

# **Routage optimal dans les écosystèmes AMM**

Optimal Routing in AMM-based Ecosystems

Arthus Goujon

Travail Encadré de Recherche — Master 1 de Mathématiques  
Aix-Marseille Université

Encadrant : Sebastian Müller  
`sebastian.muller@univ-amu.fr`

19 juin 2026

## Résumé

**Résumé.** On étudie la modélisation du routage optimal dans les réseaux de Constant Function Market Makers (CFMM). Un exemple introductif présente le fonctionnement d'un Automated Market Maker (AMM) et les notions de prix marginal, de prix effectif, de slippage et d'arbitrage local sur le modèle Uniswap V2. Le routage optimal est ensuite formulé comme un problème d'optimisation sur un réseau de pools. La relaxation des contraintes de trading produit un programme convexe, exact sous des hypothèses de croissance et de saturation. Sa dualité fournit un système de prix d'équilibre, et la détection d'arbitrage en est un cas particulier : l'absence d'arbitrage équivaut à l'existence d'un système de prix global compatible, à un facteur d'échelle près, avec les prix marginaux locaux de chaque pool. Cette condition se lit en termes de cycles : un arbitrage est un cycle de swaps profitable, et l'absence d'arbitrage borne le produit des prix marginaux le long de tout cycle par les frais d'un aller-retour. Une expérience numérique reproductible sur un réseau de trois pools Uniswap V2, à réserves synthétiques réalistes, illustre l'intérêt du fractionnement d'un ordre entre plusieurs routes.

**Abstract.** We study the mathematical modelling of optimal routing in networks of CFMMs. After an introductory example illustrating how an AMM works, together with marginal price, effective price, slippage and local arbitrage on the Uniswap V2 model, optimal routing is cast as an optimisation problem on a network of pools. Relaxing the trading constraints yields a convex program which, under suitable growth and saturation assumptions, is exact. Convex duality extracts an equilibrium price system, and arbitrage detection appears as a special case : the absence of arbitrage is equivalent to the existence of a global price system compatible, up to a scaling factor, with the local marginal prices of every pool. This condition reads in terms of cycles : an arbitrage is a profitable cycle of swaps, and the no-arbitrage condition bounds the product of marginal prices along any cycle by the cost of a round trip. A reproducible numerical experiment on a three-pool Uniswap V2 network, with realistic synthetic reserves, illustrates the practical value of splitting an order across several routes.

# Table des matières

|                                                                 |           |
|-----------------------------------------------------------------|-----------|
| <b>Notations</b>                                                | <b>3</b>  |
| <b>1 Introduction</b>                                           | <b>4</b>  |
| 1.1 Une première illustration du slippage . . . . .             | 4         |
| 1.2 Le cas Uniswap V2 : invariant de produit constant . . . . . | 5         |
| 1.3 Prix marginal, prix effectif et slippage . . . . .          | 6         |
| 1.4 Arbitrage local et efficacité des prix . . . . .            | 7         |
| 1.5 Du pool isolé au réseau de pools . . . . .                  | 7         |
| <b>2 Modélisation mathématique d'un réseau de CFMM</b>          | <b>8</b>  |
| 2.1 Réseau de $m$ CFMM et $n$ tokens . . . . .                  | 8         |
| 2.2 Indexation locale/globale et matrices $A_i$ . . . . .       | 8         |
| 2.3 Paniers fournis et reçus . . . . .                          | 9         |
| 2.4 Fonction de trading et frais . . . . .                      | 9         |
| 2.5 Vecteur de transaction réseau . . . . .                     | 9         |
| 2.6 Fonction d'utilité du trader . . . . .                      | 10        |
| <b>3 Problème de routage optimal</b>                            | <b>11</b> |
| 3.1 Formulation avec contraintes d'égalité . . . . .            | 11        |
| 3.2 Relaxation convexe . . . . .                                | 11        |
| 3.3 Exactitude de la relaxation . . . . .                       | 12        |
| 3.4 Exemple : liquidation d'un token vers un autre . . . . .    | 13        |
| <b>4 Dualité et prix d'équilibre</b>                            | <b>14</b> |
| 4.1 Linéarisation locale des contraintes de trading . . . . .   | 14        |
| 4.2 Lagrangien et dualité faible . . . . .                      | 14        |
| 4.3 Condition de Slater et conditions KKT . . . . .             | 15        |
| 4.4 Interprétation des multiplicateurs duaux . . . . .          | 16        |
| <b>5 Arbitrage et équilibre du marché</b>                       | <b>17</b> |
| 5.1 Définition d'un arbitrage . . . . .                         | 17        |
| 5.2 Détection d'arbitrage par routage optimal . . . . .         | 17        |
| 5.3 Conditions de non-arbitrage . . . . .                       | 17        |
| 5.4 Prix globaux, prix locaux, bande de non-arbitrage . . . . . | 18        |
| 5.5 Cycles d'arbitrage : un exemple élémentaire . . . . .       | 18        |
| 5.6 Arbitrage optimal et taille de transaction . . . . .        | 19        |
| 5.7 Structure de l'ensemble des prix sans arbitrage . . . . .   | 19        |
| <b>6 Application numérique</b>                                  | <b>21</b> |
| 6.1 Données . . . . .                                           | 21        |
| 6.2 Formulation numérique simplifiée . . . . .                  | 21        |
| 6.3 Résultat pour un ordre de 10 WETH . . . . .                 | 21        |
| 6.4 Sensibilité à la taille de l'ordre . . . . .                | 22        |
| 6.5 Résolution du programme convexe général . . . . .           | 22        |
| 6.6 Détection d'arbitrage et certificat de prix . . . . .       | 23        |
| <b>7 Conclusion</b>                                             | <b>24</b> |
| <b>A Reproductibilité de l'expérience numérique</b>             | <b>25</b> |
| <b>B Glossaire minimal des notions AMM</b>                      | <b>27</b> |

## Notations

On rassemble ici les notations principales. Les objets primaux décrivent les transactions ; les objets duaux (prix, multiplicateurs) n'apparaissent qu'à partir de la section 4.

### Notations primales.

| Symbole                                                    | Signification                                                            |
|------------------------------------------------------------|--------------------------------------------------------------------------|
| $m, n$                                                     | nombre de CFMM et nombre de tokens du réseau                             |
| $n_i$                                                      | nombre de tokens du CFMM $i$ , avec $2 \leq n_i \leq n$                  |
| $R_i \in \mathbb{R}_{++}^{n_i}$                            | réserves locales du CFMM $i$                                             |
| $\varphi_i : \mathbb{R}_+^{n_i} \rightarrow \mathbb{R}$    | fonction de trading du CFMM $i$                                          |
| $\gamma_i \in (0, 1]$                                      | paramètre de frais ( $\gamma_i = 0,997$ pour Uniswap V2)                 |
| $A_i \in \mathbb{R}^{n \times n_i}$                        | matrice locale $\rightarrow$ global                                      |
| $\Delta_i, \Lambda_i \in \mathbb{R}_+^{n_i}$               | paniers fourni et reçu dans le CFMM $i$                                  |
| $\Psi \in \mathbb{R}^n$                                    | vecteur de transaction réseau, $\Psi = \sum_i A_i(\Lambda_i - \Delta_i)$ |
| $U : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty\}$ | utilité du trader, concave et croissante                                 |

### Notations duales (sections 4–5).

| Symbole                       | Signification                                                    |
|-------------------------------|------------------------------------------------------------------|
| $P_i = \nabla \varphi_i(R_i)$ | prix marginaux locaux du CFMM $i$ (à un facteur d'échelle près)  |
| $g \in \mathbb{R}_{++}^n$     | vecteur de prix global (variable duale)                          |
| $\lambda_i > 0$               | facteur d'échelle / valeur marginale de la liquidité du pool $i$ |

**Conventions.**  $e_j$  est le  $j$ -ième vecteur de la base canonique ;  $\mathbf{1}$  le vecteur dont toutes les composantes valent 1 ;  $x \leq y$  l'inégalité composante par composante. On note  $\mathbb{R}_+^n = \{x \in \mathbb{R}^n : x \geq 0\}$  et  $\mathbb{R}_{++}^n = \{x \in \mathbb{R}^n : x > 0\}$ . Une fonction  $f$  est croissante composante par composante si  $x \leq y \Rightarrow f(x) \leq f(y)$ .

# 1 Introduction

Les échanges décentralisés (*decentralized exchanges*, DEX) permettent d'échanger des actifs numériques directement sur une blockchain, sans plateforme centralisée. Un token est un tel actif : une unité transférable enregistrée sur la blockchain. L'ether (ETH), ainsi que des stablecoins comme l'USDC ou le DAI, dont la valeur est arrimée au dollar, en sont des exemples.

Une bourse classique apparie acheteurs et vendeurs au moyen d'un carnet d'ordres. Sur une blockchain, ce mécanisme est coûteux : chaque ordre est un enregistrement à stocker, mettre à jour et annuler. Les teneurs de marché automatisés (*Automated Market Makers*, AMM) reposent sur une architecture différente. Un pool de liquidité est un contrat autonome détenant des réserves de deux tokens. Dans ce pool quiconque peut y échanger (*swap*), c'est-à-dire déposer une quantité d'un token contre une quantité de l'autre, sans appariement avec un autre utilisateur. Les réserves sont déposées par des fournisseurs de liquidité, rémunérés par les frais prélevés sur chaque échange.

L'idée centrale est que le prix ne résulte pas de l'appariement d'ordres, mais d'une règle déterministe portant sur les réserves du pool. C'est ce qui fait de la tarification un objet mathématique, analysable par l'optimisation.

On s'intéresse plus précisément aux *Constant Function Market Makers* (CFMM) [3], où les échanges doivent respecter une règle imposée aux réserves, décrite par une fonction de trading. Le cas canonique est Uniswap V2, où le produit des deux réserves reste constant [1, 2].

Lorsqu'un même échange peut emprunter plusieurs pools, une question naturelle se pose : comment répartir les flux entre les routes disponibles pour obtenir la meilleure exécution ? Elle importe surtout lorsque la taille de l'ordre est significative relativement aux réserves, car le prix d'exécution dépend alors de la quantité échangée. On montrera que, sous des hypothèses adaptées sur les fonctions décrivant les pools, ce problème se formule comme un programme convexe [4], ce qui relie aussi le routage à la détection d'opportunités d'arbitrage.

## 1.1 Une première illustration du slippage

Considérons un pool détenant deux actifs, du WETH et de l'USDC, de réserves initiales égales :

$$R_{\text{WETH}} = 1000, \quad R_{\text{USDC}} = 1000.$$

La règle d'Uniswap V2 maintient constant le produit des réserves, ici

$$R_{\text{WETH}} R_{\text{USDC}} = 10^6.$$

Les réserves étant égales, le prix marginal initial est de 1 USDC pour 1 WETH.

Acheteons de l'USDC réglé en WETH. Au prix marginal initial, 3 WETH rapporteraient 3 USDC ; l'invariance impose une autre valeur. Après le versement, la réserve de WETH passe à 1003, et celle d'USDC tombe à  $10^6/1003 = 997,01$  pour conserver le produit. On reçoit donc  $1000 - 997,01 = 2,99$  USDC, soit un écart relatif d'environ 0,3 % au prix marginal : pour un petit ordre, l'écart reste faible.

Pour un gros ordre, 1000 WETH versés en une fois, la réserve de WETH double à 2000 et celle d'USDC tombe à  $10^6/2000 = 500$ . On ne reçoit que 500 USDC au lieu de 1000 : le prix effectif moyen vaut 2 WETH par USDC, le double du prix initial. La table 1 décrit les régimes intermédiaires. On notera toujours  $\Delta$  la quantité donnée et  $\Lambda$  la quantité reçue d'un token.

Plus l'ordre est gros relativement aux réserves, plus le prix se dégrade. On appelle prix effectif le coût unitaire réellement payé,  $P_{\text{eff}}(\Delta) = \Delta/\Lambda$ , et slippage son écart relatif au prix marginal initial. Ici, le prix effectif est explicite,

$$P_{\text{eff}}(\Delta) = \frac{\Delta}{\Lambda_{\text{USDC}}(\Delta)} = 1 + \frac{\Delta}{1000}, \quad (1)$$

TABLE 1 – Achat d’USDC dans le pool (1000 WETH, 1000 USDC), règle du produit constant (frais négligés).

| WETH payés $\Delta$ | USDC reçus $\Lambda$ | Prix effectif (WETH/USDC) | Slippage |
|---------------------|----------------------|---------------------------|----------|
| 3                   | 2,99                 | 1,003                     | 0,3 %    |
| 30                  | 29,13                | 1,030                     | 3,0 %    |
| 100                 | 90,91                | 1,100                     | 10,0 %   |
| 1000                | 500,00               | 2,000                     | 100,0 %  |

une droite : le slippage vaut exactement  $\Delta/1000$ , la taille de l’ordre rapportée à la profondeur du pool (figure 1).

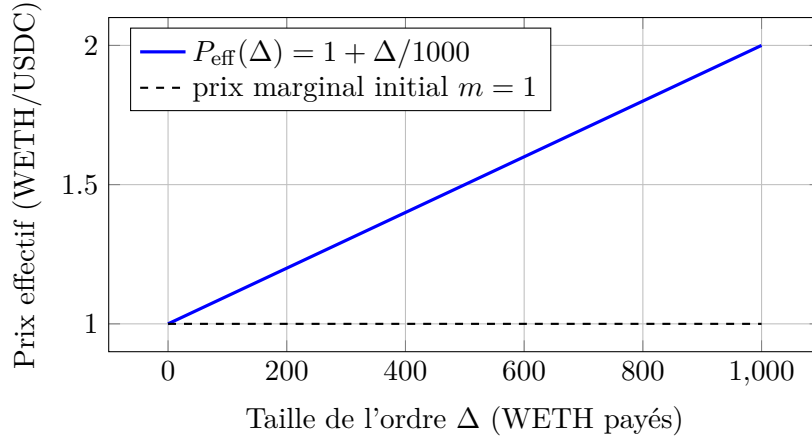


FIGURE 1 – Le prix effectif d’un achat d’USDC croît linéairement avec la taille de l’ordre (produit constant, frais négligés). L’écart à la droite  $m = 1$  est le slippage.

Ce slippage dépend de la taille *relative* de l’ordre, non de sa taille absolue. Comparons deux pools de même prix initial : le pool précédent (1000/1000) et un pool dix fois plus profond (10 000/10 000). Pour un même ordre  $\Delta$ , le second a pour prix effectif

$$P_{\text{eff}}(\Delta) = 1 + \frac{\Delta}{10\,000},$$

soit un slippage dix fois plus faible. Un ordre de 100 WETH subit ainsi 10 % de slippage dans le pool 1000/1000, mais 1 % dans le pool 10 000/10 000. À taille fixée, le slippage est donc d’autant plus faible que les réserves sont grandes : la *liquidité* d’un pool mesure sa capacité à absorber des ordres sans dégrader le prix d’exécution.

Exécuter un gros ordre en une seule transaction est donc coûteux, et ce coût décroît avec la profondeur. D’où le problème central de ce mémoire : fractionner l’ordre sur plusieurs pools pour agréger leur liquidité. Les pools réels prélèvent en outre des frais (0,3 % chez Uniswap V2), intégrés dès la section 1.2 via un facteur  $\gamma = 0,997$  ; ils ajoutent un léger surcoût constant sans modifier le mécanisme qualitatif.

## 1.2 Le cas Uniswap V2 : invariant de produit constant

Soit un pool de deux actifs  $\alpha$  et  $\beta$ , de réserves  $R_\alpha, R_\beta > 0$ , d’invariant  $R_\alpha R_\beta = k$ . Si un utilisateur apporte  $\Delta_\beta > 0$  pour recevoir  $\Lambda_\alpha > 0$ , la contrainte sans frais est

$$(R_\alpha - \Lambda_\alpha)(R_\beta + \Delta_\beta) = R_\alpha R_\beta.$$

Avec frais proportionnels, on note  $\gamma \in (0, 1]$  la fraction du montant entrant prise en compte ; les frais de 0,3 % d'Uniswap V2 correspondent à  $\gamma = 0,997$ . La contrainte devient

$$(R_\alpha - \Lambda_\alpha)(R_\beta + \gamma \Delta_\beta) = R_\alpha R_\beta, \quad \text{d'où} \quad \Lambda_\alpha = \frac{R_\alpha \gamma \Delta_\beta}{R_\beta + \gamma \Delta_\beta}.$$

La sortie n'est donc pas linéaire en la taille de l'ordre [2]. On réécrira cet invariant sous une forme adaptée à l'analyse convexe à la section 2.4.

**Lecture géométrique.** L'invariant  $R_\alpha R_\beta = k$  décrit une hyperbole dans le plan des réserves (figure 2). Un échange déplace l'état du pool le long de cette courbe : apporter  $\Delta_\beta$  et recevoir  $\Lambda_\alpha$  fait croître  $R_\beta$  et décroître  $R_\alpha$ . Le prix marginal est la pente de la tangente, le prix effectif celle de la corde reliant l'état initial à l'état final. La courbe étant convexe, la corde est toujours plus défavorable que la tangente : cet écart est le slippage.

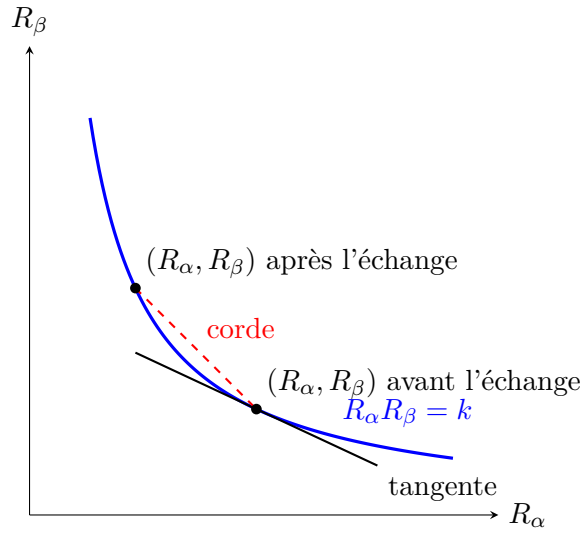


FIGURE 2 – Courbe de réserves  $R_\alpha R_\beta = k$ . Un échange déplace le pool le long de la courbe. La pente de la tangente est le prix marginal, celle de la corde le prix effectif ; leur écart est le slippage, et la convexité garantit que tout échange dégrade le prix.

### 1.3 Prix marginal, prix effectif et slippage

Le prix marginal est le taux d'échange infinitésimal. Sans frais,

$$p_\alpha^\beta = \frac{R_\beta}{R_\alpha} :$$

il faut marginalement  $R_\beta/R_\alpha$  unités de  $\beta$  pour une unité de  $\alpha$ . Pour un ordre fini fournissant  $\Delta_\beta$  et recevant  $\Lambda_\alpha$ , le prix effectif est

$$P_{\text{eff}}(\Delta_\beta) = \frac{\Delta_\beta}{\Lambda_\alpha} = \frac{R_\beta + \gamma \Delta_\beta}{R_\alpha \gamma},$$

croissant en  $\Delta_\beta$ . Le slippage est cette dégradation lorsque la taille de l'ordre devient significative relativement aux réserves, illustrée à la section 1.1.

## 1.4 Arbitrage local et efficacité des prix

Le prix d'un AMM ne dépendant que de ses réserves, il peut s'écarter d'un prix de référence externe. Les arbitrageurs corrigent cet écart en achetant l'actif sous-évalué et vendant l'actif surévalué, ce qui rapproche le prix du pool du prix externe.

Notons  $m_p$  le prix externe de l'actif  $\alpha$  en unités de  $\beta$ , et  $m_u$  le prix marginal du pool dans la même unité. Acheter marginalement  $\alpha$  dans le pool coûte  $m_u/\gamma$  lorsque les frais portent sur le montant entrant ; l'arbitrage « acheter au pool, revendre au marché » est profitable si  $m_p > m_u/\gamma$ , soit  $m_u < \gamma m_p$ . Dans l'autre sens, vendre marginalement  $\alpha$  au pool rapporte  $\gamma m_u$ , profitable si  $\gamma m_u > m_p$ . L'absence d'arbitrage local impose donc

$$\gamma m_p \leq m_u \leq \gamma^{-1} m_p. \quad (2)$$

Les frais créent ainsi une bande de non-arbitrage [2], que l'on retrouvera sous forme vectorielle dans le cadre réseau (section 5.3).

## 1.5 Du pool isolé au réseau de pools

Un écosystème DeFi est un réseau de pools reliant plusieurs actifs. Un échange WETH→DAI peut passer directement par un pool WETH/DAI, ou indirectement via WETH→USDC→DAI, voire être fractionné entre plusieurs routes pour réduire le slippage total. Le problème étudié est donc : étant donné un réseau de CFMM, comment choisir les transactions locales dans chaque pool pour maximiser l'utilité finale du trader ?

## 2 Modélisation mathématique d'un réseau de CFMM

### 2.1 Réseau de $m$ CFMM et $n$ tokens

On considère  $m$  CFMM, indexés par  $i = 1, \dots, m$ , et un ensemble global de  $n$  tokens, indexés par  $j = 1, \dots, n$ . Le CFMM  $i$  ne contient qu'un sous-ensemble de  $n_i$  tokens, avec  $2 \leq n_i \leq n$ .

Cette structure se représente par un graphe biparti : un ensemble de sommets pour les tokens, un autre pour les CFMM, une arête reliant un token à un pool lorsqu'il appartient à ses réserves [4]. Pour l'univers WETH, USDC, DAI, WBTC, on peut considérer les pools

WETH/USDC, WETH/DAI, USDC/DAI, WETH/WBTC.

Un échange WETH→DAI emprunte alors la route directe WETH/DAI ou la route indirecte WETH→USDC→DAI. Décrire simultanément les flux dans plusieurs pools impose de distinguer une indexation locale, propre à chaque pool, d'une indexation globale commune au réseau (figure 3).

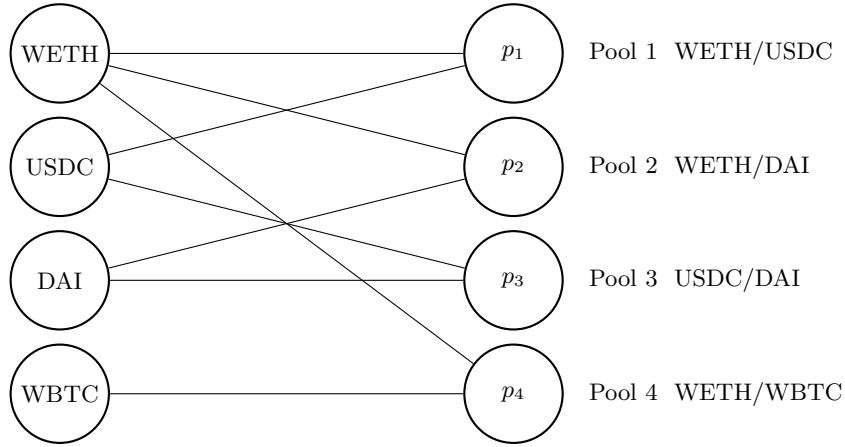


FIGURE 3 – Représentation bipartite d'un réseau de CFMM. Les sommets de gauche sont les tokens, ceux de droite les pools. Une arête indique qu'un token appartient aux réserves du pool.

### 2.2 Indexation locale/globale et matrices $A_i$

L'espace local du CFMM  $i$  est  $\mathbb{R}^{n_i}$ , l'espace global  $\mathbb{R}^n$ . La matrice  $A_i \in \mathbb{R}^{n \times n_i}$  envoie les coordonnées locales du pool  $i$  vers les coordonnées globales : si le  $\ell$ -ième token local du pool  $i$  est le token global  $j(i, \ell)$ , alors

$$A_i e_\ell = e_{j(i, \ell)}.$$

Chaque colonne de  $A_i$  est donc un vecteur de la base canonique de  $\mathbb{R}^n$ , et  $A_i$  est une matrice de sélection plongeant les tokens locaux dans l'espace global.

**Exemple 2.1.** Avec l'indexation globale (WETH, USDC, DAI, WBTC) et l'indexation locale (WETH, DAI) pour le pool WETH/DAI :

$$A_{\text{WETH/DAI}} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad A_{\text{WETH/DAI}} \begin{pmatrix} x_{\text{WETH}} \\ x_{\text{DAI}} \end{pmatrix} = \begin{pmatrix} x_{\text{WETH}} \\ 0 \\ x_{\text{DAI}} \\ 0 \end{pmatrix}.$$

### 2.3 Paniers fournis et reçus

Une transaction locale dans le CFMM  $i$  est décrite par deux vecteurs  $\Delta_i, \Lambda_i \in \mathbb{R}_+^{n_i}$  :  $\Delta_i$  est fourni au pool,  $\Lambda_i$  reçu en retour. Le gain net local  $\Lambda_i - \Delta_i$  s'exprime dans l'espace global par  $A_i(\Lambda_i - \Delta_i)$ .

Par exemple, un swap WETH→DAI dans le pool précédent s'écrit

$$\Delta_i = \begin{pmatrix} \Delta_{\text{WETH}} \\ 0 \end{pmatrix}, \quad \Lambda_i = \begin{pmatrix} 0 \\ \Lambda_{\text{DAI}} \end{pmatrix}, \quad A_i(\Lambda_i - \Delta_i) = \begin{pmatrix} -\Delta_{\text{WETH}} \\ 0 \\ \Lambda_{\text{DAI}} \\ 0 \end{pmatrix} :$$

le trader fournit  $\Delta_{\text{WETH}}$  unités de WETH, reçoit  $\Lambda_{\text{DAI}}$  unités de DAI, et ne modifie pas son bilan en USDC ni en WBTC. Cette notation couvre indifféremment les swaps à deux actifs et les transactions multi-actifs.

### 2.4 Fonction de trading et frais

Le CFMM  $i$  est défini par une fonction de trading  $\varphi_i : \mathbb{R}_+^{n_i} \rightarrow \mathbb{R}$  appliquée aux réserves  $R_i \in \mathbb{R}_+^{n_i}$ . Après une transaction  $(\Delta_i, \Lambda_i)$ , les réserves sont  $R_i + \gamma_i \Delta_i - \Lambda_i$ , où  $\gamma_i \in (0, 1]$  est le paramètre de frais. Une transaction est *admissible* si elle ne diminue pas la fonction de trading :

$$\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i). \quad (3)$$

Dans le cas usuel où le trader obtient le maximum possible en retour, cette contrainte est saturée.

Pour Uniswap V2, l'invariant maintient constant le produit  $R_1 R_2$  ; on le représente par

$$\varphi(R_1, R_2) = \sqrt{R_1 R_2}.$$

La racine étant strictement croissante,  $R_1 R_2$  et  $\sqrt{R_1 R_2}$  ont les mêmes courbes de niveau : les deux décrivent le même invariant. On retient  $\sqrt{R_1 R_2}$  parce qu'elle est concave sur  $\mathbb{R}_+^2$ , contrairement au produit brut ; c'est cette concavité qui rendra convexes les contraintes de sur-niveau (3), donc le problème de routage (section 3).

**Lemme 2.2** (Concavité de l'invariant Uniswap V2). *La fonction  $\varphi(R_1, R_2) = \sqrt{R_1 R_2}$  est concave et croissante composante par composante sur  $\mathbb{R}_+^2$ .*

*Démonstration.* La croissance est immédiate. Sur  $\mathbb{R}_{++}^2$ , la hessienne vaut

$$\nabla^2 \varphi(R_1, R_2) = \frac{1}{4} \begin{pmatrix} -R_1^{-3/2} R_2^{1/2} & R_1^{-1/2} R_2^{-1/2} \\ R_1^{-1/2} R_2^{-1/2} & -R_1^{1/2} R_2^{-3/2} \end{pmatrix},$$

et pour tout  $z = (z_1, z_2)$ ,

$$z^\top \nabla^2 \varphi(R_1, R_2) z = -\frac{1}{4} \left( \frac{R_2^{1/4}}{R_1^{3/4}} z_1 - \frac{R_1^{1/4}}{R_2^{3/4}} z_2 \right)^2 \leq 0.$$

Donc  $\nabla^2 \varphi \preceq 0$  sur  $\mathbb{R}_{++}^2$ . Comme  $\mathbb{R}_{++}^2$  est dense dans  $\mathbb{R}_+^2$  et  $\varphi$  continue, la concavité s'étend au bord par passage à la limite dans  $\varphi(\theta x_n + (1 - \theta)y_n) \geq \theta \varphi(x_n) + (1 - \theta)\varphi(y_n)$ .  $\square$

### 2.5 Vecteur de transaction réseau

Le vecteur de transaction réseau est

$$\Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i) \in \mathbb{R}^n.$$

Sa composante  $\Psi_j$  est la variation nette du token  $j$  dans le portefeuille du trader :  $\Psi_j > 0$  une réception,  $\Psi_j < 0$  une fourniture,  $\Psi_j = 0$  une position nette nulle. Une liquidation pure de WETH vers DAI s'impose par  $\Psi = (-\Delta_{\text{WETH}}, 0, \Lambda_{\text{DAI}}, 0)^\top$  ; la nullité de la composante USDC n'est pas automatique, c'est une contrainte de bilan choisie par le trader (USDC comme actif purement intermédiaire).

## 2.6 Fonction d'utilité du trader

La fonction d'utilité  $U$  attribue une valeur au bilan final  $\Psi$ . Elle n'est pas imposée par le protocole mais choisie par le trader, ou le modélisateur ; on la suppose  $U : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty\}$  concave et croissante composante par composante, et le trader choisit  $(\Delta_i, \Lambda_i)$  de manière à maximiser  $U(\Psi)$ .

Le choix de  $U$  dépend de l'objectif. Pour liquider le token  $a$  vers le token  $b$  (fournir une quantité fixée de  $a$ , recevoir le maximum de  $b$ ), on prend

$$U(\Psi) = e_b^\top \Psi, \quad e_a^\top \Psi = -q,$$

où  $q > 0$  est la quantité fournie. Pour une valorisation externe selon un vecteur de prix  $\pi \in \mathbb{R}_+^n$ , on prend  $U(\Psi) = \pi^\top \Psi$  ; pour la détection d'arbitrage, un objectif encore différent (section 5.2). Le cadre général d'une utilité concave croissante est nécessaire à la dualité et aux prix d'équilibre (section 4) : on ne se restreint donc pas au cas de liquidation.

**Remarque 2.3.** L'utilité de liquidation  $U(\Psi) = e_b^\top \Psi$  ne dépend que de la composante  $b$  : elle est croissante, mais pas *strictement*. Cette distinction, sans conséquence ici, demandera une précaution pour l'exactitude de la relaxation (section 3.3).

### 3 Problème de routage optimal

**Un flot non linéaire.** Le routage se lit comme un problème de flot : acheminer un token-source vers un token-cible à travers des arêtes (les pools), en répartissant le flux. Contrairement au plus court chemin classique, le coût d'une arête dépend de la quantité échangée. Me rendement d'une arête décroît avec le flux, c'est la concavité de  $\varphi_i$ , c'est-à-dire le slippage. C'est cette décroissance qui rend le fractionnement utile : au-delà d'un certain flux, une route supplémentaire devient plus rentable qu'un surcroît de flux sur la première. La concavité permet de reformuler le tout dans un cadre convexe.

On travaille dans le cadre général des CFMM pour faire apparaître la structure convexe ; le cas Uniswap V2, défini par  $\varphi(R_1, R_2) = \sqrt{R_1 R_2}$ , en est l'exemple principal et celui de l'expérience numérique.

**Hypothèses 3.1** (Hypothèses de base). Pour tout CFMM  $i$  :

- $\varphi_i : \mathbb{R}_+^{n_i} \rightarrow \mathbb{R}$  est continue, concave et croissante composante par composante ;
- $R_i \in \mathbb{R}_{++}^{n_i}$ , et les réserves post-transaction vérifient  $R_i + \gamma_i \Delta_i - \Lambda_i \in \mathbb{R}_+^{n_i}$  ;
- $U : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty\}$  est concave et croissante composante par composante.

**Hypothèses 3.2** (Hypothèses supplémentaires). Selon les résultats, on suppose en outre :

- (a)  $\varphi_i$  différentiable en  $R_i$ , avec  $\nabla \varphi_i(R_i) \in \mathbb{R}_{++}^{n_i}$ , pour les résultats faisant intervenir les prix marginaux ;
- (b)  $U$  strictement croissante, ou la sélection d'une solution saturant les contraintes, pour l'exactitude stricte de la relaxation ;
- (c) une qualification des contraintes, pour la dualité.

La convexité du problème relaxé (proposition 3.3) ne requiert que la concavité et la croissance des  $\varphi_i$  ; la différentiabilité n'intervient que pour les prix marginaux.

#### 3.1 Formulation avec contraintes d'égalité

Les variables sont  $\Delta_i, \Lambda_i \in \mathbb{R}_+^{n_i}$  et le vecteur auxiliaire  $\Psi \in \mathbb{R}^n$ , que l'on conserve plutôt que de le substituer afin d'exprimer directement les contraintes globales. Le routage s'écrit

$$\begin{aligned}
 & \max_{\Delta, \Lambda, \Psi} U(\Psi) \\
 \text{s.c.} \quad & \Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i), \\
 & \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) = \varphi_i(R_i), \quad i = 1, \dots, m, \\
 & R_i + \gamma_i \Delta_i - \Lambda_i \geq 0, \quad i = 1, \dots, m, \\
 & \Delta_i \geq 0, \quad \Lambda_i \geq 0, \quad i = 1, \dots, m.
 \end{aligned} \tag{4}$$

Ce problème n'est pas convexe en général : les égalités non linéaires définissent rarement des ensembles convexes. Pour un pool Uniswap V2 à deux tokens, en notant  $\tilde{R}_i = R_i + \gamma_i \Delta_i - \Lambda_i$  les réserves post-transaction, la contrainte  $\sqrt{\tilde{R}_{i,1} \tilde{R}_{i,2}} = \sqrt{R_{i,1} R_{i,2}}$  équivaut (réserves positives) à

$$(R_{i,1} + \gamma_i \Delta_{i,1} - \Lambda_{i,1})(R_{i,2} + \gamma_i \Delta_{i,2} - \Lambda_{i,2}) = R_{i,1} R_{i,2},$$

soit l'invariant de produit constant de la section 1.2, écrit pour les réserves post-transaction.

#### 3.2 Relaxation convexe

On relâche chaque égalité en l'inégalité de sur-niveau correspondante : une transaction est admise dès qu'elle ne diminue pas la fonction de trading. Comme  $\varphi_i$  est concave, son sur-niveau

est convexe, ce qui rend le problème convexe et donc justiciable de la dualité et des solveurs d'optimisation convexe :

$$\begin{aligned}
& \max_{\Delta, \Lambda, \Psi} U(\Psi) \\
& \text{s.c.} \quad \Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i), \\
& \quad \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i), \quad i = 1, \dots, m, \\
& \quad R_i + \gamma_i \Delta_i - \Lambda_i \geq 0, \quad i = 1, \dots, m, \\
& \quad \Delta_i \geq 0, \quad \Lambda_i \geq 0, \quad i = 1, \dots, m.
\end{aligned} \tag{5}$$

Dans la convention de maximisation adoptée, un *programme convexe* maximise une fonction concave sur un ensemble convexe c'est l'équivalent, au signe près, de la minimisation d'une fonction convexe.

**Proposition 3.3** (Convexité du problème relaxé). *Sous les hypothèses 3.1, le problème (5) est un programme convexe.*

*Démonstration.* La contrainte de bilan et les contraintes  $\Delta_i \geq 0$ ,  $\Lambda_i \geq 0$ ,  $R_i + \gamma_i \Delta_i - \Lambda_i \geq 0$  sont affines. L'application  $(\Delta_i, \Lambda_i) \mapsto R_i + \gamma_i \Delta_i - \Lambda_i$  étant affine et  $\varphi_i$  concave, la composée  $(\Delta_i, \Lambda_i) \mapsto \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i)$  est concave, de sur-niveau convexe [5]. L'intersection est convexe, et l'on y maximise la fonction concave  $U$ .  $\square$

**Remarque 3.4.** La transaction nulle  $\Delta_i = \Lambda_i = 0$  est toujours admissible. Avec des contraintes de liquidation  $e_a^\top \Psi = -q$ , la faisabilité dépend de la connectivité, des réserves et de la taille de l'ordre.

**Remarque 3.5** (Existence d'un optimum). Les résultats d'optimalité qui suivent supposent la valeur optimale finie et atteinte, ce qui évite des difficultés topologiques périphériques. C'est naturel pour les liquidations à taille d'ordre fixée : les quantités reçues sont bornées par les réserves, et l'expérience de la section 6 se ramène à l'optimisation d'une fonction continue sur un segment compact. Dans les énoncés généraux, l'existence d'une solution optimale doit donc se lire comme une hypothèse technique.

### 3.3 Exactitude de la relaxation

La relaxation (5) n'est pertinente que si ses solutions résolvent aussi (4), c'est-à-dire saturent les contraintes de trading.

**Proposition 3.6** (Exactitude de la relaxation). *Sous les hypothèses 3.1, avec  $U$  strictement croissante composante par composante et les réserves post-transaction strictement positives à l'optimum, toute solution optimale de (5) sature les contraintes de trading :*

$$\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) = \varphi_i(R_i), \quad i = 1, \dots, m.$$

*Démonstration.* Soit  $(\Delta_i, \Lambda_i, \Psi)_i$  optimale, et supposons par l'absurde  $\varphi_k(\tilde{R}_k) > \varphi_k(R_k)$  pour un pool  $k$ , où  $\tilde{R}_k = R_k + \gamma_k \Delta_k - \Lambda_k \in \mathbb{R}_{++}^{n_k}$ . Fixons  $\ell$ . Comme  $\tilde{R}_k > 0$ , il existe  $\eta > 0$  tel que  $\tilde{R}_k - \varepsilon e_\ell \in \mathbb{R}_+^{n_k}$  pour  $\varepsilon \in [0, \eta]$  ; par continuité et stricte inégalité en  $\varepsilon = 0$ , on a  $\varphi_k(\tilde{R}_k - \varepsilon e_\ell) \geq \varphi_k(R_k)$  pour un  $0 < \varepsilon \leq \eta$ . En posant  $\hat{\Lambda}_k = \Lambda_k + \varepsilon e_\ell$  (autres variables inchangées), la contrainte du pool  $k$  reste vérifiée et  $\hat{\Psi} = \Psi + \varepsilon A_k e_\ell \geq \Psi$ ,  $\hat{\Psi} \neq \Psi$ , car  $A_k e_\ell$  est canonique. Comme  $U$  est strictement croissante,  $U(\hat{\Psi}) > U(\Psi)$ , contredisant l'optimalité.  $\square$

**Remarque 3.7** (Utilité non strictement croissante). La proposition 3.6 exclut l'utilité de liquidation  $U(\Psi) = e_b^\top \Psi$ , croissante mais non strictement. L'exactitude n'est alors plus garantie pour

*toute* solution : un optimum peut laisser une contrainte non saturée, l'amélioration correspondante n'augmentant pas  $U$ . On peut néanmoins toujours *sélectionner* une solution saturée. Pour  $\varepsilon > 0$ , on perturbe l'objectif en

$$U_\varepsilon(\Psi) = U(\Psi) + \varepsilon \mathbf{1}^\top \Psi,$$

strictement croissant, auquel s'applique la proposition 3.6. Pour les liquidations considérées ( $e_a^\top \Psi = -q$ ), les quantités reçues sont bornées par les réserves, donc la valeur optimale est atteinte (remarque 3.5) et chaque problème perturbé admet une solution saturée. Quand  $\varepsilon \rightarrow 0$ , toute valeur d'adhérence de ces solutions reste admissible, sature encore les contraintes par continuité des  $\varphi_i$ , et est optimale pour  $U$ . Hors de ce cadre (utilité quelconque, réseau général), une telle sélection requiert une hypothèse de compacité ou de coercivité que l'on ne suppose pas ici.

### 3.4 Exemple : liquidation d'un token vers un autre

Pour  $a$  le token d'entrée,  $b$  le token cible et  $q > 0$  la quantité fournie, on résout

$$\begin{aligned} \max \quad & e_b^\top \Psi \\ \text{s.c.} \quad & \Psi = \sum_i A_i(\Lambda_i - \Delta_i), \quad \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i), \\ & R_i + \gamma_i \Delta_i - \Lambda_i \geq 0, \quad e_a^\top \Psi = -q, \quad \Psi_j \geq 0 \ (j \neq a, b), \\ & \Delta_i \geq 0, \ \Lambda_i \geq 0. \end{aligned} \tag{6}$$

Les variables étant définies simultanément pour tous les pools, le programme n'impose pas une route unique : il peut envoyer une part du flux par le pool direct WETH/DAI, une autre par WETH→USDC→DAI, et retenir la répartition maximisant la quantité finale de DAI.

## 4 Dualité et prix d'équilibre

Le problème de routage relaxé étant convexe, on en étudie la dualité lagrangienne. Les multiplicateurs y reçoivent une interprétation naturelle : celui de la contrainte de bilan global joue le rôle d'un système de prix des tokens, ceux des contraintes de trading mesurent la valeur marginale de la liquidité de chaque pool. Cette lecture duale fournira ensuite la caractérisation de l'absence d'arbitrage.

On suit la dualité convexe classique [5], et l'on suppose les  $\varphi_i$  différentiables aux points considérés. On note

$$P_i = \nabla \varphi_i(R_i) \in \mathbb{R}_{++}^{n_i}$$

les prix marginaux locaux du pool  $i$  au point initial  $R_i$ . Pour Uniswap V2,

$$\nabla \varphi(R_1, R_2) = \frac{1}{2} \begin{pmatrix} \sqrt{R_2/R_1} \\ \sqrt{R_1/R_2} \end{pmatrix},$$

dont le rapport des composantes vaut  $R_2/R_1$ , le prix marginal usuel du premier token dans le second.

### 4.1 Linéarisation locale des contraintes de trading

La contrainte relâchée du pool  $i$ ,  $\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i)$ , est non linéaire ; la concavité de  $\varphi_i$  en donne une condition linéaire nécessaire, exprimant qu'au prix marginal local, la valeur fournie après frais vaut au moins la valeur reçue.

**Proposition 4.1** (Inclusion au premier ordre). *Sous les hypothèses 3.1, avec  $\varphi_i$  différentiable en  $R_i$ , toute transaction admissible  $(\Delta_i, \Lambda_i)_{i=1}^m$  vérifie  $\gamma_i P_i^\top \Delta_i \geq P_i^\top \Lambda_i$  pour chaque pool  $i$ . En particulier, l'ensemble des bilans atteignables est contenu dans le cône convexe*

$$\mathcal{C} = \left\{ \sum_{i=1}^m A_i(\Lambda_i - \Delta_i) : \Delta_i, \Lambda_i \geq 0, \gamma_i P_i^\top \Delta_i \geq P_i^\top \Lambda_i \text{ pour tout } i \right\}.$$

*Démonstration.* Pour une transaction admissible, la concavité et la différentiabilité de  $\varphi_i$  donnent l'inégalité de sur-gradient

$$\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \leq \varphi_i(R_i) + P_i^\top (\gamma_i \Delta_i - \Lambda_i),$$

qui, combinée à l'admissibilité, fournit  $P_i^\top (\gamma_i \Delta_i - \Lambda_i) \geq 0$ , soit  $\gamma_i P_i^\top \Delta_i \geq P_i^\top \Lambda_i$ . Le bilan global  $\Psi = \sum_i A_i(\Lambda_i - \Delta_i)$  appartient donc à  $\mathcal{C}$ . Enfin  $\mathcal{C}$  est un cône convexe : pour  $t \geq 0$ , les variables  $(t\Delta_i, t\Lambda_i)$  vérifient encore les contraintes et donnent le bilan  $t\Psi$  ; et  $\mathcal{C}$  est l'image par une application linéaire d'un ensemble défini par des inégalités linéaires.  $\square$

L'ensemble exact des bilans atteignables n'est pas un cône, les réserves étant finies :  $\mathcal{C}$  est un objet local, décrivant les bilans compatibles avec les contraintes linéarisées aux prix marginaux initiaux. Il servira à relier prix marginaux et absence d'arbitrage.

### 4.2 Lagrangien et dualité faible

On pose

$$f_i(\Delta_i, \Lambda_i) = \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) - \varphi_i(R_i),$$

de sorte que la contrainte de trading du pool  $i$  s'écrit  $f_i \geq 0$ . Le problème primal relaxé est

$$\begin{aligned} p^\star &= \sup_{\Delta, \Lambda, \Psi} U(\Psi) \\ \text{s.c.} \quad &\Psi = \sum_{i=1}^m A_i(\Lambda_i - \Delta_i), \quad f_i(\Delta_i, \Lambda_i) \geq 0, \\ &R_i + \gamma_i \Delta_i - \Lambda_i \geq 0, \quad \Delta_i \geq 0, \Lambda_i \geq 0. \end{aligned}$$

On dualise la contrainte de bilan, par un multiplicateur  $g \in \mathbb{R}^n$ , et les contraintes de trading, par  $\lambda_i \geq 0$ ; les contraintes de positivité restent dans le domaine. Le Lagrangien est

$$L(\Delta, \Lambda, \Psi, g, \lambda) = U(\Psi) - g^\top \left( \Psi - \sum_{i=1}^m A_i(\Lambda_i - \Delta_i) \right) + \sum_{i=1}^m \lambda_i (\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) - \varphi_i(R_i)).$$

Le signe des  $\lambda_i$  est cohérent avec la convention de maximisation et l'écriture  $f_i \geq 0$ . La fonction duale est  $h(g, \lambda) = \sup_{\Delta, \Lambda, \Psi} L$  (supremum sur le domaine de positivité), et le dual  $d^* = \inf_{g, \lambda \geq 0} h(g, \lambda)$ .

**Proposition 4.2** (Dualité faible). *Pour tout  $g \in \mathbb{R}^n$  et tout  $\lambda \geq 0$ ,  $p^* \leq h(g, \lambda)$ ; en particulier  $p^* \leq d^*$ .*

*Démonstration.* Pour une solution primale admissible, le terme en  $g$  s'annule (contrainte de bilan), et  $\sum_i \lambda_i f_i \geq 0$  car  $\lambda_i \geq 0$  et  $f_i \geq 0$ . Donc  $L \geq U(\Psi)$ , puis  $h(g, \lambda) \geq L \geq U(\Psi)$ . En prenant le supremum sur les solutions primales,  $p^* \leq h(g, \lambda)$ , et en minimisant sur  $g, \lambda \geq 0$ ,  $p^* \leq d^*$ .  $\square$

La dualité faible ne donne qu'une borne supérieure. Pour l'atteindre et caractériser l'optimum par les multiplicateurs, on invoque une condition de régularité.

### 4.3 Condition de Slater et conditions KKT

La condition de Slater demande un point admissible satisfaisant strictement les contraintes d'inégalité non affines. Vérifiée, elle entraîne la dualité forte  $p^* = d^*$ , et les conditions de Karush–Kuhn–Tucker sont alors nécessaires et suffisantes [5, §5.2.3]. Ici, les conditions KKT comprennent la faisabilité primale, la faisabilité duale  $\lambda_i \geq 0$ , la complémentarité  $\lambda_i f_i = 0$  (une contrainte non saturée a un multiplicateur nul), et la stationnarité du Lagrangien sous les contraintes de positivité.

**Proposition 4.3** (Conditions d'optimalité du problème relaxé). *Supposons  $U$  et les  $\varphi_i$  différentiables aux points considérés, les  $\varphi_i$  strictement croissantes au voisinage des réserves, et la valeur optimale finie et atteinte avec réserves post-transaction strictement positives. Alors la condition de Slater est vérifiée, la dualité forte a lieu, et les conditions KKT caractérisent l'optimum. Pour une solution optimale  $(\Delta_i, \Lambda_i, \Psi)_i$ , en posant*

$$\tilde{R}_i = R_i + \gamma_i \Delta_i - \Lambda_i, \quad \tilde{P}_i = \nabla \varphi_i(\tilde{R}_i),$$

*il existe  $g \in \mathbb{R}^n$  et  $\lambda_i \geq 0$  tels que  $g = \nabla U(\Psi)$  et, pour tout  $i$ ,*

$$\gamma_i \lambda_i \tilde{P}_i \leq A_i^\top g \leq \lambda_i \tilde{P}_i, \quad \lambda_i (\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) - \varphi_i(R_i)) = 0.$$

*Démonstration. Slater.* Les seules contraintes non affines sont les contraintes de trading. Pour chaque pool, choisissons  $\Delta_i = \delta \mathbf{1}$ ,  $\Lambda_i = \delta' \mathbf{1}$  avec  $0 < \delta' < \gamma_i \delta$  et  $\delta > 0$  petit. Alors  $R_i + \gamma_i \Delta_i - \Lambda_i = R_i + (\gamma_i \delta - \delta') \mathbf{1} > R_i$  composante par composante, d'où  $\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) > \varphi_i(R_i)$  par stricte croissance;  $\delta$  petit conserve les réserves positives, et la contrainte affine détermine  $\Psi$ . Slater est vérifiée, donc dualité forte et KKT nécessaires et suffisantes.

*Stationnarité en  $\Psi$ .* Les termes en  $\Psi$  sont  $U(\Psi) - g^\top \Psi$ , d'où  $\nabla U(\Psi) = g$ .

*Stationnarité en  $\Delta_i, \Lambda_i$ .* Avec  $\tilde{P}_i = \nabla \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i)$ ,

$$\nabla_{\Delta_i} L = -A_i^\top g + \gamma_i \lambda_i \tilde{P}_i, \quad \nabla_{\Lambda_i} L = A_i^\top g - \lambda_i \tilde{P}_i.$$

Les variables étant contraintes positives, la stationnarité d'une maximisation concave sur le cône positif impose  $\nabla_{\Delta_i} L \leq 0$  et  $\nabla_{\Lambda_i} L \leq 0$  (égalité sur les composantes strictement positives), soit

$$\gamma_i \lambda_i \tilde{P}_i \leq A_i^\top g \leq \lambda_i \tilde{P}_i.$$

*Complémentarité.* La contrainte de trading du pool  $i$  donne  $\lambda_i (\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) - \varphi_i(R_i)) = 0$ .  $\square$

#### 4.4 Interprétation des multiplicateurs duaux

L'égalité  $g = \nabla U(\Psi)$  identifie  $g$  aux valeurs marginales de l'objectif à l'optimum :  $g_j$  est l'effet marginal d'une unité du token  $j$  dans le bilan final. Pour une utilité linéaire  $U(\Psi) = \pi^\top \Psi$ , on a simplement  $g = \pi$  ; en général,  $g$  est un système de prix implicites endogène.

La double inégalité  $\gamma_i \lambda_i \tilde{P}_i \leq A_i^\top g \leq \lambda_i \tilde{P}_i$  exprime la compatibilité entre prix globaux et prix locaux :  $A_i^\top g$  est la restriction du système global aux tokens du pool  $i$ ,  $\tilde{P}_i$  ses prix marginaux post-transaction, et  $\lambda_i$  ajuste l'échelle, le gradient ne déterminant les prix marginaux qu'à un facteur multiplicatif près. Sans frais ( $\gamma_i = 1$ ), cela se réduit à  $A_i^\top g = \lambda_i \tilde{P}_i$  : prix globaux et prix locaux coïncident à l'échelle  $\lambda_i$  près. Avec frais, l'égalité devient une bande, et les frais autorisent un écart sans incohérence exploitable.

Enfin, la complémentarité montre que  $\lambda_i$ , multiplicateur de la contrainte de trading, n'est strictement positif que lorsque cette contrainte est active : c'est la valeur marginale de la liquidité du pool.

Cette lecture prépare la caractérisation de l'absence d'arbitrage. À la section suivante, on applique une condition analogue aux prix marginaux *initiaux* : un réseau est sans arbitrage lorsqu'il existe  $g \in \mathbb{R}_{++}^n$  et  $\lambda_i > 0$  tels que

$$\gamma_i \lambda_i P_i \leq A_i^\top g \leq \lambda_i P_i, \quad i = 1, \dots, m,$$

c'est-à-dire lorsque les prix marginaux locaux de tous les pools se rendent compatibles avec un même système de prix global, à l'intérieur des bandes de frais.

## 5 Arbitrage et équilibre du marché

### 5.1 Définition d'un arbitrage

**Définition 5.1.** Une transaction réseau admissible est un arbitrage si son bilan  $\Psi$  vérifie  $\Psi \geq 0$  et  $\Psi \neq 0$ .

Le trader ne fournit aucun token net et reçoit une quantité strictement positive d'au moins un token. Cette définition, indépendante d'un numéraire, suit [4].

**Remarque 5.2.** Elle est plus forte qu'une profitabilité relative à un prix externe  $g \in \mathbb{R}_{++}^n$ , qui ne demanderait que  $g^\top \Psi > 0$  : on impose ici un gain net non négatif dans chaque token.

### 5.2 Détection d'arbitrage par routage optimal

On prend  $U(\Psi) = \mathbf{1}^\top \Psi$  et l'on impose  $\Psi \geq 0$  :

$$\begin{aligned} \max \quad & \mathbf{1}^\top \Psi \\ \text{s.c.} \quad & \Psi = \sum_i A_i(\Lambda_i - \Delta_i), \quad \varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \varphi_i(R_i), \\ & R_i + \gamma_i \Delta_i - \Lambda_i \geq 0, \quad \Psi \geq 0, \quad \Delta_i \geq 0, \quad \Lambda_i \geq 0. \end{aligned} \quad (7)$$

L'objectif  $\mathbf{1}^\top \Psi$  n'est pas une valorisation (additionner des tokens distincts n'aurait pas de sens financier) ; comme on impose déjà  $\Psi \geq 0$ , la condition  $\mathbf{1}^\top \Psi > 0$  dit simplement qu'au moins une composante est strictement positive. Il sert donc de *détecteur* d'un bilan positif non nul. La solution nulle est admissible, de valeur 0. Si l'optimum vérifie  $\mathbf{1}^\top \Psi^* > 0$ , alors  $\Psi^* \geq 0$ ,  $\Psi^* \neq 0$  : il y a arbitrage. Réciproquement, tout arbitrage donne une valeur strictement positive pour toute utilité strictement croissante. Le réseau est donc sans arbitrage si et seulement si la valeur optimale de (7) est nulle, c'est-à-dire si la transaction nulle est optimale.

### 5.3 Conditions de non-arbitrage

Pour chaque CFMM  $i$ , on suppose  $\varphi_i$  différentiable en  $R_i$  et l'on note  $P_i = \nabla \varphi_i(R_i) \in \mathbb{R}_{++}^{n_i}$  ;  $A_i^\top g$  est la restriction des prix globaux aux tokens du pool  $i$ .

**Théorème 5.3** (Caractérisation du non-arbitrage [4]). *Sous les hypothèses 3.1 avec  $\varphi_i$  différentiable en  $R_i$  et  $P_i > 0$ , le réseau ne contient pas d'arbitrage si et seulement s'il existe  $g \in \mathbb{R}_{++}^n$  et des scalaires  $\lambda_i > 0$  tels que*

$$\gamma_i \lambda_i P_i \leq A_i^\top g \leq \lambda_i P_i, \quad i = 1, \dots, m. \quad (8)$$

*Preuve. Sens suffisant.*  $g \in \mathbb{R}_{++}^n$  et  $\lambda_i > 0$  vérifiant (8), et  $(\Delta_i, \Lambda_i)_i$  admissible de bilan  $\Psi$ . L'inégalité de sur-gradient donne, comme à la proposition 4.1,

$$\gamma_i P_i^\top \Delta_i \geq P_i^\top \Lambda_i. \quad (9)$$

Avec  $\Lambda_i, \Delta_i \geq 0$ , les bornes  $A_i^\top g \leq \lambda_i P_i$  et  $A_i^\top g \geq \gamma_i \lambda_i P_i$  donnent

$$g^\top A_i(\Lambda_i - \Delta_i) \leq \lambda_i P_i^\top \Lambda_i - \gamma_i \lambda_i P_i^\top \Delta_i = \lambda_i (P_i^\top \Lambda_i - \gamma_i P_i^\top \Delta_i) \leq 0$$

par (9). En sommant,  $g^\top \Psi \leq 0$ . Si  $\Psi$  était un arbitrage ( $\Psi \geq 0$ ,  $\Psi \neq 0$ ), on aurait  $g^\top \Psi > 0$  puisque  $g > 0$  : contradiction.

*Sens nécessaire.* Le sens suffisant suffit à comprendre la bande. Le sens nécessaire est plus technique et repose sur le théorème de dualité (ou de séparation) de Angeris et al. [4], dont on donne seulement l'idée. Le couple  $(g, \lambda)$  est constitué des variables duales du routage, et la bande (8)

en est la condition de faisabilité duale (proposition 4.3) : à un état sans arbitrage, la transaction nulle est optimale pour (7), les prix marginaux post-transaction valent  $\tilde{P}_i = \nabla \varphi_i(R_i) = P_i$ , et la proposition 4.3 fournit la bande. La difficulté tient à ce que l'optimum est la transaction nulle, où la contrainte  $\Psi \geq 0$  devient active au bord et peut faire échouer Slater au sens strict ; l'existence des multiplicateurs dans ce cas dégénéré est établie dans [4]. La stricte positivité de  $g$  traduit l'existence d'un prix de référence pour tous les tokens, et fait intervenir la connectivité du réseau.  $\square$

**Remarque 5.4.** Le sens suffisant exploite l'inclusion de la proposition 4.1 : toute transaction admissible vit dans  $\mathcal{C}$ , où la bande (8) contrôle le signe de  $g^\top \Psi$ . Sans frais, (8) se réduit à  $A_i^\top g = \lambda_i P_i$ . Pour Uniswap V2,  $\nabla \varphi(R_1, R_2) = \frac{1}{2}(\sqrt{R_2/R_1}, \sqrt{R_1/R_2})$ , et le rapport des composantes donne  $R_2/R_1$ , le prix marginal de la section 1.3.

## 5.4 Prix globaux, prix locaux, bande de non-arbitrage

Dans un pool isolé, la bande s'écrit  $\gamma m_p \leq m_u \leq \gamma^{-1} m_p$  (équation (2)). Dans un réseau,  $g$  joue le rôle de  $m_p$ , et la condition  $\gamma_i \lambda_i P_i \leq A_i^\top g \leq \lambda_i P_i$  doit valoir simultanément pour tous les pools : chacun impose une contrainte locale, mais toutes doivent être compatibles avec un même  $g$ . L'absence d'arbitrage à l'échelle du réseau n'est donc pas la juxtaposition d'absences locales, mais une cohérence globale du système de prix. Un arbitrage apparaît lorsque les prix locaux ne se laissent plus représenter par un système global cohérent.

## 5.5 Cycles d'arbitrage : un exemple élémentaire

On reprend les tokens WETH, USDC, DAI et les pools WETH/USDC, USDC/DAI, DAI/WETH du réseau de référence (section 2.1), de frais  $\gamma_1, \gamma_2, \gamma_3$ . Ces trois pools forment un cycle (figure 4). Un arbitrage cyclique part du WETH, parcourt  $\text{WETH} \rightarrow \text{USDC} \rightarrow \text{DAI} \rightarrow \text{WETH}$  et revient avec une quantité de WETH strictement supérieure à l'engagement initial.

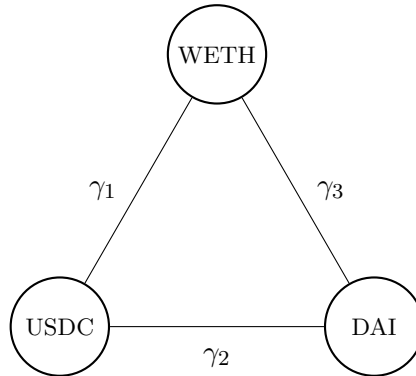


FIGURE 4 – Cycle de trois pools WETH/USDC, USDC/DAI, DAI/WETH. Le cycle  $\text{WETH} \rightarrow \text{USDC} \rightarrow \text{DAI} \rightarrow \text{WETH}$  est un arbitrage s'il conduit à une quantité finale de WETH strictement supérieure à la quantité engagée.

Notons  $r_{X/Y}$  le prix marginal du token  $Y$  en unités de  $X$ , hors frais : lors d'un swap marginal  $A \rightarrow B$ , on reçoit  $r_{B/A}$  unités de  $B$  par unité de  $A$ . En enchaînant les trois swaps marginaux avec frais, le facteur multiplicatif sur la quantité de WETH après un tour est, au premier ordre,

$$\gamma_1 \gamma_2 \gamma_3 r_{\text{USDC}/\text{WETH}} r_{\text{DAI}/\text{USDC}} r_{\text{WETH}/\text{DAI}}. \quad (10)$$

Le cycle est marginalement profitable si ce facteur dépasse 1. Sans frais, la non-arbitrage cyclique impose que le produit des prix marginaux le long du cycle vaille 1 : les prix locaux doivent dériver d'un système global. Avec frais, on tolère un écart borné par le coût d'un aller-retour  $\gamma_1 \gamma_2 \gamma_3$  —

la version cyclique de la bande (8), que la section 5.7 généralise. On vérifiera les deux cycles triangulaires de ce réseau à la section 6.6.

**Remarque 5.5** (Arbitrage théorique et arbitrage exécutable). La condition  $\Psi \geq 0$ ,  $\Psi \neq 0$  caractérise un arbitrage dans le modèle statique. En pratique, l'exécution dépend aussi des frais de gaz, de la latence, de la concurrence et du MEV (*maximal extractable value*). Les frais proportionnels créent déjà une bande ; les coûts fixes de gaz réduisent encore les opportunités exploitables (d'où une extension MICP) ; et la liquidité borne la taille des arbitrages, le prix marginal se déplaçant avec les réserves, de sorte qu'un arbitrage optimal est de taille finie (section 5.6). Ces effets ne remettent pas en cause la caractérisation mathématique ; ils déterminent quelles opportunités sont effectivement exploitables.

## 5.6 Arbitrage optimal et taille de transaction

On revient à un pool Uniswap V2 à deux tokens. La détection précédente est binaire : elle dit s'il existe un arbitrage, non sa taille. On la complète par l'arbitrage optimal face à un prix de référence externe, en forme fermée. Soit un pool  $(R_1, R_2)$  de frais  $\gamma$ , et un prix externe relatif  $p = \nu_1/\nu_2 > 0$  du token 1 en token 2. L'arbitragiste résout

$$\max_{\Delta, \Lambda \geq 0} \nu^\top (\Lambda - \Delta) \quad \text{s.c.} \quad \sqrt{(R_1 + \gamma\Delta_1 - \Lambda_1)(R_2 + \gamma\Delta_2 - \Lambda_2)} \geq \sqrt{R_1 R_2}.$$

**Proposition 5.6** (Arbitrage optimal pour un pool à produit constant). *Soit  $m = R_2/R_1$  et la bande  $[\gamma m, m/\gamma]$ .*

- *Si  $p \in [\gamma m, m/\gamma]$ , la transaction nulle est optimale : aucun arbitrage strictement profitable.*
- *Si  $p > m/\gamma$  (le pool sous-évalue le token 1), l'arbitrage optimal achète le token 1 en injectant*

$$\Delta_2^* = \frac{1}{\gamma} \left( \sqrt{\gamma p R_1 R_2} - R_2 \right),$$

*ce qui ramène le prix marginal d'achat, frais inclus, au prix externe  $p$  (le bord de la bande), pour un profit*

$$\Pi^* = \frac{\nu_2}{\gamma R_2} \left( \sqrt{\gamma p R_1 R_2} - R_2 \right)^2.$$

- *Si  $p < \gamma m$ , le cas est symétrique (vente du token 1 au pool).*

*Démonstration.* L'utilité étant croissante, la contrainte est saturée et une seule direction utilisée. Pour l'achat du token 1 ( $\Lambda_1, \Delta_2 \geq 0$ ,  $\Delta_1 = \Lambda_2 = 0$ ), la saturation  $(R_1 - \Lambda_1)(R_2 + \gamma\Delta_2) = R_1 R_2$  donne  $\Lambda_1 = R_1 \gamma \Delta_2 / (R_2 + \gamma\Delta_2)$ . L'objectif  $f(\Delta_2) = \nu_1 \Lambda_1 - \nu_2 \Delta_2$  est concave, et  $f'(\Delta_2) = \nu_1 R_1 \gamma R_2 / (R_2 + \gamma\Delta_2)^2 - \nu_2$  s'annule en  $R_2 + \gamma\Delta_2 = \sqrt{\gamma p R_1 R_2}$ , valeur positive ssi  $p > m/\gamma$ . Les réserves deviennent  $R'_2 = \sqrt{\gamma p R_1 R_2}$ ,  $R'_1 = R_1 R_2 / R'_2$ , de prix marginal  $R'^2_2 / (R_1 R_2) = \gamma p$  ; le prix d'achat frais inclus vaut  $\gamma p / \gamma = p$ . En substituant,  $\Pi^* = \nu_2 (R'_2 - R_2)^2 / (\gamma R_2)$ .  $\square$

Ceci quantifie la remarque 5.5 : l'arbitrage optimal est de taille finie et ramène le prix marginal au bord de la bande, exploitant l'écart jusqu'à disparition de l'opportunité une fois les frais pris en compte, sans épuiser le pool. Sur un réseau, l'arbitrage optimal aux prix  $\nu$  est le programme de détection d'objectif  $\nu^\top \Psi$ , convexe, mais sans forme close pool par pool en général.

## 5.7 Structure de l'ensemble des prix sans arbitrage

Le théorème 5.3 caractérise l'absence d'arbitrage par l'existence d'un certificat de prix  $g$  ; on décrit ici l'ensemble de tous ces certificats. On suppose les pools à deux tokens, le pool  $i$  reliant  $a(i)$  et  $b(i)$ , de prix marginal local  $\pi_i = (P_i)_{a(i)} / (P_i)_{b(i)}$ . En éliminant  $\lambda_i$  dans (8), la condition du pool  $i$  devient

$$\frac{g_{a(i)}}{g_{b(i)}} \in [\gamma_i \pi_i, \pi_i / \gamma_i].$$

Cet ensemble n'est pas un polyèdre tel quel (contraintes sur des rapports), mais le devient en coordonnées logarithmiques.

**Proposition 5.7** (Le domaine des prix est un polyèdre). *En coordonnées logarithmiques  $u = \ln g$ , l'ensemble  $G$  des certificats de prix sans arbitrage est le polyèdre convexe*

$$G = \left\{ u \in \mathbb{R}^n : |u_{a(i)} - u_{b(i)} - c_i| \leq \varepsilon_i, \ i = 1, \dots, m \right\}, \quad c_i = \ln \pi_i, \ \varepsilon_i = \ln(1/\gamma_i) \geq 0,$$

*invariant par translation  $u \mapsto u + t\mathbf{1}$  (soit  $g \mapsto e^t g$ ) : le libre choix d'un numéraire.*

*Démonstration.* Chaque encadrement de  $g_{a(i)}/g_{b(i)}$  s'écrit, en logarithme,  $|u_{a(i)} - u_{b(i)} - c_i| \leq \varepsilon_i$ ;  $G$  est l'intersection de ces paires de demi-espaces, donc un polyèdre convexe, invariant par  $u \mapsto u + t\mathbf{1}$  puisque les contraintes ne portent que sur des différences.  $\square$

On associe au réseau le graphe  $\mathcal{G} = (V, E)$  de sommets  $V$  (les tokens) et d'arêtes  $E$  (les pools, arête  $i$  entre  $a(i)$  et  $b(i)$ ). Les log-prix  $u$  jouent le rôle d'un potentiel sur  $\mathcal{G}$ , et les écarts  $c_i = \ln \pi_i$  celui de différences de potentiel le long des arêtes. Sans frais, exiger  $u_{a(i)} - u_{b(i)} = c_i$  revient à demander que ces différences dérivent d'un potentiel : c'est la loi des mailles, une somme nulle sur tout cycle. Avec frais, on tolère sur chaque maille une chute bornée par le coût d'un aller-retour.

## 6 Application numérique

On illustre le routage sur un réseau triangulaire de trois pools Uniswap V2, à réserves synthétiques mais réalistes. On échange du WETH contre du DAI, en comparant trois stratégies : route directe seule, route indirecte seule, et fractionnement optimal. La reproductibilité figure en annexe A.

### 6.1 Données

Le réseau comporte les pools WETH/USDC, WETH/DAI et USDC/DAI : le sous-réseau triangulaire des pools 1 à 3 de la figure 3 (le pool WETH/WBTC, sans route vers DAI, n'intervient pas). Les réserves (table 2) sont *synthétiques* : elles ne proviennent pas d'un relevé on-chain, mais sont choisies dans des ordres de grandeur réalistes et à prix mutuellement cohérents (WETH à 3000 unités de chaque stablecoin, stablecoins à 1), de sorte que le réseau soit sans arbitrage par construction (section 6.6). Le facteur de frais est  $\gamma = 0,997$  pour les trois pools. Le pool USDC/DAI étant nettement moins profond, la route indirecte sera vite pénalisée par le slippage sur ce segment.

TABLE 2 – Réserves synthétiques des trois pools (prix mutuellement cohérents).

| Pool      | Réserve 1                   | Réserve 2                       |
|-----------|-----------------------------|---------------------------------|
| WETH/USDC | $R_{\text{WETH}} = 2\,000$  | $R_{\text{USDC}} = 6\,000\,000$ |
| WETH/DAI  | $R_{\text{WETH}} = 800$     | $R_{\text{DAI}} = 2\,400\,000$  |
| USDC/DAI  | $R_{\text{USDC}} = 50\,000$ | $R_{\text{DAI}} = 50\,000$      |

### 6.2 Formulation numérique simplifiée

Pour un échange de  $q$  WETH vers DAI, deux routes existent : la route directe via WETH/DAI, et la route indirecte WETH→USDC→DAI via WETH/USDC puis USDC/DAI. En notant

$$f(z; R_{\text{in}}, R_{\text{out}}) = \frac{R_{\text{out}} \gamma z}{R_{\text{in}} + \gamma z}$$

la sortie d'un swap, la route directe produit  $f^{\text{dir}}(x) = f(x; 800, 2\,400\,000)$ , et la route indirecte, pour une entrée  $z$ ,

$$f^{\text{ind}}(z) = f(f(z; 2\,000, 6\,000\,000); 50\,000, 50\,000).$$

En envoyant  $x$  WETH par la route directe et  $q - x$  par l'indirecte, la sortie totale est

$$F_q(x) = f^{\text{dir}}(x) + f^{\text{ind}}(q - x), \quad x \in [0, q], \quad (11)$$

spécialisation de (6) aux deux routes. Comme  $f^{\text{dir}}$  est concave croissante,  $f^{\text{ind}}$  une composée de concaves croissantes, et  $z = q - x$  affine,  $F_q$  est concave sur  $[0, q]$  et admet un maximum unique, calculé par recherche par section dorée.

### 6.3 Résultat pour un ordre de 10 WETH

Pour  $q = 10$  WETH, les trois stratégies donnent les sorties de la table 3.

Le fractionnement optimal envoie  $x^* = 9,8217$  WETH par la route directe et 0,1783 par l'indirecte, soit une part directe de 98,22 %. Le gain sur la meilleure route unique (la directe) est de

$$29\,547,5280 - 29\,541,8349 = +5,6931 \text{ DAI} \approx 1,93 \text{ points de base.}$$

La route indirecte seule est très défavorable (18 621,4867 DAI) à cause de la faible profondeur du pool USDC/DAI ; mais en lui confiant une petite fraction de l'ordre, on exploite sa liquidité résiduelle pour réduire le slippage global.

TABLE 3 – Échange de 10 WETH vers DAI : comparaison des trois stratégies.

| Stratégie                         | DAI reçus   |
|-----------------------------------|-------------|
| Route directe seule ( $x = q$ )   | 29 541,8349 |
| Route indirecte seule ( $x = 0$ ) | 18 621,4867 |
| Fractionnement optimal            | 29 547,5280 |

#### 6.4 Sensibilité à la taille de l'ordre

On répète l'optimisation pour des tailles croissantes (table 4, figure 5).

TABLE 4 – Sensibilité du fractionnement optimal à la taille de l'ordre.

| $q$ (WETH) | DAI (directe) | DAI (optimal) | Part directe | Gain (bps) |
|------------|---------------|---------------|--------------|------------|
| 1          | 2 987,2771    | 2 987,2771    | 100,00 %     | 0,00       |
| 5          | 14 862,3887   | 14 863,4613   | 98,46 %      | 0,72       |
| 10         | 29 541,8349   | 29 547,5280   | 98,22 %      | 1,93       |
| 25         | 72 515,6832   | 72 555,8807   | 98,07 %      | 5,54       |
| 50         | 140 777,7843  | 140 937,1510  | 98,02 %      | 11,32      |
| 100        | 265 955,3184  | 266 538,8691  | 98,00 %      | 21,94      |

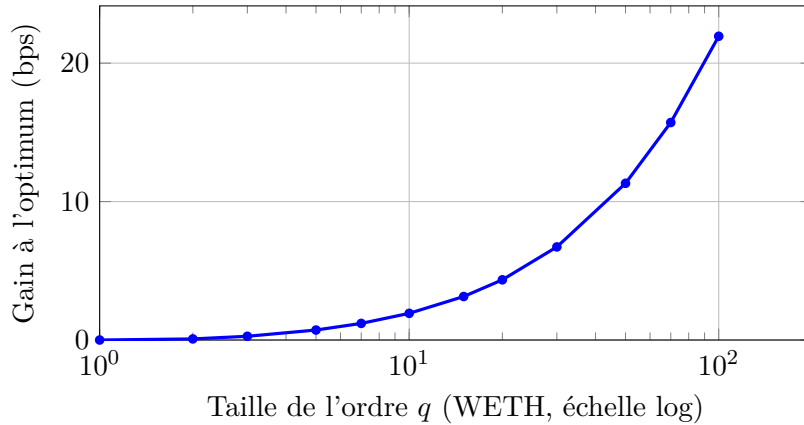


FIGURE 5 – Gain du fractionnement optimal sur la meilleure route unique, en fonction de la taille de l'ordre. Le bénéfice croît avec la taille, le slippage de la route directe augmentant et le partage devenant plus avantageux.

Deux tendances. Le gain croît avec la taille de l'ordre : nul pour 1 WETH (la route directe seule est optimale), il atteint près de 22 bps pour 100 WETH. Et la part optimale confiée à la route directe décroît lentement, de 100,00 % à 98,00 % : plus l'ordre est gros, plus il est utile d'en dériver une fraction vers l'indirecte, mais cette fraction reste modeste, limitée par la faible profondeur du pool USDC/DAI. Ces résultats confirment l'intuition de la section 1.5 : le bon objet n'est pas le meilleur chemin, mais la meilleure allocation de flux.

#### 6.5 Résolution du programme convexe général

La réduction (11) suppose connues les deux routes pertinentes. Pour la valider, on résout directement le programme convexe général (5) spécialisé à la liquidation WETH vers DAI, sans présumer de routes : les variables sont les paniers  $\Delta_i, \Lambda_i \in \mathbb{R}_+^2$  des trois pools et le bilan

$\Psi \in \mathbb{R}^3$ , avec  $\varphi_i(R) = \sqrt{R_1 R_2}$ , l'objectif  $e_{\text{DAI}}^\top \Psi$  et les contraintes  $e_{\text{WETH}}^\top \Psi = -q$ ,  $e_{\text{USDC}}^\top \Psi \geq 0$ . La résolution se fait par un solveur conique de points intérieurs (CLARABEL, via cvxpy).

TABLE 5 – Programme convexe général (cvxpy) contre réduction 1-D (11).

| $q$ (WETH) | DAI (général) | DAI (réduction 1-D) | écart                |
|------------|---------------|---------------------|----------------------|
| 10         | 29 547,5288   | 29 547,5280         | $8,1 \times 10^{-4}$ |
| 100        | 266 538,8850  | 266 538,8691        | $1,6 \times 10^{-2}$ |

Le programme général retrouve l'optimum à la tolérance du solveur près, et place la quasi-totalité du flux sur la route directe, le reliquat d'USDC intermédiaire étant nul. La réduction à deux routes capturerait donc l'optimum, et le cadre convexe des sections 3–4 s'applique directement au réseau.

## 6.6 Détection d'arbitrage et certificat de prix

On teste l'absence d'arbitrage sur le même réseau. Les prix marginaux locaux sont  $P_i = \frac{1}{2}(\sqrt{R_{i,2}/R_{i,1}}, \sqrt{R_{i,1}/R_{i,2}})$ . On propose le certificat le plus naturel pour un WETH à 3000 dollars et deux stablecoins à 1,  $g = (3000, 1, 1)$ , et l'on vérifie la bande (8) pool par pool en posant  $\lambda_i = \max_k (A_i^\top g)_k / (P_i)_k$ .

TABLE 6 – Vérification du certificat  $g = (3000, 1, 1)$  : la bande (8) est satisfaite pour les trois pools.

| Pool      | $P_i$             | $A_i^\top g$ | $\lambda_i$ | $\gamma \lambda_i P_i \leq A_i^\top g$ | $A_i^\top g \leq \lambda_i P_i$ |
|-----------|-------------------|--------------|-------------|----------------------------------------|---------------------------------|
| WETH/USDC | (27,3861, 0,0091) | (3000, 1)    | 109,5445    | (2 991,0000, 0,9970)                   | (3 000,0000, 1,0000)            |
| WETH/DAI  | (27,3861, 0,0091) | (3000, 1)    | 109,5445    | (2 991,0000, 0,9970)                   | (3 000,0000, 1,0000)            |
| USDC/DAI  | (0,5000, 0,5000)  | (1, 1)       | 2,0000      | (0,9970, 0,9970)                       | (1,0000, 1,0000)                |

Les trois bandes sont vérifiées : par le sens suffisant du théorème 5.3, le réseau ne contient pas d'arbitrage. Un prix de 3000 \$ pour le WETH et de 1 \$ pour les deux stablecoins est donc compatible, à frais près, avec les prix marginaux de tous les pools. (La borne supérieure est atteinte avec égalité sur la composante réalisant le maximum, par définition de  $\lambda_i$  ; c'est la borne inférieure qui constitue le contenu de la vérification. Les prix étant exactement cohérents, les deux composantes saturant simultanément la borne supérieure.)

On corrobore le diagnostic en évaluant les deux cycles triangulaires, frais inclus (facteur  $\gamma^3$  pour trois swaps), conformément à la section 5.5 :

$$\underbrace{\text{WETH} \rightarrow \text{USDC} \rightarrow \text{DAI} \rightarrow \text{WETH}}_{\text{facteur } 0,991027}, \quad \underbrace{\text{WETH} \rightarrow \text{DAI} \rightarrow \text{USDC} \rightarrow \text{WETH}}_{\text{facteur } 0,991027}.$$

Les deux facteurs sont strictement inférieurs à 1 : aucun arbitrage triangulaire.

**Remarque 6.1.** Un certificat de prix prouve l'absence d'arbitrage directement et exactement : il suffit de vérifier des inégalités. Résoudre le programme de détection (7) à un état sans arbitrage est en revanche numériquement délicat, l'optimum y étant nul et dégénéré : une raison pratique de privilégier l'exhibition d'un certificat.

## 7 Conclusion

On a présenté la modélisation du routage optimal dans les réseaux de CFMM. Après une introduction au fonctionnement d'un AMM et aux notions de prix marginal, de prix effectif et de slippage, le routage a été formulé comme un problème d'optimisation sur un réseau de pools. La relaxation des contraintes de trading produit un programme convexe (proposition 3.3), exact sous des hypothèses de croissance et de saturation (proposition 3.6).

La dualité convexe (section 4) en interprète les variables duales comme un système de prix d'équilibre  $g = \nabla U(\Psi)$  et comme la valeur marginale de la liquidité  $\lambda_i$  de chaque pool, cette dernière s'interprétant comme la sensibilité de la valeur optimale à la profondeur du pool. La condition de faisabilité duale est précisément la bande caractérisant l'absence d'arbitrage.

La détection d'arbitrage est un cas particulier du même cadre, pour une utilité linéaire et un bilan non négatif. L'absence d'arbitrage équivaut à l'existence d'un système de prix global compatible, à un facteur d'échelle local près, avec les prix marginaux de chaque pool (théorème 5.3). Le sens suffisant a été démontré élémentairement à partir de la concavité ; le sens nécessaire via la dualité forte du routage, la spécialisation au cas dégénéré étant admise d'après [4]. Cette condition se lit en termes de cycles : un arbitrage est un cycle de swaps profitable, et l'absence d'arbitrage borne le produit des prix marginaux le long de tout cycle par les frais d'un aller-retour. L'ensemble des certificats forme, en coordonnées logarithmiques, un polyèdre convexe (proposition 5.7) qui, à la limite sans frais et sur un réseau connexe, se réduit à un prix unique au numéraire près.

L'application numérique sur le réseau triangulaire confirme l'intérêt du fractionnement : le gain sur la meilleure route unique croît avec la taille de l'ordre, atteignant près de 22 points de base pour 100 WETH. Le programme convexe général (cvxpy) retrouve cet optimum sans présupposer de routes, et le certificat  $g = (3000, 1, 1)$  établit l'absence d'arbitrage.

## A Reproductibilité de l'expérience numérique

**Données.** Les réserves de la table 2 sont *synthétiques* : elles ne proviennent d'aucun relevé on-chain. Choies dans des ordres de grandeur réalistes et à prix mutuellement cohérents (WETH à 3000 unités de chaque stablecoin, stablecoins à 1), elles rendent le réseau sans arbitrage par construction. Le facteur de frais est  $\gamma = 0,997$  (0,30 %). Ce choix reproduit l'intégralité des résultats sans dépendance à un indexeur ou à un nœud Ethereum.

**Modèle de swap.** Chaque swap suit la formule à produit constant avec frais sur le montant entrant,  $f(z; R_{\text{in}}, R_{\text{out}}) = R_{\text{out}}\gamma z / (R_{\text{in}} + \gamma z)$ . La route directe l'applique une fois (WETH/DAI) ; la route indirecte deux fois, en composant WETH→USDC puis USDC→DAI.

**Optimisation.** Pour une taille  $q$ , on maximise  $F_q$  de (11) sur  $[0, q]$  par recherche par section dorée. Le code ci-dessous reproduit les tables 3 et 4.

```
gamma = 0.997
def swap(z, Rin, Rout):
    return Rout*gamma*z/(Rin + gamma*z) if z > 0 else 0.0

# WETH/DAI (direct) ; WETH/USDC ; USDC/DAI (reserves synthétiques)
def direct(x): return swap(x, 800.0, 2400000.0)
def indirect(z):
    y = swap(z, 2000.0, 6000000.0) # WETH -> USDC
    return swap(y, 50000.0, 50000.0) # USDC -> DAI

def F(x, q): return direct(x) + indirect(q - x)

def optimize(q): # section dorée (F concave)
    import math
    a, b, gr = 0.0, q, (math.sqrt(5)-1)/2
    c, d = b-gr*(b-a), a+gr*(b-a)
    for _ in range(200):
        if F(c,q) < F(d,q): a = c
        else: b = d
        c, d = b-gr*(b-a), a+gr*(b-a)
    x = 0.5*(a+b)
    return x, F(x, q)
```

**Programme convexe général et certificat.** La table 5 résout le programme général (5), spécialisé à la liquidation et modélisé en cvxpy. Les contraintes de trading y sont écrites sous la forme

$$\text{geo\_mean}(R_i + \gamma_i \Delta_i - \Lambda_i) \geq \sqrt{R_{i,1} R_{i,2}},$$

puis résolues par CLARABEL. La table 6 vérifie les inégalités de non-arbitrage : pour  $g = (3000, 1, 1)$ , on calcule

$$P_i = \frac{1}{2} \left( \sqrt{R_{i,2}/R_{i,1}}, \sqrt{R_{i,1}/R_{i,2}} \right), \quad \lambda_i = \max_k \frac{(A_i^\top g)_k}{(P_i)_k},$$

et l'on contrôle  $\gamma \lambda_i P_i \leq A_i^\top g \leq \lambda_i P_i$ . Les facteurs triangulaires sont des produits de taux marginaux multipliés par  $\gamma^3$ .

**Environnement.** La réduction 1-D ne requiert que la bibliothèque standard de Python 3 (module `math`). Le programme général utilise `numpy` et `cvxpy` (solveur CLARABEL). Toutes les sorties sont déterministes; les écarts résiduels de la table 5 relèvent de la tolérance du solveur conique.

## B Glossaire minimal des notions AMM

Le vocabulaire des marchés décentralisés recouvre des objets mathématiques simples : un pool s'identifie à une courbe, un échange à un déplacement le long de cette courbe, un prix à une pente. La table 7 en rassemble les correspondances.

TABLE 7 – Traduction du vocabulaire des AMM en objets mathématiques.

| Terme du domaine                | Objet mathématique                                       |
|---------------------------------|----------------------------------------------------------|
| Token                           | Une coordonnée de $\mathbb{R}^n$ (un actif)              |
| Pool (à deux tokens)            | Une courbe du plan des réserves ; une arête du réseau    |
| Réserves $R_i$                  | Un point sur la courbe (l'état du pool)                  |
| Échange (swap)                  | Un déplacement le long de la courbe                      |
| Fonction de trading $\varphi_i$ | Fonction concave ; son sur-niveau = échanges admissibles |
| Frais $\gamma_i \in (0, 1]$     | Friction : remplace une égalité par une bande            |
| Prix marginal                   | Pente de la tangente, $\nabla \varphi_i$                 |
| Prix effectif                   | Pente de la corde (échange de taille finie)              |
| Arbitrage                       | Gain sans apport net initial                             |
| Routage optimal                 | Un problème de flot sur le réseau de pools               |
| Prix d'équilibre $g$            | Variables duales du programme de routage                 |

## Références

- [1] H. Adams, N. Zinsmeister et D. Robinson. *Uniswap v2 Core*. Livre blanc technique, 2020.
- [2] G. Angeris, H.-T. Kao, R. Chiang, C. Noyes et T. Chitra. *An Analysis of Uniswap Markets*. arXiv :1911.03380, 2019.
- [3] G. Angeris, A. Agrawal, A. Evans, T. Chitra et S. Boyd. *Constant Function Market Makers : Multi-Asset Trades via Convex Optimization*. arXiv :2107.12484, 2021.
- [4] G. Angeris, A. Evans, T. Chitra et S. Boyd. *Optimal Routing for Constant Function Market Makers*. Proceedings of the 23rd ACM Conference on Economics and Computation (EC '22), 2022.
- [5] S. Boyd et L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [6] F. Martinelli et N. Mushegian. *Balancer : A Non-Custodial Portfolio Manager, Liquidity Provider, and Price Sensor*. Livre blanc, 2019.