

Bachelorarbeit

Vergleich der Constant Function Market Maker UniSwap, Balancer, Curve und Bancor

Dario Thürkauf

Betreut von:

Prof. Dr. Fabian Schär

Credit Suisse Asset Management (Schweiz) Professor for
Distributed Ledger Technologies and Fintech
Center for Innovative Finance, University of Basel

Abstract

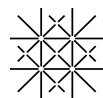
In dieser Bachelorarbeit werden die Constant Function Market Maker Protokolle von UniSwap, Balancer, Curve und Bancor verglichen. Dieser Vergleich erfolgt anhand der jeweiligen Preisfunktion, dem Slippage, dem Bereitstellen und Abziehen von Liquidität und dem Impermanent Loss. Der Fokus gilt dabei der mathematischen Umsetzung. In der Diskussion folgt ein Effizienzabgleich, bei dem die Auswirkungen der unterschiedlichen Implementierungen auf Liquiditätsprovider und Händler dargelegt werden. Schlussendlich wird das Front-Running Problem im Kontext von Constant Function Market Makers erörtert.

Keywords: Decentralized Finance, Smart Contracts, dezentrale Tauschbörsen, Constant Function Market Makers, Liquiditätspools

JEL: G15, G23

Inhaltsverzeichnis

1	Einleitung	1
2	Protokolle	4
2.1	UniSwap	5
2.2	Balancer	9
2.3	Curve	12
2.4	Bancor	16
3	Diskussion	19
3.1	Slippage	19
3.2	Impermanent Loss	21
3.3	Front-Running Problem bei CFMMs	22
4	Zusammenfassung	24
5	Anhang	27
5.1	Balancer (UniSwap) Spot-Preis	27
5.2	Balancer Tausch-Formel	28
5.3	Bancor Tausch-Formel	28
	Verzeichnisse	i



**University
of Basel**

Center for
Innovative Finance

Plagiatserklärung

Ich bezeuge mit meiner Unterschrift, dass meine Angaben über die bei der Abfassung meiner Arbeit benutzten Hilfsmittel sowie über die mir zuteil gewordene Hilfe in jeder Hinsicht der Wahrheit entsprechen und vollständig sind. Ich habe das Merkblatt zu Plagiat und Betrug vom 22. Februar 2011 gelesen und bin mir der Konsequenzen eines solchen Handelns bewusst.

A handwritten signature in black ink, appearing to read 'D. Thürkauf'. The signature is stylized with a large 'D' and a long, sweeping tail.

Dario Thürkauf

1 Einleitung

Decentralized Finance (DeFi) zeichnet sich durch die Unabhängigkeit von Intermediären und zentralen Instanzen aus. In den letzten Jahren gewann diese Ethereum Blockchain-Anwendung zunehmend an Bedeutung. (vgl. Schär, 2020, S.1)

Die Anzahl digitaler Werteinheiten nimmt stetig zu (John, 2020). Stand November 2020 sind 7600 Kryptoassets auf Coinmarketcap gelistet.¹ Dies verdeutlicht die Notwendigkeit von Tauschbörsen für den Handel dieser Kryptoassets.

Zentrale Tauschbörsen sind laut Schär (2020, S.8) zwar relativ effizient, verfügen aber über einige Nachteile. Ein Nutzer muss vor dem Handel seine Kryptoassets auf der Tauschbörse hinterlegen. Vertrauen gegenüber dem Anbieter ist erforderlich, da die direkte Kontrolle über die Assets abgegeben wird und der Anbieter diese für sich behalten oder verlieren könnte. Gleichzeitig bietet sich ein zentraler Angriffspunkt für Drittparteien. Zusätzlich ist die regulatorische Grundlage solcher Tauschbörsen oft unklar und einige mussten innert kurzer Zeit starke Skalierungsmaßnahmen vornehmen.

In der Vergangenheit gingen bei Hacks von populären Tauschbörsen wie Mt. Gox, Quadriga oder Bitfinex Kundengelder in Milliardenhöhe verloren (Angeris et al., 2019, S.1). Weitere Probleme von zentralen Tauschbörsen sind hohe Listungsgebühren sowie fehlende Liquidität für Assets mit niedrigem Marktvolumen, wie beispielsweise persönliche Token (John, 2020).

Dezentrale Tauschbörsen (DEX) versuchen diese Probleme zu lösen, wobei es eine Reihe unterschiedlicher Modelle gibt. Order Bücher sind laut Angeris et al. (2019, S.2) der dominante Lösungsansatz für Tauschbörsen im traditionellen Finanzsystem und können auch auf einer Blockchain mittels Smart Contracts (on-Chain) abgewickelt werden.

Dabei wird jede Order in einem Smart Contract gespeichert, unabhängig von Drittanbietern oder zusätzlicher Infrastruktur. Dieser Prozess ist relativ langsam und teuer. Bereits die Intention zu Handeln kann

¹www.coinmarketcap.com, Zugriff: 8.11.2020

Netzwerk-Gebühren erzeugen. Dieser Nachteil wiegt vor allem auf volatilen Märkten schwer, wo Gebote oft zurückgezogen werden. (vgl. Schär, 2020, S.9)

Eine Alternative ist die Abwicklung von Order Büchern abseits der Blockchain (off-Chain). Gegen eine Gebühr führen zentrale Anbieter Listen mit Angeboten. Die Anbieter führen weder die Trades aus, noch sind sie in Besitz der Kryptoassets. Sie sind somit nur für den Informationsaustausch zwischen Käufer- und Verkäufer zuständig, was das Sicherheitsrisiko reduziert. Wettbewerb unter den Anbietern und die Offenheit des Protokolls stellen sicher, dass Abhängigkeiten vermieden werden. (vgl. Schär, 2020, S.8)

Doch auch dieses Modell ist laut Angeris et al. (2019, S.2) mit einigen Sicherheitslücken verbunden, weshalb Nutzer oft zusätzliche Sicherheitsmassnahmen treffen müssen.

Peer-to-Peer (P2P) und Over-the-Counter (OTC) Protokolle ermöglichen bilateralen Handel zwischen zwei Parteien. Über das Netzwerk können Nutzer Gegenparteien suchen und Preise verhandeln. Dieser Prozess ist normalerweise automatisiert und off-Chain Indexes unterstützen bei der Preisfindung. (vgl. Schär, 2020, S.11)

Bei Smart Contract basierten Reserve Aggregatoren werden Liquiditätsreserven von Liquiditätsprovidern in einem Smart Contract konsolidiert. Ein Nutzer kann eine Order dem Smart Contract senden, wobei dieser die verschiedenen Preise vergleicht und den besten aus Sicht des Nutzers akzeptiert. (vgl. Schär, 2020, S.10f.)

Smart Contract basierte Liquiditätspools

Laut Berenzon (2020) geht der Begriff Automated Market Maker (AMM) auf die algorithmische Spieltheorie zurück. AMMs dienen der Informationsaggregation und finden Anwendung in Märkten, bei denen die Auszahlungen von einem zukünftigen Status abhängig sind. Ein Beispiel dafür sind Prognosemärkte. Eine spezielle Form von AMMs sind *Constant Function Market Makers* (CFMM). Sie wurden für den dezentralen Austausch von digitalen Werteinheiten konstruiert. CFMMs sind algorithmi-

sche Agenten, welche vordefinierte Funktionen ausführen und als Resultat Liquidität für elektronische Märkte bereitstellen. Dies geschieht mithilfe von Smart Contract basierten Liquiditätspools. Diese halten Liquiditätsreserven von zwei oder mehr Kryptoassets und werden von einem Code kontrolliert. Nutzer handeln direkt gegen den Smart Contract, wobei der Preis intern über die definierte Funktion bestimmt wird. Ein zentraler Marktmacher wird dadurch nicht mehr benötigt und Spreads fallen weg. Liquiditätspools können von beliebigen Personen (Liquiditätsprovider) bereitgestellt werden. Dies wird mittels Schaffung von ökonomischen Anreizen sichergestellt. (vgl. Berenzon, 2020)

CFMMs haben einige vorteilhafte Eigenschaften:

- Nutzer bleiben im Besitz ihrer Kryptoassets, das Gegenpartearisiko entfällt. Beide Seiten des Tauschs finden atomar über eine einzelne Blockchain-Transaktion statt. (vgl. Schär, 2020, S.8)
- Jede Person kann mit dem Smart Contract handeln oder Liquidität bereitstellen. Listungsgebühren fallen weg. CFMMs sind zensurrensistent und unterstützen auch Kryptoassets mit niedrigem Marktvolumen. (vgl. John, 2020)
- Im Gegensatz zu Order Büchern brauchen CFMMs weniger Speicherplatz. Der Status eines CFMMs kann durch die Anzahl der gepoolten Assets eindeutig repräsentiert werden. Dadurch sind sie einfacher auf der Blockchain zu bewirtschaften. (vgl. Angeris et al., 2019, S.3)
- Die interne Logik der Applikation ist öffentlich und kann von jedem beobachtet werden. Die Durchführung des Smart Contracts ist über die Blockchain besichert (Schär, 2020, S.2). Dies steht im Gegensatz zu Matching-Algorithmen bei zentralisierten Tauschbörsen, welche nicht öffentlich einsehbar sind (Angeris et al., 2019, S.3).
- Smart Contracts können in weitere Applikationen integriert werden. CFMMs sind somit ein wichtiger Bestandteil eines dezentralen Finanzsystems. (vgl. Schär, 2020, S.4)

Es gibt eine Reihe unterschiedlicher Funktionen für CFMMs. Gemeinsamkeit aller Implementierungen ist die Anwendung einer konvexen Funktion. Ein Liquiditätspool mit dieser Eigenschaft kann immer Liquidität bereitstellen, da der relative Preis eines Tokens mit abnehmenden Reserven gegen Unendlich strebt. (vgl. Schär, 2020, S.9)

Constant Product Market Makers, Constant Mean Market Makers, Hybrid Function Market Makers und Constant Reserve Market Makers sind Implementierungen, die bereits erfolgreich umgesetzt wurden und aus diesem Grund in Kapitel 2 verglichen werden. Abbildung 1 liefert eine Übersicht der verschiedenen Krypto-Tauschbörsen.

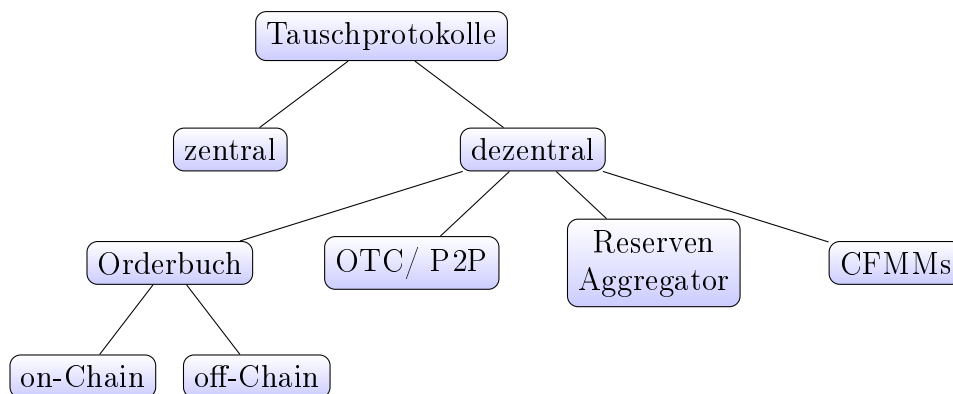


Abbildung 1: Übersicht Krypto-Tauschprotokolle, Quelle: eigene Darstellung in Anlehnung an Schär (2020, S.8-11)

2 Protokolle

Es folgt die Betrachtung der Protokolle von UniSwap, Balancer, Curve und Bancor. Diese Protokolle werden anhand der zugrundeliegenden Funktionsgleichung, dem Slippage, dem Bereitstellen und Abziehen von Liquidität und dem Impermanent Loss verglichen. Die Begrifflichkeiten Slippage und Impermanent Loss werden anfangs anhand von UniSwap dargelegt.

2.1 UniSwap

UniSwap V2 ist ein Constant Product Market Maker (Berenzon, 2020). Das Protokoll besteht aus einem Set an Smart Contracts und ermöglicht den automatischen Tausch zwischen zwei beliebigen ERC20-Token. Über den UniSwap Factory Kontrakt kann jeder Nutzer ein neues Tauschpaar erstellen. Der Factory Kontrakt führt Register über die bereits vorhandenen Paare, da es pro Paar nur einen UniSwap Exchange Kontrakt geben kann. Ein Exchange Kontrakt hält einen Liquiditätspool bestehend aus beiden Assets. Nutzer handeln gegen den Exchange Kontrakt. Die Tauschgebühr bei UniSwap beläuft sich momentan auf 0.3%. Diese Gebühr wird nach dem Tausch dem Liquiditätspool hinzugefügt. (vgl. Adams, 2020)

Somit wächst mit jedem Tausch der Liquiditätspool um die Tauschgebühr, wie in Abbildung 2 veranschaulicht.

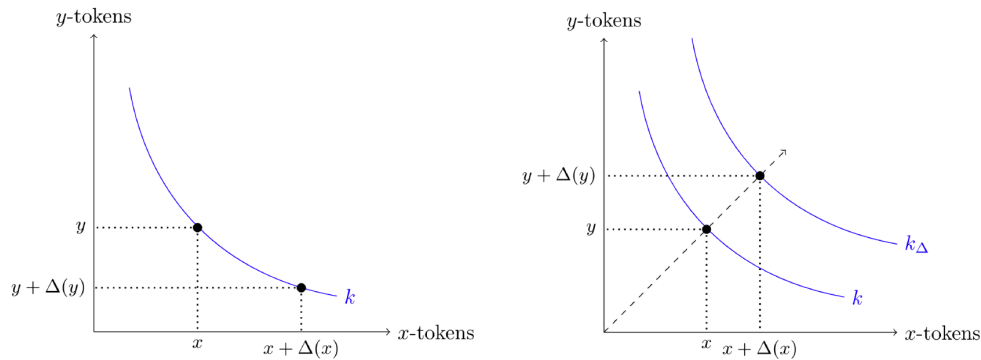


Abbildung 2: Constant Product Funktion, Quelle: Schär (2020, S.10)

Funktionsgleichung

Der Preis für einen Tausch wird laut Adams (2020) über die Constant Product Gleichung (1) bestimmt:

$$x \cdot y = k \quad (1)$$

x steht für die Anzahl Token von Asset A und y für die Anzahl Token von Asset B. k wird aus dem Produkt von x und y berechnet und während einem Tausch konstant gehalten. Das Verhältnis zwischen den Token im Liquiditätspool bestimmt deren relativen Preis. Über Erweiterung von Gleichung (1) und unter Berücksichtigung der Handelsgebühr f lässt sich die Anzahl Token Δy berechnen, welche ein Nutzer für einen bestimmten Input Δx erhält:

$$(x + (1 - f)\Delta x) \cdot (y + \Delta y) = k \quad (2)$$

$$\Delta y = \frac{k}{x + (1 - f)\Delta x} - y \quad (3)$$

Für $\Delta x > 0$ nimmt Δy immer negative Werte an und umgekehrt. Durch einen Tausch wird die Reserve eines Assets erhöht, während die Reserve des anderen Assets schrumpft. (vgl. Schär, 2020, S.10)

Der effektive Preis für Token x in Einheiten von Token y bei einem Tausch ist gemäss Martinelli und Mushegian (2019):

$$EP_x = \frac{\Delta y}{\Delta x} \quad (4)$$

Abbildung 2 zeigt den Funktionsgraphen der Constant Product Gleichung. Alle möglichen Verteilungen der Token im Liquiditätspool liegen auf k . Es gilt: Je geringer die Reserve eines Assets, desto höher dessen relativer Preis. Durch die Konvexität der Kurve ist es nicht möglich, ein Asset vollständig dem Pool zu entziehen (Buterin, 2018). Die Steigung des Graphen in einem bestimmten Punkt ist der Preis für einen indefinitesimalen Trade. Angeris et al. (2019, S.4) zeigen, dass die Gleichung für den marginalen Preis SP_x , nachfolgend Spot-Preis genannt, von Token x in Einheiten von Token y über partielle Ableitung der Constant Product Funktion (1) hergeleitet werden kann:

$$SP_x = \frac{y}{x} \quad (5)$$

Slippage

Bei einem Tausch weicht der effektive Preis vom Spot-Preis ab. Gemäss Martinelli (2020b) ist Slippage definiert als die prozentuale Abweichung des effektiven Preis vom Spot-Preis:

$$SL = \frac{EP_x}{SP_x} - 1 \quad (6)$$

Wobei $SP_x = \lim_{\Delta x, \Delta y \rightarrow 0} EP_x$ gilt. Durch Einsetzen der Gleichungen 4 und 5 in (6) lässt sich zeigen, dass der Slippage abhängig von der Höhe des Inputs (Δx), sowie der Poolgrösse (x und y) ist. Je höher der Input im Verhältnis zur Poolgrösse, desto höher der Slippage (Berenzon, 2020).

Liquidität bereitstellen und abziehen

Der initiale Liquiditätsprovider des Exchange Kontrakts kann das anfängliche Token-Verhältnis im Liquiditätspool beliebig festlegen. Rationale initiale Liquiditätsprovider werden Token im gleichen Wert dem Pool bereitstellen. Ansonsten weicht der relative Preis der Token im Pool vom Aussenmarkt ab und Arbitrageure können auf Kosten der Liquiditätsprovider risikolose Gewinne erzielen. Nachfolgende Liquiditätsprovider benutzen das Verhältnis zum Zeitpunkt ihres Deposits. Liquiditätsprovider erhalten Liquiditätstoken für den Beitrag zum Liquiditätspool. (vgl. Adams, 2020)

Die Anzahl neu geschaffener LT_{neu} berechnet sich gemäss Adams (2020) über:

$$LT_{neu} = LT_{alt} \cdot \frac{\Delta A}{A} \quad (7)$$

Wobei LT_{alt} die Anzahl bereits vorhandener Liquiditätstoken, ΔA die vom Liquiditätsprovider beigesteuerte Anzahl Token x und A die bisherige Anzahl Token x im Kontrakt ist.

Liquiditätsprovider können jederzeit ihren proportionalen Anteil am Liquiditätspool abziehen. Dabei werden die Liquiditätstoken „vernichtet“. Das Verhältnis der erhaltenen Token entspricht dabei dem momentanen

Allokationsverhältnis im Pool und kann vom Anfangsinvestment abweichen. Da bei dem Hinzufügen der Tauschgebühren zum Liquiditätspool keine neuen Liquiditätstoken geschaffen werden, sind in den abgezogenen Token ein proportionaler Anteil aller gesammelten Gebühren enthalten. (vgl. Adams, 2020)

Impermanent Loss

Verändern sich die relativen Preise der Pool-Token im Aussenmarkt, werden Arbitrageure gegen den Kontrakt handeln, bis der relative Pool-Preis dem Aussenmarkt entspricht. Wegen der Handelsgebühr kann eine kleine Abweichung bestehen bleiben. (vgl. Angeris et al., 2019, S.4f.)

Das Verhältnis der Token widerspiegelt die relative Nachfrage nach den Assets. Eine Preisveränderung zieht daher eine Veränderung der Token-Reserven im Pool nach sich. Relativ zum Halten der Assets resultiert ein einseitiger Preisunterschied zwingend in einem Verlust für Liquiditätsprovider. Dieser Verlust wird Impermanent Loss genannt, da er sich erst bei Abzug der Mittel vom Liquiditätspool materialisiert. (vgl. Pintail, 2019)

Gemäss Martinelli (2020b) lautet die allgemeine Formel für den Impermanent Loss ausgedrückt in US-Dollar:

$$IL = \frac{PoolWert_{neu}^{USD}}{Haltewert_{neu}^{USD}} - 1 \quad (8)$$

Aus Gleichung 8 leitet Martinelli (2020b) den Impermanent Loss in Abhängigkeit der Preisänderung der Token ab, dadurch folgt:

$$IL = \frac{2(\Delta P_x^{USD} \Delta P_y^{USD})^{0.5}}{\Delta P_x^{USD} + \Delta P_y^{USD}} - 1 \quad (9)$$

Wobei $\Delta P_{x,y}^{USD}$ als $\frac{Preis_{neu}}{Preis_{alt}}$ der Token in US Dollar definiert ist. Je stärker der relative Preisunterschied zwischen den Assets, desto höher der Impermanent Loss. Gemäss Pintail (2019) ist der Impermanent Loss bei UniSwap unabhängig von der Richtung des relativen Preisunterschieds.

2.2 Balancer

Balancer generalisiert die Constant Product Funktion von UniSwap auf Liquiditätspools mit bis zu 8 verschiedenen Token. Berenzon betitelt diese Implementierung *Constant Mean Market Maker* (Berenzon, 2020).

Funktionsgleichung und Slippage

Martinelli und Mushegian (2019) definieren den Preismechanismus über die Constant Value Funktion:

$$V = \prod B_t^{w_t} \quad (10)$$

V ist der Gesamtwert des Liquiditätspools und wird während einem Tausch konstant gehalten. Wiederum wird nach jedem Tausch die Handelsgebühr hinzugefügt. B_t ist die Anzahl und w_t das normalisierte Gewicht von Token t . w_t sollte dem Wertanteil eines Tokens t im Verhältnis zum Wert des gesamten Liquiditätspools entsprechen. Vorausgesetzt Arbitrage findet statt und die Handelsgebühr ist tief. Die Summe aller normalisierten Gewichte $\sum_t w_t$ ergibt 1.

Balancer Core Pools sind entweder kontrolliert (privat) oder finalisiert (öffentlich). Kontrollierte Pools können von einer einzelnen Adresse konfiguriert werden. Diese Adresse kann Liquidität dem Pool hinzufügen oder abziehen, sowie die gepoolten Token-Typen und deren jeweilige Gewichte verändern. Über eine Einwegfunktion können kontrollierte Pools finalisiert werden. Finalisierte Pools haben fixierte Token-Typen, -Gewichte und Tauschgebühren. Der Pool wird dadurch für weitere Liquiditätsprovider zugänglich. Tauschgebühren werden bei einem Tausch und bei Abzug der Liquidität fällig. (vgl. Martinelli und Mushegian, 2019)

Der Spot Preis SP_i (siehe 5.1) und effektive Preis EP_i von Token i in Einheiten Token o ist laut Martinelli und Mushegian (2019):

$$SP_i = \frac{\frac{B_o}{w_o}}{\frac{B_i}{w_i}} \quad EP_i = \frac{A_o}{A_i} \quad (11)$$

Index i steht für die dem Pool hinzugefügten, o für die vom Pool abgezogenen Token. B sind die Pool-Reserven und A die Anzahl der getauschten Token. Diese Formeln sind äquivalent zu den UniSwap Preis-Formeln (4) und (5), wo die Gewichte jeweils 0.5 sind und gekürzt werden können. Martinelli und Mushegian (2019) zeigen, dass die Anzahl Token A_o , die Nutzer für den Input einer bestimmten Anzahl Token A_i erhalten, sich wie folgt berechnet (siehe 5.2): ²

$$A_o = B_o \cdot \left(1 - \left(\frac{B_i}{B_i + A_i} \right)^{\frac{w_i}{w_o}} \right) \quad (12)$$

Gleichung 12 kann nach A_i umgestellt werden, um den nötigen Input für eine gewünschte Anzahl Output A_o zu erhalten.

Der Slippage definiert sich gleich wie in Gleichung 6. Daher lässt sich schreiben:

$$SL = \frac{A_o}{A_i} \cdot \frac{B_i}{B_o} \cdot \frac{w_o}{w_i} - 1 \quad (13)$$

Die Verteilung der Gewichte spielt also eine Rolle für die Höhe des Slippage. Über Einsetzen der Gleichung 12 in (13) lässt sich der Slippage in Abhängigkeit des Inputs A_i bei gegebenen Pool-Reserven und Gewichten berechnen.

Liquidität bereitstellen und abziehen

Liquiditätsprovider können alle Token mit entsprechender Gewichtung oder nur einen einzigen Token öffentlichen Liquiditätspools hinzufügen. Wie bei UniSwap werden durch das Hinzufügen der Assets neue Liquiditätstoken geschaffen. Das Hinzufügen eines einzigen Tokens A ist gleichzusetzen mit dem Poolen aller Token und dem sofortigen Tausch der restlichen Token zu Token A . Im Gegensatz zum Poolen aller Token beinhaltet das Hinzufügen eines einzigen Tokens Tauschgebühren. (vgl. Martinelli und Mushegian, 2019)

²ohne Handelsgebühr

Impermanent Loss

Für die Berechnung des Impermanent Loss in einem Balancer Pool gilt gemäss Martinelli (2020b) Gleichung 9, generalisiert für n-Token:

$$IL = \frac{\prod_t (\Delta P_t^{USD})^{w_t}}{\sum_t (\Delta P_t^{USD} \cdot w_t)} - 1 \quad (14)$$

ΔP_t^{USD} ist wiederum der Preisunterschied $\frac{Preis_{neu}}{Preis_{alt}}$ eines Tokens t in US Dollar. Der Impermanent Loss ist abhängig von den Gewichten der jeweiligen Token im Pool.

Balancer Smart Pools

Zusätzlich zu den bereits erwähnten Core Pools hat Balancer das Protokoll mit sogenannten Smart Pools erweitert. Anstelle von fixierten Pool Parametern, wie bei finalisierten Pools, verfügen Smart Pools über die Möglichkeit die Pool-Parameter flexibel zu gestalten. Die veränderbaren Parameter sind gemäss Hoffman (2020):

- Token-Typen, Token-Gewichte, Tauschgebühr, zugelassene Liquiditätsprovider, maximaler Deposit Wert, Möglichkeit zu Handeln

Diese Variablen können durch einen externen Smart Contract kontrolliert werden. Sie sind also programmierbar, was viele weitere Möglichkeiten eröffnet. Hoffman (2020) nennt zwei beispielhafte Anwendungen:

1. Dynamische Gebühren: Anpassung der Tauschgebühr an die Tauschnachfrage (über Orakel). Bei hoher Tauschnachfrage könnte die Tauschgebühr erhöht werden, bei tiefer Nachfrage verringert. Eine Implementierung dieser Art könnte den Ertrag für Liquiditätsprovider optimieren.
2. Initiale Token Distribution: Gewichtung des zu verteilenden Tokens beginnt hoch (bspw. 0.9) und sinkt über einen bestimmten Zeitraum (bswp. 0.1 Schritte jeden Tag bis auf 0.1). Umgekehrt wird

dem Pool eine beliebige Anzahl Ether hinzugefügt, dessen Gewicht in gleichen Schritten steigt. Wenn niemand ETH gegen den Token tauscht, sinkt der Spot-Preis stetig. Beim Erwerben des Tokens steigt der Spot-Preis. Eine Distribution dieser Art kann also bei der Preisfindung eines neu emittierenden Tokens helfen und hat die Eigenschaften einer niederländischen Auktion. Ebenfalls sind die Kapitalerfordernisse für die Verteiler tief.

Aufgrund der unterschiedlichen Eigenschaften, werden Smart Pools in der Diskussion nicht den anderen Protokollen gegenübergestellt.

2.3 Curve

Stablecoins sind ein zentraler Bestandteil des DeFi Systems (Egorov, 2019, S.1). Beispiele für an den US-Dollar gebundene Stablecoins sind on-Chain kollateralisierte DAI-Token oder off-Chain kollateralisierte USDC- oder USDT-Token (Schär, 2020, S.6).

Laut Egorov stieg durch das Aufkommen einer Vielzahl unterschiedlicher Stablecoins auch der Tauschbedarf über CFMMs. Für inhärent volatile Assets birgt Slippage weniger Probleme als für Assets die preisstabil sein sollten. Hoher Slippage führt zu tiefem Handelsvolumen. Tiefes Handelsvolumen führt zu geringeren Returns aus Handelsgebühren für Liquiditätsprovider, was wiederum die Poolgrösse negativ beeinflusst. Curve verwendet eine Kombination der Constant Sum- und Constant Product Funktion, um geeignete Eigenschaften für Liquiditätspools bestehend aus gegenseitig preisstabilen Assets zu erhalten. (vgl. Egorov, 2019, S.1)

Dazu gehören neben Stablecoins auch Assets mit dem gleichen Underlying dazu, beispielsweise Bitcoin auf der Ethereum Blockchain (renBTC, wBTC, sBTC). Berenzon betitelt diese Implementierung *Hybrid Constant Function Market Maker* (Berenzon, 2020).

Funktionsgleichung

Die folgende Herleitung der Curve-Funktionsgleichung wurde von Egorov (2019, S.4f.) übernommen und dient zur Verdeutlichung der Eigenschaften verschiedener Funktionen von CFMMs. Die Constant Sum Funktion lautet:

$$\sum x_i = D \quad (15)$$

Wobei D die Summe aller Token x_i ist, wenn diese den gleichen Preis haben. Der effektive Preis bei einer Constant Sum Funktion ist konstant und verursacht daher keinen Slippage. Weicht der Aussenmarkt-Preis eines Tokens ab, werden Arbitrageure gegen diesen Kontrakt handeln und die Pool-Reserve eines Tokens vollständig entziehen. Ein Liquiditätspool mit diesem Preismechanismus kann nicht unbegrenzt Liquidität bereitstellen. Aus diesem Grund wird die bereits bekannte Constant Product Funktion benötigt, hier generalisiert für n Token mit gleichen Preisen, benötigt:

$$\prod x_i = \left(\frac{D}{n}\right)^n \quad (16)$$

Um beide Funktionen zu kombinieren, werden die Gleichungen (15) und (16) addiert:

$$\sum x_i + \prod x_i = D + \left(\frac{D}{n}\right)^n \quad (17)$$

Es folgt die Multiplikation der Constant Sum Gleichung mit dem Koeffizienten χ , genannt Leverage. Gilt $\chi = 0$, ergibt sich die Constant Product Funktion. Wird χ erhöht, nähert sich die Gleichung einer Constant Sum Funktion an. χ sollte dimensionslos (d.h unabhängig von der Anzahl Token) sein, weshalb noch mit D^{n-1} multipliziert wird. Daraus ergibt sich:

$$\chi D^{n-1} \sum x_i + \prod x_i = \chi D^n + \left(\frac{D}{n}\right)^n \quad (18)$$

Diese Gleichung gilt nur, wenn alle Token den gleichen Preis haben. Um für Preisänderungen zu korrigieren wird χ dynamisch gemacht:

$$\chi = \frac{A \prod x_i}{(D/n)^n} \quad (19)$$

A ist eine Konstante und der sogenannte Amplifikationskoeffizient. Je höher A gesetzt wird, desto mehr ähnelt die Curve Funktion der Constant Sum Funktion. Bei gleichmässiger Verteilung der Token im Pool gilt: $\chi = A$. Sind die Token ungleichmässig verteilt, weichen die Preise von 1 ab, und der Leverage χ tendiert gegen 0. Die Funktion wird krümmt und nähert sich der Constant Product Funktion an. Der effektive Preis verändert sich und ein Asset kann nicht mehr vollständig aus dem Liquiditätspool entzogen werden. Einsetzen von χ in 18 führt über einige Zwischenschritte zur Curve-Gleichung:

$$An^n \sum x_i + D = ADn^n + \frac{D^{n+1}}{n^n \prod x_i} \quad (20)$$

Ist ein Liquiditätspool anfänglich festgelegt, wird D berechnet. D wird bei einem Tausch konstant gehalten. Die Anzahl der getauschten Token x_i wird so verändert, dass Gleichung 20 erfüllt ist.

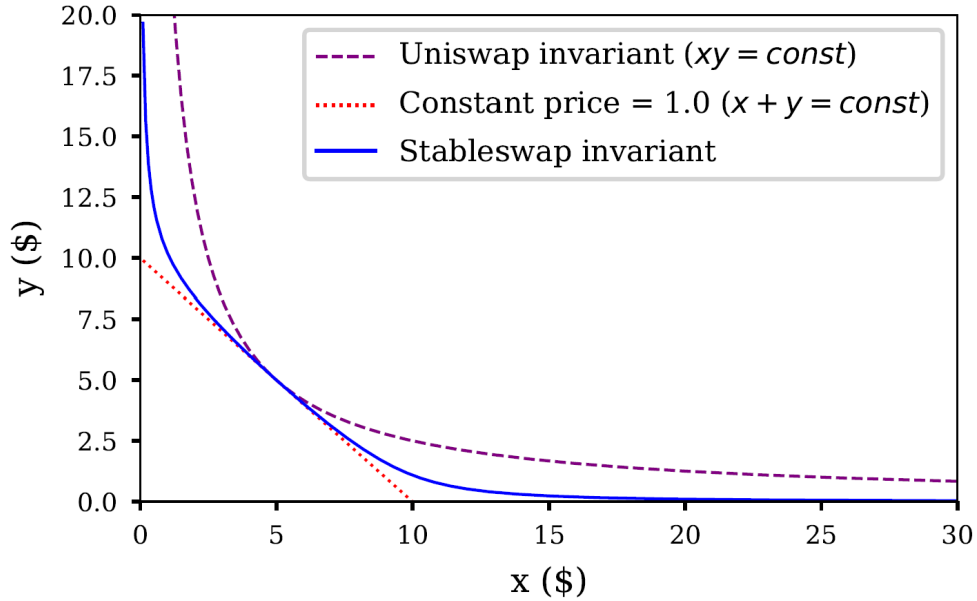


Abbildung 3: Curve Funktiongraph im Vergleich zu Constant Sum- und Constant Product Funktion, Quelle: Egorov (2019, S.3)

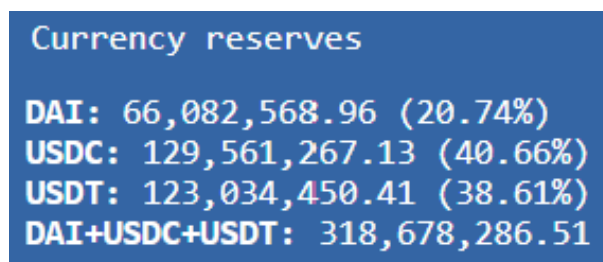
Slippage

Der blaue Graph in Abbildung 3 zeigt die Curve-Funktion. Bei Token-Gleichgewicht ist die Funktion fast linear. In Richtung der Achsen wird die Funktion konvexer. Schwache Krümmung bedeutet tiefer Slippage, da der effektive Preis nicht stark vom Spot-Preis abweicht. Verändert sich das Preisverhältnis im Aussenmarkt, ist aufgrund von Arbitrage die Verteilung der beiden Token im Pool nicht mehr im Gleichgewicht und ein Tausch ist mit Slippage verbunden. Gleichzeitig wird stetige Liquidität sichergestellt. Grundsätzlich sollten die Pool-Assets preisstabil bleiben, wodurch erwartet wird, dass die Verteilung der Token im Pool im Gleichgewicht und der Slippage tief bleibt. (vgl. Egorov, 2019, S.3f.)

Liquidität bereitstellen und Impermanent Loss

Liquiditätsprovider können einen, mehrere oder alle vorhandenen Token eines Curve-Pools hinzufügen oder abziehen. Bei Deposit erhalten Liquiditätsprovider poolspezifische Liquiditätstoken. Wie bei UniSwap und Balancer erhalten Liquiditätsprovider einen proportionalen Anteil der Tauschgebühren. Diese Gebühr liegt momentan in allen Pools bei 0.04%. Zusätzlich gibt es Pools, welche die Liquiditätstoken weiteren DeFi Lending Protokollen, wie Compound oder Aave hinzufügen um zusätzliche Rendite zu erzielen. (vgl. Charlie, 2020)

Abbildung 4 zeigt die Verteilung der Stablecoins im Curve 3Pool:



Currency reserves	
DAI:	66,082,568.96 (20.74%)
USDC:	129,561,267.13 (40.66%)
USDT:	123,034,450.41 (38.61%)
DAI+USDC+USDT:	318,678,286.51

Abbildung 4: Zusammensetzung des Curve 3Pools, Quelle: www.curve.fi/3pool, Zugriff: 10.11.20

Die Reserven im 3Pool sind ungleichmässig verteilt. Curve schafft Anreize diesem Ungleichgewicht entgegenzuwirken. Laut Zhang (2020) ist die Tauschrate vorteilhaft, wenn die Reserven wieder näher an das Gleichgewicht gebracht werden. Gemäss Charlie (2020) erhalten Liquiditätsprovider einen Bonus für das Hinzufügen des Assets mit dem geringsten Anteil, sowie für das Abziehen des Assets mit dem höchsten Anteil. Impermanent Loss ist bei Curve-Pools kein zentrales Problem, da die Assets preisstabil bleiben sollten (Reimi, 2020).

2.4 Bancor

Bancor V1 war zusammen mit UniSwap eine der ersten Implementierungen eines Constant Function Market Makers. Mittlerweile erfolgte mit Bancor V2 ein Protokoll-Update. Bancor selbst bezeichnet die neue Version als *Dynamic Automated Market Maker* (Bancor, 2020).

Wichtigste Neuerung bei Bancor V2 ist die Anpassung der Tokenengewichte über ein Orakel. Dazu werden Orakel von Chainlink benutzt. (vgl. Chainlink, 2020)

Im Zentrum des Protokolls stehen Smart Token, die auf dem ERC-20 Token Standard basieren und über einen integrierten Liquiditätsmechanismus verfügen. Dieser stellt sicher, dass Smart Token immer gegen Connector-Token (beliebige ERC20-Token oder ETH) ausgetauscht werden können. Dazu hat jeder Smart Token eine oder mehrere Verbindungen. Momentan einziger Smart Token ist der Bancor Network Token (BNT). Die Verbindungen bilden Liquiditätspools bestehend aus dem Smart Token und dem Connector-Token. Smart Token können somit als Brücke für den Tausch zwischen verschiedenen Connector-Token genutzt werden. (vgl. Hertzog et al., 2018, S.5)

Hertzog et al. (2018, S.5f.) führen folgende Beispiele für einen Tausch über das Bancor Protokoll auf:

- Smart Token Kauf: Der Käufer sendet den Connector-Token an den Smart Contract, dafür werden neue Smart Token geschaffen und direkt an den Käufer ausgeschüttet. Die Anzahl der Smart Token

im Umlauf sowie die Anzahl Connector-Token in der Reserve erhöht sich.

- Smart Token Verkauf: Der Verkäufer sendet den Smart-Token an den Smart Contract und erhält dafür eine bestimmte Anzahl Connector-Token. Die Anzahl Smart Token im Umlauf, sowie die Reserve des Connector-Tokens sinkt.

Funktionsgleichung

Die Funktionsgleichung für Bancor V2 ist nicht formal dokumentiert. Bancor V1 nutzte eine Constant Reserve Ratio (CRR) Gleichung. Dabei wurde gemäss Rosenfeld (2017, S.1) ein konstantes Verhältnis F zwischen der Reserve des Connector-Tokens R und dem Gesamtwert ($S \cdot P$) des Smart Tokens beibehalten:

$$F = \frac{R}{S \cdot P} \quad (21)$$

S ist die Anzahl und P der Preis eines Smart Token im Umlauf. F wird bei der initialen Konfiguration des Smart Tokens festgelegt und muss zwischen 0 und 1 liegen. Hertzog et al. (2018, S.9) bezeichnen F als Connector-Gewicht. Der gewählte Wert für F hat direkte Implikationen auf die Preisentwicklung des Smart Tokens.

Die von Rosenfeld (2017, S.2) formalisierten Gleichungen für den Kauf (22) und Verkauf (23) eines Smart Token (siehe 5.3) sind nachfolgend dargestellt. T steht für Smart Token, E für Connector Token:

$$T_{erhalten} = S_0 \cdot \left(\left(1 + \frac{E_{gesendet}}{R_0} \right)^F - 1 \right) \quad (22)$$

$$E_{erhalten} = R_0 \cdot \left(\left(1 + \frac{T_{gesendet}}{S_0} \right)^{1/F} - 1 \right) \quad (23)$$

Slippage

Laut Shachav (2020) benutzt Bancor V2 einen Mechanismus zur Liquiditäts-Amplifikation. Die genaue Funktionsweise ist ebenfalls nicht dokumentiert. Vereinfacht gesagt können die Pool-Reserven in einer gewissen Preisregion künstlich erhöht werden, wodurch der Slippage bei einem Tausch verringert werden soll.

Liquidität bereitstellen

Bancor V2 bietet Liquiditätsprovidern die Möglichkeit, ein einzelnes Asset dem Pool hinzuzufügen. Liquiditätsprovider müssen daher nicht mehr zwingend den Smart Token besitzen. Aus diesem Grund wird pro Pool-Asset ein separater Liquiditätstoken geschaffen. Bei einem einseitigen Token-Deposit erhöht sich das Pool-Gewicht des Tokens proportional. Da die Gebühren aber unabhängig von der momentanen Pool Gewichtung auf beide Liquiditätstoken gleichmässig aufgeteilt werden, gibt es den Anreiz für Liquiditätsprovider das Asset mit dem tieferen Gewicht dem Pool hinzuzufügen. (vgl. Shachav, 2020)

Impermanent Loss

In Bancor V2 verändert ein Pool seine internen Gewichte proportional zu allen Liquiditäts- sowie Preisänderungen. Verändert sich der Preis eines Assets auf dem Aussenmarkt, wird dies von den Orakeln erkannt und die Ziel-Gewichte der beiden Token entsprechend angepasst, damit der interne Preis dem Aussenmarkt-Preis entspricht. Die Arbitragemöglichkeit entfällt. Arbitrageure können somit nicht mehr auf Kosten der Liquiditätsprovider dem Pool Wert entziehen und der Impermanent Loss wird eliminiert. Einzig bei Veränderungen der Pool-Verteilungen durch Trades werden Arbitrageure benötigt, um den Pool wieder in die vom Orakel definierten Ziel-Gewichte zu bringen. (vgl. Chainlink, 2020)

3 Diskussion

Bisher wurden die verschiedenen Implementierungen bezüglich Preisfunktion, Slippage, dem Bereitstellen von Liquidität sowie Impermanent Loss dargestellt. In diesem Kapitel werden die wichtigsten Unterschiede in Bezug auf Slippage und Impermanent Loss herausgestrichen und die Auswirkungen auf Händler und Liquiditätsprovider betrachtet. Zum Abschluss der Diskussion wird das Front-Running Problem im Kontext von CFMMs diskutiert.

Protokoll	Funktionsgleichung	Art der Funktion	Token pro Pool
UniSwap	$x \cdot y = k$	Constant Product	2
Balancer	$V = \prod B_i^{w_i}$	Constant Mean	2-8
Curve	$s \cdot \sum x_i + \prod x_i = k$	Constant Sum/-Product	2-4
Bancor	$F = R/SP$	Constant Product	2

Tabelle 1: Übersicht der Protokoll-Funktionen, eigene Darstellung

3.1 Slippage

Im Vergleich der unterschiedlichen CFMMs ist für Nutzer der Slippage neben der Handelsgebühr von grösster Bedeutung. Die folgende Diskussion bezieht sich einzig auf die zugrundeliegenden Preisfunktion. Die Poolgrösse der verschiedenen Protokolle wird ausser Acht gelassen.

Mit der Kombination von Constant Sum- und Constant Product Funktion minimiert Curve den Slippage für einen gewissen Preisbereich der im Pool enthaltenen Assets. Dieser Bereich ist abhängig von der Wahl von A . Verändern sich die Preise der Pool-Token im Aussenmarkt, wird sich die Zusammensetzung des Pool verändern. Wie in Abbildung 3 zu sehen ist, kann die Krümmung und damit der Slippage bei einem Curve-Tausch im Ungleichgewicht höher sein als bei UniSwap.

Abgesehen von der Curve-Funktion wird laut Martinelli (2020a) der Slippage bei einem Tausch zwischen Token mit gleichem Pool-Gewicht minimiert. Abbildung 5 zeigt den Slippage in Abhängigkeit des relativen Gewichts der gehandelten Token.

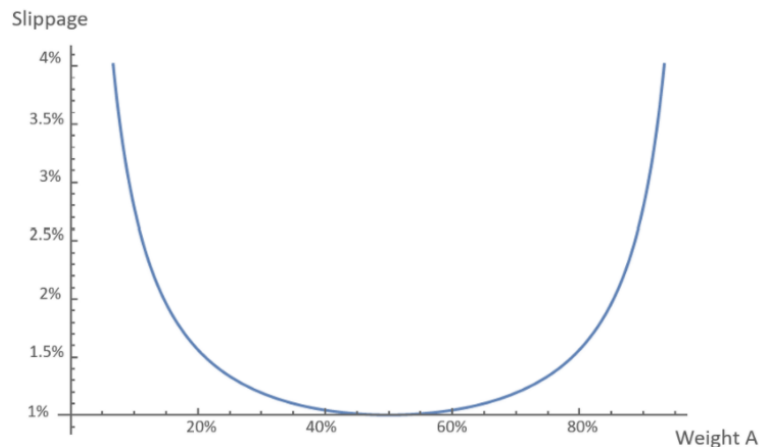


Abbildung 5: Slippage in Abhängigkeit des relativen Token Gewichts, Quelle: Martinelli (2020a)

Beispiel: Pool mit einer Verteilung der Token-Gewichte von $A = 0.4$, $B = 0.4$, $C = 0.1$ und $D = 0.1$. Bei einem Handel von A mit B ist das relative Gewicht 50% und der Slippage tiefer als bei einem Handel zwischen A und C mit einem relativen Token-Gewicht von 80%. (vgl. Martinelli, 2020a) Wenn man von der relativen Grösse eines Trades gegenüber des Pools absieht, lässt sich für Händler zusammenfassen:

- Curve ist die beste Option für den Tausch von Stablecoins, der Slippage ist gering und die Handelsgebühr beläuft sich momentan in allen Pools auf 0.04%.
- Ein Tausch über UniSwap oder zwischen zwei gleichgewichteten Token über Balancer ist in Bezug auf Slippage äquivalent. Bei beiden Protokollen lässt sich der Slippage nicht vollständig eliminieren. UniSwap hat eine Tauschgebühr von 0.3%, während bei Balancer-Pools die Gebühr flexibel festgelegt werden kann. Ein Tausch zwi-

schen zwei ungleichgewichteten Token über Balancer hat den höchsten Slippage.

- Ein Tausch über Bancor V2 sollte gemäss Shachav (2020) aufgrund des Amplifikationsmechanismus niedrigeren Slippage als UniSwap und Balancer verursachen.

Grundsätzlich müssen Nutzer nicht selbst berechnen, über welches Protokoll der Slippage am geringsten ist. Beispielsweise bietet der Aggregator „linch“ einen Algorithmus an, welche die beste Tauschrate über den effizientesten Weg zwischen den führenden DEX-Protokollen findet.³

3.2 Impermanent Loss

Für Liquiditätsprovider ist neben dem Handelsvolumen und zusätzlichen Pool-Anreizen der Impermanent Loss zentral. Abbildung 6 veranschaulicht die Höhe des Impermanent Loss für Pools bestehend aus zwei Assets mit unterschiedlichen Gewichtungen.

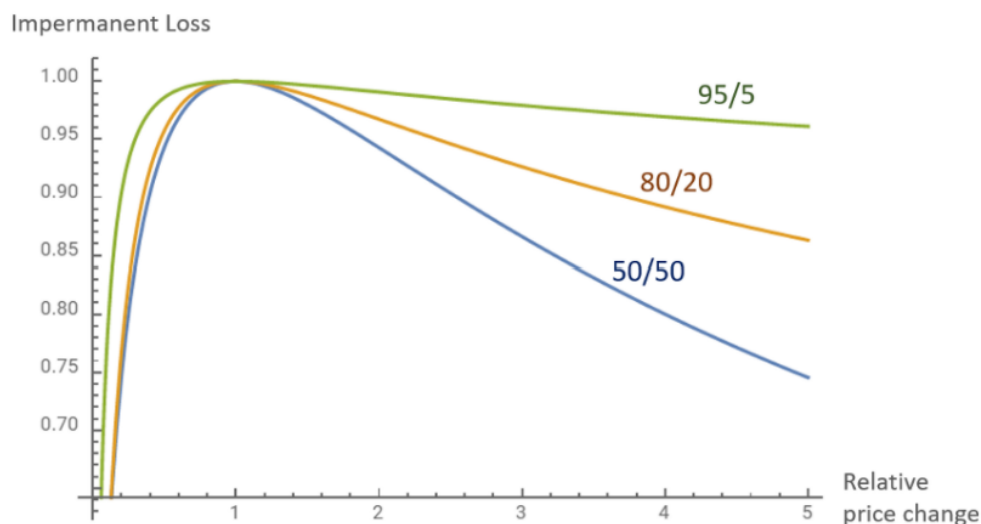


Abbildung 6: Impermanent Loss für verschiedene Pool Gewichte, Quelle: Martinelli (2020a)

³<https://linch.exchange>

Die x-Achse bildet den relativen Preisunterschied der beiden Token ab. Wie bereits in Kapitel 2.1 angesprochen, führt ein einseitiger Preisunterschied immer zu Impermanent Loss bei Abzug von Liquidität. Ein Liquiditätspool mit einer Gewichtung von je 50% maximiert den Impermanent Loss (Martinelli, 2020a).

UniSwap Liquiditätsprovider erfahren den höchsten Impermanent Loss. Bei Balancer Liquiditätspools lassen sich die Token-Gewichte beliebig festlegen. Liquiditätsprovider können damit die Korrelation gegenüber einem Asset erhöhen oder reduzieren. Bei starker Gewichtung behält ein Liquiditätsprovider den Upside einer positiven Preisveränderung und kann gleichzeitig Handelsgebühren aus dem Pool einsammeln. Auch bei einer negativen Preisveränderung ist der Verlust geringer als bei einem 50/50 Pool. Jedoch bedeutet stärkere Gewichtung eines Tokens ebenfalls mehr Slippage, wodurch das Handelsvolumen und damit Gebühren für Liquiditätsprovider sinken. (vgl. Martinelli, 2020a)

Bei Curve Liquiditätspools ist Impermanent Loss nicht von zentraler Bedeutung, da die Assets preisstabil bleiben sollten und der relative Preisunterschied nur gering ist.

Bancor V2 versuchte den Impermanent Loss mittels stetiger Anpassung der Token Gewichte über Orakel zu eliminieren (Chainlink, 2020). Laut Shevchenko (2020) dauert die Aktualisierung über die Orakel zu lange und Front-Running ist möglich. Aus diesem Grund ist Bancor von der ursprünglichen Lösung abgerückt und hat mit Version 2.1 eine Impermanent Loss Versicherung eingeführt.

3.3 Front-Running Problem bei CFMMs

Bisher galt die Annahme, dass die Interaktion der Nutzer mit den Smart Contracts und der Ethereum Blockchain problemlos funktioniert. In der Realität gibt es noch einige Probleme bei DeFi-Applikationen. Eines dieser Probleme ist das „Front Running“.

Front-Running bezeichnet allgemein das Ausnützen einer (privaten) Information, die den Preis eines Assets ändern könnte, für finanziellen Gewinn (Zhou et al., 2020, S.1).

Beispielsweise identifizieren sogenannte Arbitrage Bots profitable Transaktionen aus dem Ethereum-Mempool. Diese Transaktionen werden kopiert und die Ziel-Adresse zu der Eigenen verändert. Über Erhöhung der Transaktionsgebühr wird versucht die eigene Transaktion vor der Original-Transaktion in einem Block zu inkludieren und den Profit selbst in Anspruch zu nehmen. (vgl. Robinson und Konstantopoulos, 2020)

Im Kontext von CFMMs gibt es sogenannte Sandwich-Attacken (Zhou, 2020). Das folgende Beispiel ist angelehnt an Buterin (2018) und zeigt den Ablauf einer Sandwich-Attacke.

1. Ausgangslage: UniSwap Liquiditätspool bestehend aus 10 Einheiten Token A und 10 Einheiten Token B. Ein Nutzer will eine Einheit A gegen B tauschen. Ein Angreifer sieht die pendente Transaktionsnachricht im Mempool und verfasst zwei eigene Transaktionsnachrichten, die er vor und nach der ursprünglichen Transaktion im Block platziert.
2. Front-Run Transaktion: Angreifer sendet eine Einheit A und erhält dafür 0.9091 Einheiten von B.
3. Ursprungstransaktion: Nutzer sendet eine Einheit A, erhält dafür 0.757576 Einheiten B.
4. Back-Run Transaktion: Angreifer sendet 0.7576 Einheiten von B und erhält 1 Einheit A.
5. Profit für den Angreifer: $0.9090 \text{ B} - 0.7575 \text{ B} = 0.1515 \text{ B}$ auf Kosten des Nutzers, der anstatt 0.9090 nur 0.7575 Einheiten von B erhält.

Aktion	Anzahl A	Anzahl B	Konstante	gesendet → erhalten
1.	10	10	100	-
2.	11	9.0909	100	1 A → 0.9090 B
3.	12	8.3333	100	1 A → 0.7575 B
4.	11	9.0909	100	0.7575 B → 1 A

Tabelle 2: Veränderung der Pool Reserven bei Sandwich-Attacke, eigene Darstellung in Anlehnung an Buterin (2018)

Der Angreifer nutzt die (öffentliche) Information über den unbestätigten Trade aus. Er verändert mit der Front-Run Transaktion die Pool-Reserven, der effektive Preis für den Nutzer verschlechtert sich. Mit der Back-Run Transaktion wird der Pool wieder näher an das Gleichgewicht gebracht, wodurch der Angreifer einen besseren effektiven Preis erhält. Er nutzt die Arbitrage-Möglichkeit direkt selber. Ein einfacher Schutz der bereits von UniSwap und anderen CFMMs implementiert wurde, ist die Definition des maximalen Slippage bei Abgabe eines Trades durch den Nutzer. Liegt der Slippage über dem definierten Wert, wird die Transaktion nicht ausgeführt. Diese Möglichkeit sollte aufgrund der anfallenden Transaktionsgebühren für den Angreifer den Anreiz für Sandwich-Attacken merklich senken. Die Diskussion weiterer Lösungsansätze würde den Umfang dieser Arbeit übersteigen.

4 Zusammenfassung

In dieser Arbeit wurden die Constant Function Market Maker-Protokolle UniSwap, Balancer, Curve und Bancor formal verglichen.

Die Einleitung lieferte einen Gesamtüberblick verschiedener Methoden für Krypto-Tauschbörsen. Smart Contract basierte Liquiditätspools wurden anderen DEX-Methoden gegenübergestellt und die jeweiligen Vor- und Nachteile besprochen. Im Hauptteil lag der Fokus auf dem Vergleich der verschiedenen CFMM Implementierungen. Zentrale Begrifflichkeiten wie Slippage und Impermanent Loss wurden anhand von UniSwap eingeführt und für jedes Protokoll einzeln dargestellt. Des Weiteren wurden die Protokoll-Funktionsgleichungen und das Bereitstellen und Abziehen von Liquidität verglichen. Anschliessend folgte ein Effizienz-Abgleich in Bezug auf Slippage und Impermanent Loss. Dabei wurden die Auswirkungen für Händler und Liquiditätsprovider diskutiert. Zum Schluss wurde das Front-Running Problem von CFMMs anhand eines Ablaufbeispiels einer Sandwich-Attacke aufgezeigt.

Krypto-Tauschbörsen können zentral oder dezentral abgewickelt werden. Zentrale Tauschbörsen sind effizient, aber nicht wirklich geeignet im Zusammenhang mit der dezentralen Infrastruktur einer Blockchain. Constant Function Market Makers zeichnen sich gegenüber anderen dezentralen Tausch-Protokollen durch interne Preisfindung und Sicherstellung stetiger Liquidität aus. Allenvoran wird kein Order-Buch benötigt.

An dieser Stelle ist zu erwähnen, dass es sich um Open-Source Protokolle handelt, wodurch sie problemlos kopiert werden können. SushiSwap, eine Fork von UniSwap, hat sich dies zunutze gemacht und gleichzeitig sogenanntes Vampire Mining betrieben, wodurch für UniSwap-Liquiditätsprovider Anreize für den Wechsel auf SushiSwap geschaffen wurden. Resultat war ein signifikanter Liquiditätsverlust für UniSwap Pools. (vgl. Defiant, 2020)

UniSwap verwendet eine Constant Product Funktion für Pools bestehend aus zwei Token. Durch die Konvexität der Funktion ist ein Tausch mit Slippage verbunden. Arbitrage sorgt dafür, dass der Wertanteil (Gewicht) der beiden Token im Pool gleich hoch ist. Ebenso bringt Arbitrage den relativen Token-Preis in Einklang mit dem Aussenmarktpreis. Daraus ergibt sich Impermanent Loss für Liquiditätsprovider.

Auf der Basis von UniSwap erweitert Balancer die Constant Product Funktion für Pools mit bis zu acht verschiedenen Token. Die resultierende Funktion nennt sich Constant Value Funktion und hält das Produkt der Token-Werte während einem Tausch konstant. Die beliebige Gewichtung der Pool-Token erlaubt Liquiditäts Providern die Korrelation gegenüber einem Asset zu erhöhen, womit der Impermanent Loss reduziert wird. Bei einer 50/50 Gewichtung wie bei UniSwap wird der Impermanent Loss maximiert. Smart Pools erweitern die Flexibilität und eröffnen neue Anwendungsmöglichkeiten von Liquiditätspools.

Curve nutzte die Preisstabilität gewisser Kryptoassets und kombiniert die Constant Product- mit der Constant Sum Funktion. Die resultierende Hybrid-Funktion verringert den Slippage für Trades zwischen preisstabilen Assets, wie Stablecoins. Zudem sind Liquiditätsprovider nicht von Impermanent Loss betroffen. Optional gibt es Curve-Pools, welche die Liquiditäts-Token zusätzlichen DeFi Lending Protokollen hinzufügen

um zusätzliche Rendite erzielen.

Im Bancor Protokoll V1 halten Smart Token eine konstante Reserve von verbundenen ETH/ERC20-Token und bilden somit ebenfalls einen Liquiditätspool. Smart Token können stets mit dem zugehörigen Connector-Token gekauft und verkauft werden. Der Smart Token-Preis passt sich an die Nachfrage an. Bancor V2 führte die Anpassung der Token-Gewichte über externe Orakel ein. Die Pool-Anreize sind so definiert, dass die Anzahl der gepoolten Token konstant bleibt und der Impermanent Loss dadurch deutlich verringert wird. Zusätzlich gibt es einen Amplifikationsmechanismus, welcher den Slippage minimieren soll. Die formale Untersuchung dieses Mechanismus war mangels Quellen nicht möglich.

Front-Running und konkret Sandwich-Attacken sind ein bestehendes Problem von CFMMs, können aber gemindert werden. In Zukunft wird vor allem die Sicherheit der Smart Contracts eine zentrale Bedeutung einnehmen. Fehler in der Programmierung können verheerend sein und die weitere Adoption der Protokolle aufhalten. Diese Risiken verstärken sich bei dem Zusammenspiel mehrerer Smart Contracts. Jedoch gibt es bereits zahlreiche Smart Contract Auditing Anbieter. Ebenso ist die weitere Adoption von CFMMs abhängig von den Skalierungsmöglichkeiten der Ethereum Blockchain. Layer 2 Lösungen wie Optimistic Rollups könnten diese Probleme in absehbarer Zeit lösen.

CFMMs sind relativ neue Protokolle und die Weiterentwicklung in vollem Gange. Gleichzeitig gehören sie momentan zu den am weitesten entwickelten und populärsten DeFi-Applikationen. Von den gesamthaft 14.4 Milliarden in DeFi Applikationen gebundenen US-Dollar sind 3.73 Milliarden auf dezentralen Tauschbörsen. UniSwap (1.32 Mia.), Curve (939.1 Mio.), SushiSwap (892,6 Mio.), Balancer (414,3 Mio.) und Bancor (89.1 Mio.) machen gemeinsam 3.655 Milliarden (fast 98%) aus.⁴

Das Netzwerk aus verschiedenen DeFi-Applikationen hat durch stetige Weiterentwicklung die Möglichkeit, ein dezentrales Finanzsystem aufzubauen, bei dem Teilnehmer unabhängig von Intermediären ihre Assets verwalten können. Dezentrale Tauschbörsen in Form von CFMMs sind ein wichtiger Schritt in diese Richtung.

⁴Quelle: <https://defipulse.com/>, Stand: 4.12.2020

5 Anhang

5.1 Balancer (UniSwap) Spot-Preis

Folgende Herleitung wurde von Martinelli and Mushegian (2019) übernommen. Ausgangspunkt sind die bekannten Gleichungen:

$$V = \prod B_t^{w_t}$$

$$EP_i^o = \frac{A_i}{A_o}$$

$$SP_i^o = \lim_{A_i, A_o \rightarrow 0} EP_i^o = \lim_{\Delta B_o, \Delta B_i \rightarrow 0} \frac{\Delta B_i}{-\Delta B_o}$$

Der Grenzwert des effektiven Preis ist per Definition die partielle Ableitung von B_i nach B_o :

$$SP_i^o = -\frac{\delta B_i}{\delta B_o}$$

Aus der Value-Funktion lässt sich B_i isolieren:

$$B_i^{w_i} = \frac{V}{(\prod_{k \neq i, o} B_k^{w_k}) \cdot B_o^{w_o}}$$

$$B_i = \left(\frac{V}{(\prod_{k \neq i, o} B_k^{w_k}) \cdot B_o^{w_o}} \right)^{1/w_i}$$

Es folgt die partielle Ableitung von B_i nach B_o :

$$\begin{aligned} SP_i^o &= -\frac{\delta B_i}{\delta B_o} = -\frac{\delta}{\delta B_o} \left(\left(\frac{V}{(\prod_{k \neq i, o} (B_k)^{w_k}) \cdot (B_o)^{w_o}} \right)^{1/w_i} \right) = \\ &= -\left(\frac{V}{\prod_{k \neq i, o} B_k^{w_k}} \right)^{1/w_i} \cdot \frac{\delta}{\delta B_o} \left(B_o^{-\frac{w_o}{w_i}} \right) = \\ &= -\left(\frac{V}{\prod_{k \neq i, o} B_k^{w_k}} \right)^{1/w_i} \cdot -\frac{w_o}{w_i} \cdot B_o^{-\frac{w_o}{w_i}-1} = \end{aligned}$$

$$\begin{aligned} \left(\frac{V}{\prod_k B_k^{w_k}} \right)^{1/w_i} \cdot B_o^{\frac{w_o}{w_i}} \cdot B_i \cdot \frac{w_o}{w_i} \cdot B_o^{-\frac{w_o}{w_i}-1} = \\ \left(\frac{V}{V} \right)^{\frac{1}{w_i}} \cdot B_o^{\frac{w_o}{w_i}} \cdot B_o^{-\frac{w_o}{w_i}} \cdot \frac{B_i}{w_i} \cdot \frac{w_o}{B_o} = \frac{\frac{B_i}{w_i}}{\frac{B_o}{w_o}} \end{aligned}$$

5.2 Balancer Tausch-Formel

Die Value Funktion vor und nach einem Trade muss gleich sein, daher kann man schreiben:

$$\begin{aligned} \prod_k (B_k)^{w_k} &= \prod_{k \neq i, o} (B_k)^{w_k} \cdot (B_o - A_o)^{w_o} \cdot (B_i + A_i)^{w_i} \\ \prod_{k \neq i, o} (B_k)^{w_k} \cdot B_o^{w_o} \cdot B_i^{w_i} &= \prod_{k \neq i, o} (B_k)^{w_k} \cdot (B_o - A_o)^{w_o} \cdot (B_i + A_i)^{w_i} \\ B_o^{w_o} \cdot B_i^{w_i} &= (B_o - A_o)^{w_o} \cdot (B_i + A_i)^{w_i} \\ B_o - A_o &= \frac{B_i^{\frac{w_i}{w_o}} \cdot B_o}{(B_i + A_i)^{\frac{w_i}{w_o}}} \\ A_o &= B_o \cdot \left(1 - \left(\frac{B_i}{B_i + A_i} \right)^{\frac{w_i}{w_o}} \right) \end{aligned}$$

5.3 Bancor Tausch-Formel

Die Herleitung der folgenden Formeln wurde von Rosenfeld (2017, S.1f.) übernommen. Ausgangspunkt ist die Bancor-Gleichung:

$$R = FSP$$

Nutzer kauft indefinitesimale Anzahl Smart Token dS und bezahlt dafür $P \cdot dS$.

$$PdS = dR = d(FSP) = F(SdP + PdS)$$

$$PdS(1 - F) = FSdP$$

$$PdS\left(\frac{1}{F} - 1\right) = SdP$$

Rosenfeld definiert $\left(\frac{1}{F} - 1\right) = \alpha$

$$PdS\alpha = SdP$$

$$\alpha \frac{dS}{S} = \frac{dP}{P}$$

$$\alpha d\log S = d\log P$$

$$\alpha \log S + A = \log P$$

$$e^A S^\alpha = P$$

$$P = \left(\frac{S}{S_0}\right)^\alpha P_0$$

Damit lässt sich Anzahl Token E , welche ein Nutzer für eine bestimmte Anzahl Smart Token T senden muss, herleiten:

$$\begin{aligned} E &= \int_{S_0}^{S_0+T} PdS = \int_{S_0}^{S_0+T} P_0(S/S_0)^\alpha P_0 = \\ &= P_0 S_0 \frac{(S/S_0)^{\alpha+1}}{\alpha+1} \Big|_{S_0}^{S_0+T} = P_0 S_0 \left(\frac{((S_0+T)/S_0)^{\alpha+1}}{\alpha+1} - \frac{(S_0/S_0)^{\alpha+1}}{\alpha+1} \right) = \\ &= \frac{P_0 S_0}{\alpha+1} \left(\left(1 + \frac{T}{S_0}\right)^{\alpha+1} - 1 \right) = F P_0 S_0 \left(\left(1 + \frac{T}{S_0}\right)^{1/F} - 1 \right) = \\ &= R_0 \left(\left(1 + \frac{T}{S_0}\right)^{1/F} - 1 \right) = R_0 \left(\sqrt[1/F]{1 + \frac{T}{S_0}} - 1 \right) \end{aligned}$$

Literatur

Adams, H. (2020), ‘Uniswap’.

URL: <https://hackmd.io/@HaydenAdams/HJ9jLsfTz>

Angeris, G., Kao, H.-T., Chiang, R., Noyes, C. and Chitra, T. (2019), ‘An analysis of uniswap markets’.

Bancor (2020), ‘Breaking down bancor v2 dynamic automated market makers’, *Medium* .

URL: <https://blog.bancor.network/breaking-down-bancor-v2-dynamic-automated-market-makers-4e90c0f9a04>

Berenzon, D. (2020), ‘Constant function market makers: Defi’s zero to one innovation’, *Medium* .

URL: <https://medium.com/bollinger-investment-group/constant-function-market-makers-defis-zero-to-one-innovation-968f77022159>

Buterin, V. (2018), ‘Improving front running resistance of $x*y=k$ market makers’, *ETHresearch* .

URL: <https://ethresear.ch/t/improving-front-running-resistance-of-x-y-k-market-makers/1281>

Chainlink (2020), ‘How to bring more capital and less risk to automated market maker dexs’, *Chainlink Website* .

URL: <https://blog.chain.link/challenges-in-defi-how-to-bring-more-capital-and-less-risk-to-automated-market-maker-dexs/>

Charlie (2020), ‘Curve.fi, how to get started’, *Medium* .

URL: <https://medium.com/@crypto.tutorials/curve-fi-how-does-it-work-b673a8fe16cc>

Defiant (2020), ‘Sushiswap: What happened, what it means for defi and what’s next.’, *Decrypt* .

URL: <https://decrypt.co/41236/sushiswap-what-happened-what-it-means-for-defi-and-whats-next>

Egorov, M. (2019), ‘Stableswap - efficient mechanism for stablecoin liquidity’.

URL: <https://www.curve.fi/stableswap-paper.pdf>

Hertzog, E., Benartzi, G., Benartzi, G. and Ross, O. (2018), ‘Bancor protocol - continuous liquidity for cryptographic tokens through their smart contracts’.

URL: <https://whitepaper.io/document/52/bancor-whitepaper>

Hoffman, D. (2020), ‘The ultimate guide to balancer smart pools’, *Bankless* .

URL: <https://bankless.substack.com/p/the-ultimate-guide-to-balancer-smart>

John, J. (2020), ‘An introduction to automated market makers’.

URL: <https://cipher.substack.com/p/an-introduction-to-automated-market>

Martinelli, F. (2020a), ‘80/20 balancer pools’, *Medium* .

URL: <https://medium.com/balancer-protocol/80-20-balancer-pools-ad7fed816c8d>

Martinelli, F. (2020b), ‘Calculating value, impermanent loss and slippage for balancer pools’, *Medium* .

URL: <https://medium.com/balancer-protocol/calculating-value-impermanent-loss-and-slippage-for-balancer-pools-4371a21f1a86>

Martinelli, F. and Mushegian, N. (2019), ‘Balancer - a non-custodial portfolio manager, liquidity provider and price sensor’.

URL: <https://balancer.finance/whitepaper/>

Pintail (2019), ‘Uniswap: A good deal for liquidity providers?’, *Medium* .

URL: <https://pintail.medium.com/uniswap-a-good-deal-for-liquidity-providers-104c0b6816f2>

Reimi, M. (2020), ‘Understanding curve: a beginner’s guide and review’, *Holderx* .

URL: <https://holdex.io/x/curve/understanding-curve-a-beginners-guide-and-review>

Rosenfeld, M. (2017), ‘Formulas for bancor system’.

URL: <http://meissereconomics.com/assets/abfe-lesson5-bancor.pdf>

Schär, F. (2020), ‘Decentralized finance: On blockchain- and smart contract-based financial markets’.

URL: <https://dx.doi.org/10.13140/RG.2.2.18469.65764>

Shachav, A. (2020), ‘Balancing mechanics’, *Bancor Network* .

URL: <https://docs.bancor.network/v2-liquidity-pools/balancing-mechanics>

Shevchenko, A. (2020), ‘Bancor updates dex to try a new approach against impermanent loss’, *Cointelegraph* .

URL: <https://cointelegraph.com/news/bancor-updates-dex-to-try-a-new-approach-against-impermanent-loss>

Zhang, D. (2020), ‘Curve.fi 101 — how it works and its meteoric rise’, *Medium* .

URL: <https://medium.com/stably-blog/curve-fi-101-how-it-works-and-its-meteoric-rise-2d6dffd30d51>

Zhou, L. (2020), ‘Demystify the dark forest on ethereum — sandwich attacks.’, *Medium* .

URL: <https://medium.com/coinmonks/demystify-the-dark-forest-on-ethereum-sandwich-attacks-5a3aec9fa33e>

Zhou, L., Qin, K., Torres, C. F., Le, D. V. and Gervais, A. (2020), ‘High-frequency trading on decentralized on-chain exchanges’.

Abbildungsverzeichnis

1	Übersicht Krypto-Tauschprotokolle, Quelle: eigene Darstellung in Anlehnung an Schär (2020, S.8-11)	4
2	Constant Product Funktion, Quelle: Schär (2020, S.10) .	5
3	Curve Funktiongraph im Vergleich zu Constant Sum- und Constant Product Funktion, Quelle: Egorov (2019, S.3) .	14
4	Zusammensetzung des Curve 3Pools, Quelle: www.curve.fi/3pool , Zugriff: 10.11.20	15
5	Slippage in Abhängigkeit des relativen Token Gewichts, Quelle: Martinelli (2020 <i>a</i>)	20
6	Impermanent Loss für verschiedene Pool Gewichte, Quelle: Martinelli (2020 <i>a</i>)	21

Tabellenverzeichnis

1	Übersicht der Protokoll-Funktionen, eigene Darstellung .	19
2	Veränderung der Pool Reserven bei Sandwich-Attacke, eigene Darstellung in Anlehnung an Buterin (2018)	23