{L}{2}\big(B - \lambda \mathbb{1}\big), \tag{55}$$

where $\mathbb{1} = \big(1, \dots, 1\big)^\top$. The vector $B \in \mathbb{R}^{N+1}$ and the matrix $A \in \mathbb{R}^{(N+1)\times(N+1)}$, defined in (14) and (15) respectively, are derived from (54). Substituting (55) in the volume constraint (9) implies:

$$\lambda^* = \frac{\mathbb{1}^\top A^{-1} B - \frac{2\xi}{L}}{\mathbb{1}^\top A^{-1} \mathbb{1}}, \tag{56}$$

and plugging λ^* in (55) yields the closed-form optimal execution schedule:

$$\delta^* = \frac{L}{2}(A^{-1}B - \lambda^* A^{-1}\mathbb{1}). \tag{57}$$

When the fundamental price process is a martingale, the expected price at each time step remains constant and equal to the initial price. The vector B reduces to:

$$B = f_0\mathbb{1}. \tag{58}$$

Plugging (58) into the general solution (57), the optimal execution schedule simplifies to (16). □

A.2 Proof of Corollary 2

Proof. Under the geometric Brownian motion assumption (17), the expectations appearing in (14) and (15) admit closed-form expressions. For $m = 0, \dots, N$, the expected fundamental price reads $\mathbb{E}[f_m] = f_0 e^{\mu m \Delta}$. Moreover, for $n \leq m$, we have:

$$\mathbb{E}[f_m\sqrt{f_n}] = f_0\sqrt{f_0}e^{\mu(m+\frac{n}{2})\Delta + \frac{3\sigma^2}{8}n\Delta}, \tag{59}$$

and for $n > m$:

$$\mathbb{E}[f_n\sqrt{f_m}] = f_0\sqrt{f_0}e^{\mu(n+\frac{m}{2})\Delta + \frac{3\sigma^2}{8}m\Delta}. \tag{60}$$

We start with a single resilience factor (i.e., $J = 0$), for which the matrix A from (15) takes the following form, up to the multiplicative factor $f_0\sqrt{f_0}$, which is omitted for notational clarity:

$$A = \begin{bmatrix} b_0 & ab_0 & a^2b_0 & a^3b_0 & \cdots & a^Nb_0 \\ ab_0 & b_1 & ab_1 & a^2b_1 & \cdots & a^{N-1}b_1 \\ a^2b_0 & ab_1 & b_2 & ab_2 & \cdots & a^{N-2}b_2 \\ a^3b_0 & a^2b_1 & ab_2 & b_3 & \cdots & a^{N-3}b_3 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a^Nb_0 & a^{N-1}b_1 & a^{N-2}b_2 & a^{N-3}b_3 & \cdots & b_N \end{bmatrix}. \tag{61}$$

where $a = e^{-(\rho-\mu)\Delta}$ and $b_n = e^{\left(\frac{3\mu}{2}+\frac{3\sigma^2}{8}\right)n\Delta}$. Let $e_0, \dots, e_N$ denote the canonical basis of $\mathbb{R}^{N+1}$ and, in the spirit of Alfonsi et al. (2008), we define the vectors $v_0, \dots, v_N$ by the recursive formula:

$$\begin{cases} v_0 = e_0 \\ v_n = v_{n-1}a + e_n\sqrt{\gamma_n}, \end{cases} \tag{62}$$

where $\gamma_n = b_n - a^2b_{n-1}$ for $n = 1, \dots, N$, with $\gamma_0 = 1$, and subject to the condition on the drift: $\mu < \frac{3\sigma^2}{4} + 4\rho$ to ensure $\gamma_n > 0$ for $n = 1, \dots, N$. This condition is automatically satisfied in the martingale case (i.e., when $\mu = 0$). The matrix A is the corresponding Gram matrix: $\langle v_m, v_n\rangle = a^{|m-n|}b_{\min(m,n)} = A_{mn}$ for $m = 0, \dots, N$ and $n = 0, \dots, N$. Indeed, first by induction: $\langle v_n, v_n\rangle = b_n$. Second, for $n < m$ we have: $\langle v_m, v_n\rangle = a^{m-n}\langle v_n, v_n\rangle$ and for $m < n$: $\langle v_m, v_n\rangle = a^{n-m}\langle v_m, v_m\rangle$. Thus, the matrix A is positive definite. Therefore, the objective function is strictly concave as the associated Hessian matrix is given by $-\frac{2}{L}A$. In addition, the volume constraint $\sum_{n=0}^N \delta_n = \xi$ defines a convex feasible set. Together, these properties guarantee that the optimal strategy δ^* in (13) is the unique maximizer of the execution problem (12).

From the recursive formula (62), we also have:

$$v_n = \sum_{k=0}^{n}\left[a^{n-k}\sqrt{\gamma_k}\right]e_k. \tag{63}$$

Let V be the upper triangular matrix whose columns are the vectors $v_0, \dots, v_N$:

$$V = \begin{bmatrix} 1 & a & a^2 & \cdots & a^N \\ 0 & \sqrt{\gamma_1} & a\sqrt{\gamma_1} & \cdots & a^{N-1}\sqrt{\gamma_1} \\ 0 & 0 & \sqrt{\gamma_2} & \cdots & a^{N-2}\sqrt{\gamma_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sqrt{\gamma_N} \end{bmatrix}. \tag{64}$$

Then, $A = V^\top V$. Indeed, for any $m = 0, \dots, N$ and $n = 0, \dots, N$:

$$(V^\top V)_{mn} = \sum_{k=0}^{N} V_{km} V_{kn} = \sum_{k=0}^{\min(m,n)} V_{km} V_{kn}, \tag{65}$$

since V is upper triangular. Substituting the expression for V_{km} and V_{kn}:

$$(V^\top V)_{mn} = \sum_{k=0}^{\min(m,n)} \left[a^{m-k}\sqrt{\gamma_k}\right]\left[a^{n-k}\sqrt{\gamma_k}\right] = a^{m+n} \sum_{k=0}^{\min(m,n)} a^{-2k}\big(b_k - a^2 b_{k-1}\big), \tag{66}$$

where $b_{-1} = 0$. This sum telescopes to $a^{-2\min(m,n)} b_{\min(m,n)}$, yielding $(V^\top V)_{mn} = a^{|m-n|} b_{\min(m,n)} = A_{mn}$. The diagonals coefficients of the upper triangular matrix V are all strictly positives. Then, V is invertible and $A^{-1} = V^{-1}(V^{-1})^\top$. Finally, the inverse matrix A^{-1} is tridiagonal and is given by (18).

In the general multi-kernel case, let $e_0, \dots, e_N$ denote the canonical basis of $\mathbb{R}^{N+1}$, and let $f_0, \dots, f_J$ denote the canonical basis of $\mathbb{R}^{J+1}$. For $j = 0, \dots, J$, we define the vectors $v_0^j, \dots, v_N^j$ by the recursive formula:

$$\begin{cases} v_0^j = e_0 \\ v_n^j = v_{n-1}^j a_j + e_n \sqrt{b_n - a_j^2 b_{n-1}}, \end{cases} \tag{67}$$

where $a_j = e^{-(\rho_j - \mu)\Delta}$, $b_n = e^{\left(\frac{3\mu}{2} + \frac{3\sigma^2}{8}\right)n\Delta}$ and subject to the condition on the drift: $\mu < \frac{3\sigma^2}{4} + 4\min_j \rho_j$, to ensure $b_n - a_j^2 b_{n-1} > 0$ for $n = 1, \dots, N$ and $j = 0, \dots, J$. Then, we define:

$$v_n = \sum_{j=0}^{J} \sqrt{\omega_j}\,(f_j \otimes v_n^j). \tag{68}$$

The matrix A is the corresponding Gram matrix:

$$\langle v_m, v_n \rangle = \sum_{j=0}^{J} \omega_j \langle v_m^j, v_n^j \rangle = \sum_{j=0}^{J} \omega_j a_j^{|m-n|} b_{\min(m,n)} = A_{mn}. \tag{69}$$

Following the same arguments as in the single resilience factor case, the objective function is strictly concave and the optimal strategy δ^* in (13) is the unique maximizer of the execution problem (12). However, the methodology used to derive the inverse of A in the single-kernel case does not extend to the general multi-kernel setting. □

A.3 Proof of Proposition 2

Proof. We consider the *ansatz* (23) with coefficients A_n, $(B_n^j)_{j=0}^J$, C_n, D_n, $(E_n)_{j=0}^J$ and $(F_n)_{j_1,j_2=0}^J$ determined by backward recursion. Substituting (23) into the Bellman equation (21) yields a quadratic objective in δ_n:

$$\begin{aligned}
v_n(x_n,(I_n^j)_{j=0}^J,f_n) = \sup_{\delta_n}\ & -\delta_n^2 f_n\sqrt{f_n}\Big[\frac{1}{L}+\frac{2}{L}\sum_{j=0}^J B_{n+1}^j e^{(\mu-\rho_j)\Delta} - C_{n+1}e^{(\frac{3\mu}{2}+\frac{3\sigma^2}{8})\Delta} \\
& - \frac{4}{L^2}\sum_{j_1=0}^J\sum_{j_2=0}^J F_{n+1}^{j_1,j_2} e^{(\frac{\mu}{2}-\rho_{j_1}-\rho_{j_2}-\frac{\sigma^2}{8})\Delta}\Big] \\
& + \delta_n\Big[f_n\big(1-A_{n+1}e^{\mu\Delta}+\frac{2}{L}\sum_{j=0}^J E_{n+1}^j e^{(\frac{\mu}{2}-\rho_j-\frac{\sigma^2}{8})\Delta}\big) \\
& + f_n\sum_{j_1=0}^J I_n^{j_1}\big(-\omega_{j_1}-B_{n+1}^{j_1}e^{(\mu-\rho_{j_1})\Delta}+\frac{4}{L}\sum_{j_2=0}^J F_{n+1}^{j_1,j_2}e^{(\frac{\mu}{2}-\rho_{j_1}-\rho_{j_2}-\frac{\sigma^2}{8})\Delta}\big) \\
& + x_n f_n\sqrt{f_n}\big(\frac{2}{L}\sum_{j=0}^J B_{n+1}^j e^{(\mu-\rho_j)\Delta} - 2C_{n+1}e^{(\frac{3\mu}{2}+\frac{3\sigma^2}{8})\Delta}\big)\Big] \\
& + x_n f_n\Big[A_{n+1}e^{\mu\Delta}+\sum_{j=0}^J B_{n+1}^j e^{(\mu-\rho_j)\Delta}I_n^j + C_{n+1}e^{(\frac{3\mu}{2}+\frac{3\sigma^2}{8})\Delta}x_n\sqrt{f_n}\Big] \\
& + \sqrt{f_n}\Big[D_{n+1}e^{(\frac{\mu}{2}-\frac{\sigma^2}{8})\Delta}+\sum_{j=0}^J E_{n+1}^j e^{(\frac{\mu}{2}-\rho_j-\frac{\sigma^2}{8})\Delta}I_n^j \\
& + \sum_{j_1=0}^J\sum_{j_2=0}^J F_{n+1}^{j_1,j_2}e^{(\frac{\mu}{2}-\rho_{j_1}-\rho_{j_2}-\frac{\sigma^2}{8})\Delta}I_n^{j_1}I_n^{j_2}\Big],
\end{aligned} \tag{70}$$

and leads to the explicit control (24). The substitution of the optimal control into the Bellman equation (21) and the terminal condition (22), and identification of coefficients in the polynomial basis yields the backward recursions (27). □

B Optimal execution on Uniswap v2: the two-period martingale case

In the case $\mu = 0$, $N = 1$ and $J = 0$, the matrix A defined in (15) reads:

$$A = f_0\sqrt{f_0}\begin{bmatrix} 1 & e^{-\rho\Delta} \\ e^{-\rho\Delta} & e^{\frac{3\sigma^2}{8}\Delta} \end{bmatrix}. \tag{71}$$

The corresponding optimal trades are:

$$\delta_0^* = \xi\frac{e^{\frac{3\sigma^2}{8}\Delta}-e^{-\rho\Delta}}{e^{\frac{3\sigma^2}{8}\Delta}+1-2e^{-\rho\Delta}},\ \delta_1^* = \xi\frac{1-e^{-\rho\Delta}}{e^{\frac{3\sigma^2}{8}\Delta}+1-2e^{-\rho\Delta}}. \tag{72}$$

A direct comparison yields $\delta_0^* > \delta_1^*$ which establishes that the first trade is larger than the last one in the two-period framework, consistently with the numerical observations reported in Section 2.3.1.

Differentiating with respect to volatility σ gives:

$$\frac{\partial\delta_0^*}{\partial\sigma} = \xi\frac{3\sigma}{4}\Delta e^{\frac{3\sigma^2}{8}\Delta}\frac{1-e^{-\rho\Delta}}{\big(e^{\frac{3\sigma^2}{8}\Delta}+1-2e^{-\rho\Delta}\big)^2} > 0, \tag{73}$$

showing that higher volatility increases the size of the initial trade. Differentiation with respect to the resilience parameter ρ gives:

$$\frac{\partial \delta_0^*}{\partial \rho} = \xi \Delta e^{-\rho\Delta} \frac{1 - e^{\frac{3\sigma^2}{8}\Delta}}{\left(e^{\frac{3\sigma^2}{8}\Delta} + 1 - 2e^{-\rho\Delta}\right)^2} < 0, \tag{74}$$

which indicates that faster price recovery of the liquidity pool reduces the size of the initial trade.

Finally, we study the asymptotic behavior of the optimal trades. When volatility becomes large, the solution converges to a fully front-loaded strategy:

$$\lim_{\sigma \to +\infty} \delta_0^* = \xi, \ \lim_{\sigma \to +\infty} \delta_1^* = 0. \tag{75}$$

When the resilience parameter becomes large, we obtain:

$$\lim_{\rho \to +\infty} \delta_0^* = \xi \frac{e^{\frac{3\sigma^2}{8}\Delta}}{e^{\frac{3\sigma^2}{8}\Delta} + 1}, \ \lim_{\rho \to +\infty} \delta_1^* = \xi \frac{1}{e^{\frac{3\sigma^2}{8}\Delta} + 1}. \tag{76}$$

In this regime, the price recovers instantaneously, and the trade allocation reflects only the effects of volatility, with a persistent asymmetry favoring earlier execution under higher expected price impact.

C Backward algorithm

Under the two-layer liquidity framework introduced in Section 3.1, the backward algorithm is provided in the form of pseudocode in Algorithm 1.

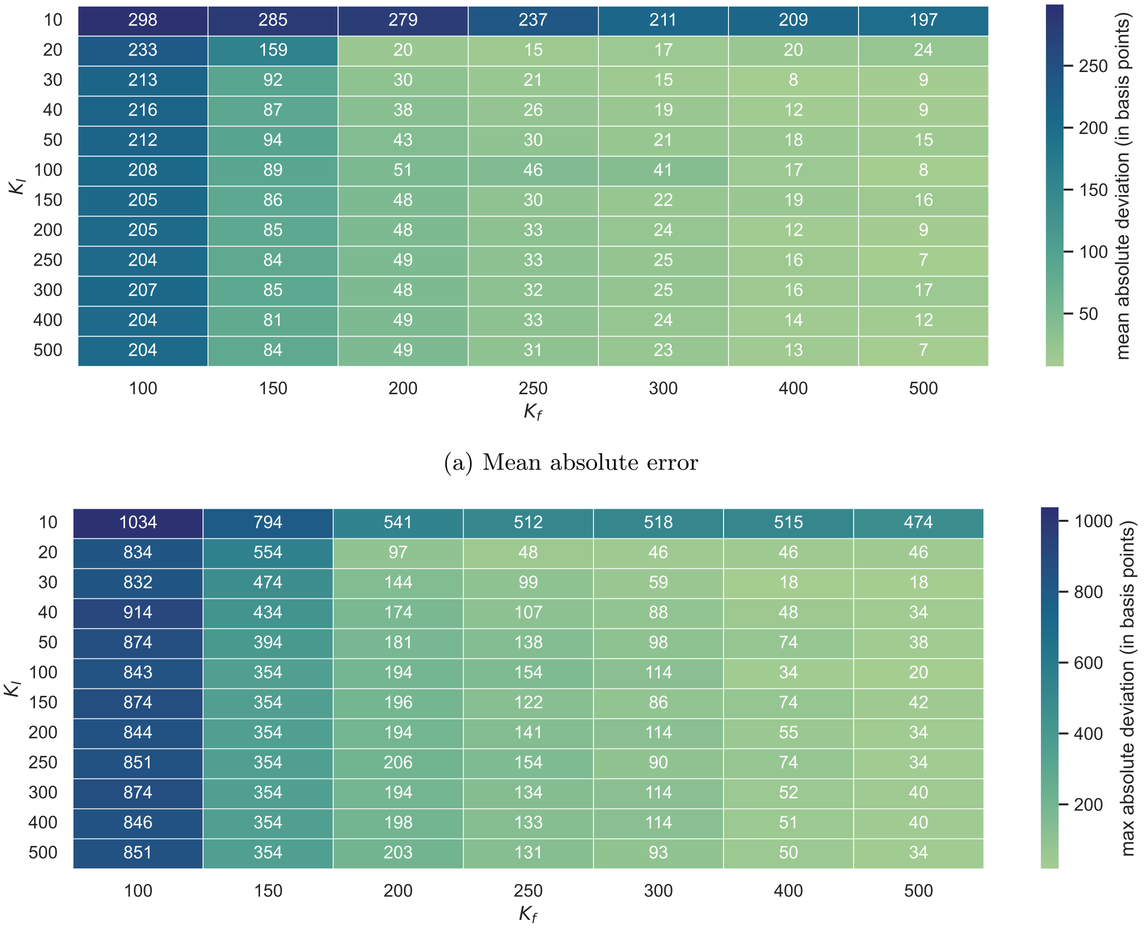
D Consistency and stability checks

First, we present consistency checks conducted within the Uniswap v2 framework. We compare the numerical results obtained using the algorithm introduced in Section 3.2, when the liquidity layers coincide (i.e., $L_0 = L_1$) with the closed-form solution derived in Section 2.2.3, both evaluated along the mean price trajectory. Specifically, we fix $K_x = 250$ and compute the maximum and mean absolute errors (expressed in basis points) for different values of the grid sizes K_f and K_I. Results are reported in Figures 10 and demonstrate that the discretization scheme provides accurate estimates of the optimal strategy in the Uniswap v2 framework. These results also indicate that the accuracy of the numerical solution is more sensitive to the price grid size than to that of the cumulative market impact. Indeed, once K_I exceeds 30, both mean and maximum errors remain nearly unchanged for fixed values of K_f.

Second, we assess the stability of the numerical scheme within the two-layer liquidity framework introduced in Section 3.1 by varying the grid resolution. We fix the threshold level at $\overline{s} = -25$bps and set $K_x = 250$. We compute the maximum and mean absolute deviations (in basis points) relative to a reference solution obtained with the high-resolution configuration ($K_f = 500$, $K_x = 250$, and $K_I = 500$), presented in Section 3.3. The results, reported in Figures 11, show that the deviation consistently decreases as the grid resolution increases. Moreover, as in the Uniswap v2 setting, accuracy is more sensitive to the resolution of the price grid than to that of the cumulative impact grid. Finally, the computational cost of the high-resolution configuration is significant, requiring approximately 45 minutes. However, with a coarser grid of $K_f = 250$, $K_x = 250$, and $K_I = 50$, the computation time drops to 2 minutes, with a maximum deviation of 105bps and a mean deviation of 45bps.

Algorithm 1 Two-layer liquidity backward algorithm

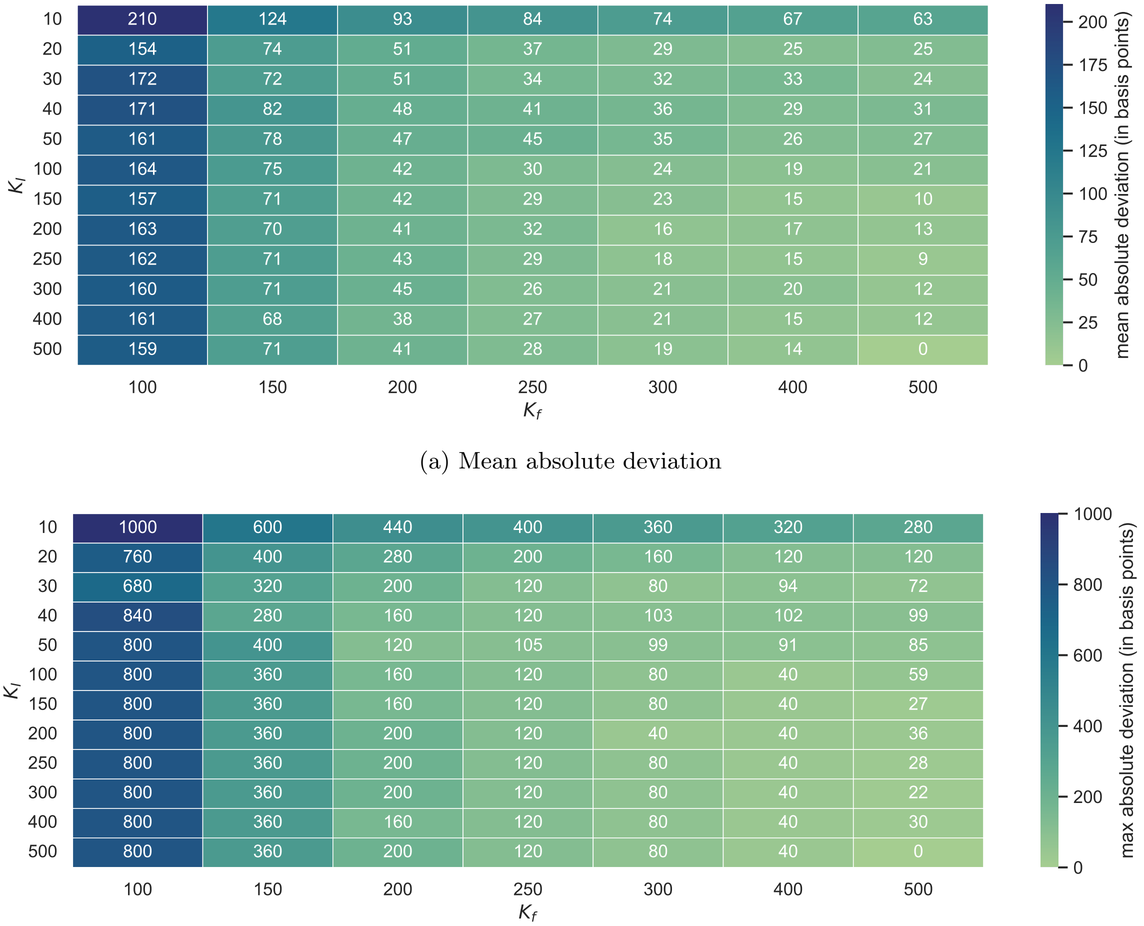
```
Input:
    - The strategy parameters: $T, N$ and $\xi$
    - The market parameters: $f_0, \overline{p}, L_0$ and $L_1$
    - The model parameters: $J, (\rho_j)_{j=0}^J, (\omega_j)_{j=0}^J, \mu$ and $\sigma$
    - The grid parameters: $K_x, K_I, K_f$ and $z$
Output:
    - The estimated optimal execution strategy $\delta^*$
    - The associated value function $v$

  Set the grids according to (45), (46), (47) and (48)
  Set $v_N$ according to (43)                              ▷ The terminal value function is initialized.

  for $n = N-1, \ldots, 0$ do                               ▷ Backward iteration.
      for $k_f = 0, \ldots, K_f$, $k_x = 0, \ldots, K_x$, $k_i^j = 0, \ldots, K_I$, $j = 0, \ldots, J$ do
          for $k = 0, \ldots, k_x$ do
              Set $\delta_n = x^{(k)}$                       ▷ Candidate trade from the inventory grid.

              Compute the resulting cash-flow $\mathcal{C}_n(x^{(k_x)}, (I^{(k_i^j)})_{j=0}^J, f^{(k_f)}, \delta_n)$ from (44)

              Update the cumulative price impacts $I_{n+1}^j(x^{(k_x)}, (I^{(k_i^j)})_{j=0}^J, f^{(k_f)}, \delta_n)$ from (41)

              Update the remaining inventory $x_{n+1} = x^{(k_x)} - \delta_n$

              Perform a $(J+1)$-d linear interpolation to derive $v_{n+1}(x^{(k_x)}, (I^{(k_i^j)})_{j=0}^J, f^{(k_f)})$ for $k_f = 0, \ldots, K_f$

              Estimate the conditional expectation $\overline{v}_{n+1}$ using (49)

              Store the total value $\mathcal{C}_n + \overline{v}_{n+1}$ for each candidate trade $\delta_n$

          end for
          Determine the optimal trade:

              $$\delta_n^*(x^{(k_x)}, (I^{(k_i^j)})_{j=0}^J, f^{(k_f)}) = \underset{\delta_n}{\operatorname{argmax}}\ \mathcal{C}_n + \overline{v}_{n+1}$$

          Update the value function $v_n(x^{(k_x)}, (I^{(k_i^j)})_{j=0}^J, f^{(k_f)})$ accordingly

      end for
  end for
```



(a) Mean absolute error

(b) Maximum absolute error

Figure 10: Mean and maximum absolute error with respect to K_I and K_f for $K_x = 250$ in the Uniswap v2 framework; $T = 1$, $N = 10$, $L = 1000$, $f_0 = 1$, $\mu = 0$, $\sigma = 0.3$, $J = 0$ and $\rho = 3$.



K_I \ K_f	100	150	200	250	300	400	500
10	210	124	93	84	74	67	63
20	154	74	51	37	29	25	25
30	172	72	51	34	32	33	24
40	171	82	48	41	36	29	31
50	161	78	47	45	35	26	27
100	164	75	42	30	24	19	21
150	157	71	42	29	23	15	10
200	163	70	41	32	16	17	13
250	162	71	43	29	18	15	9
300	160	71	45	26	21	20	12
400	161	68	38	27	21	15	12
500	159	71	41	28	19	14	0

(a) Mean absolute deviation

K_I \ K_f	100	150	200	250	300	400	500
10	1000	600	440	400	360	320	280
20	760	400	280	200	160	120	120
30	680	320	200	120	80	94	72
40	840	280	160	120	103	102	99
50	800	400	120	105	99	91	85
100	800	360	160	120	80	40	59
150	800	360	160	120	80	40	27
200	800	360	200	120	40	40	36
250	800	360	200	120	80	40	28
300	800	360	200	120	80	40	22
400	800	360	160	120	80	40	30
500	800	360	200	120	80	40	0

(b) Maximum absolute deviation

Figure 11: Mean and maximum absolute deviation with respect to K_I and K_f for $K_x = 250$ in the two-layer liquidity framework; $T = 1$, $N = 10$, $L_0 = 1000$, $L_1 = 500$, $f_0 = 1$, $\overline{s} = -25\text{bps}$, $\mu = 0$, $\sigma = 0.3$, $J = 0$ and $\rho = 3$.