Constant-Ellipse Market Maker (CEMM): Overview

Ariah Klages Mundt Steffen Schuldenzucker

Feb 2022

In the constant-ellipse market marker, trading takes place along part of an ellipse curve. Mathematically, the ellipse is represented as a shifted deformation of a circle. This is convenient because it allows us to reason about some of the operations in the much simpler space of a circle centered at the origin, rather than an ellipse. To specify this, we use the parameters in Table 1.

Overview Figure 1 illustrates the process and Table 2 lists the relevant values used during the computation. The leftmost sub-figure in Figure 1 depicts the final shape of the ellipse. The trading curve is constructed based on the current reserve state, marked by an orange point and usually labeled $t = (x, y)$, and various parameters. The slope of the orange tangent corresponds to the current price.[1] Trading takes place along the "lower" part of the ellipse between the two points marked in green, where the ellipse intersects the axes. These points

[1] More in detail, the price of asset x denoted in units of asset y is the negative of the tangent slope.

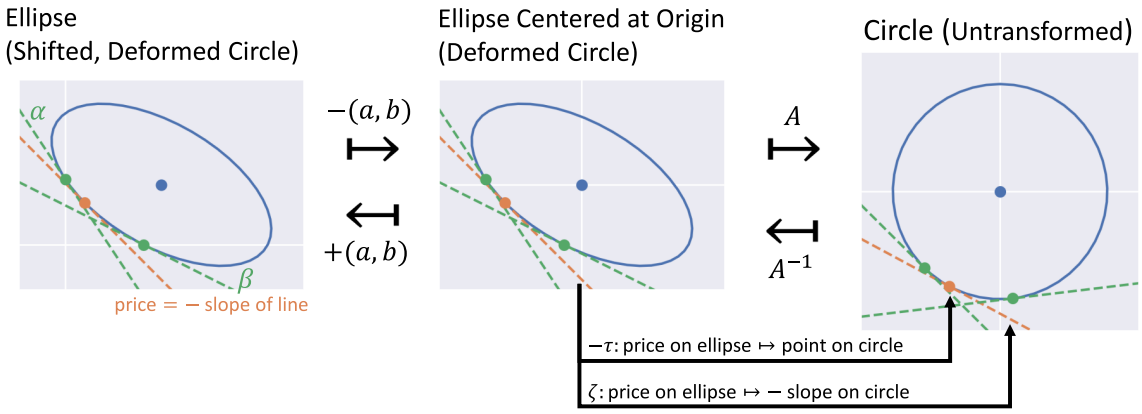

Figure 1: The three stages of the transformation.

Parameter	Description
$\alpha > 0$	Lower bound of the price range
$\beta > \alpha$	Upper bound of the price range
$\lambda \geq 1$	Stretching factor. $\lambda = 1$ implies a circle.
φ	Rotation angle. Kept implicit.
$s, c \geq 0$	Rotation point. Any point with $\|(s, c)\| = 1$ is allowed. We have $c = \cos(-\varphi)$ and $s = \sin(-\varphi)$.

Table 1: Exogenous Parameters for the CEMM

Value	Description
$t = (x, y)$	Current or computed reserve state
(a, b)	Shifting offset, sometimes called "virtual parameters"; midpoint of the shifted ellipse.
x^+, y^+	Intersection points with x and y axis. Also, maximal reserve amount at given invariant value.
r	Radius of the ellipse and also circle. Swap invariant.

Table 2: Relevant values during the computation

are denoted by $(x^+, 0)$ and $(0, y^+)$. The center (a, b) of the ellipse as well as its radius[2] r are chosen dynamically such that (1) the ellipse passes through the reserve state t and (2) the prices at the green points equal given parameters α and β. This defines a finite price range within which trading is possible.

Shifting the ellipse by $-(a, b)$ produces an ellipse centered at the origin, shown in the middle sub-figure. Note that this operation preserves the slope of any tangent. The coordinates x' and y' can be positive or negative.

The centered ellipse can be understood as a circle that has been stretched in x direction by a factor $\lambda \geq 1$ and then rotated to the right by an angle $\varphi \in (-90°, 0]$. The converse operation, rotating to the left and then squishing by $1/\lambda$, transforms the ellipse into a circle, depicted in the rightmost sub-figure.

[2]There are several ways to define the center and diameter of an ellipse. We use the following one: the *center* is the midpoint of the longest diameter, i.e., the longest line that connects two points on the ellipse. This is marked by a blue point in Figure 1. The *radius* is the length of the shortest line that connects the center to any point on the ellipse.

Throughout the documentation, the invariant is either labeled r or L. Note that L is a more standard name for the invariant in the AMM literature.

Transformation Matrix The transformations can be described by the matrices

$$A = \begin{pmatrix} c/\lambda & -s/\lambda \\ s & c \end{pmatrix}, \qquad A^{-1} = \begin{pmatrix} c\lambda & s \\ -s\lambda & c \end{pmatrix}$$

where $s := \sin(-\varphi)$ and $c := \cos(-\varphi)$. Observe that $s, c \geq 0$.

Note that these matrices are simple, analytically known, and they are static in the sense that they only depend on exogenous parameters and not on the current state of the system. The deformation defined by A affects both the position of points and the slopes of tangents. Again, the coordinates can be negative.

Translation Functions In addition to these operations, we use two functions to map prices on the ellipse to the circle. Specifically, if p_x is a price on the ellipse and e_x and e_y are the unit vectors, then

$$\zeta : \mathbb{R} \to \mathbb{R} \qquad \text{where} \qquad \zeta(p_x) := -\frac{e_y \cdot A(-1, p_x)^T}{e_x \cdot A(-1, p_x)^T}$$

is the corresponding price (i.e., negative slope) on the circle. This price can be positive or negative. If p_x^c is any price on the circle, then

$$\eta : \mathbb{R} \to \mathbb{R}^2 \qquad \text{where} \qquad \eta(p_x^c) := \frac{1}{\sqrt{1 + (p_x^c)^2}} \begin{pmatrix} p_x^c \\ 1 \end{pmatrix}$$

is the negative of the point on the circle at which this price is present. We combine these functions to receive, given a price p_x on the ellipse, the negative point on the circle

$$\tau : \mathbb{R} \to \mathbb{R}^2 \qquad \text{where} \qquad \tau(p_x) := \eta(\zeta(p_x))$$

that is the transformed image of the point with price p_x.

Shifting Vector, Intersection Points, Invariant Equation We can compute the shifting vector and intersection points as

$$(a, -y^{+\prime}) := rA^{-1}\tau(\beta)$$
$$(-x^{+\prime}, b) := rA^{-1}\tau(\alpha).$$

and we have

$$x^+ = x^{+\prime} + a$$
$$y^+ = y^{+\prime} + b.$$

Note that these values are linear in r and otherwise only depend on the parameters.

Overall, the trading curve of the CEMM follows the defining equation of the circle after both transformations, i.e.,

$$(e_x \cdot A(t - (a, b)))^2 + (e_y \cdot A(t - (a, b)))^2 = r^2.$$

Calculating the Invariant r We can solve these equations for x, y, and r, respectively, to calculate the invariant r and to execute swaps. Given $t = (x, y)$, we can calculate the invariant r as follows:

$$r = \frac{At \cdot A\chi + \sqrt{(At \cdot A\chi)^2 - (A\chi \cdot A\chi - 1)\, At \cdot At}}{A\chi \cdot A\chi - 1},$$

where $\chi := (e_x \cdot A^{-1}\tau(\beta),\, e_y \cdot A^{-1}\tau(\alpha))$. Note that this formula contains both (scalar) products of vectors and products of real numbers.

Swap Execution When r and one of x or y is given, we can compute the other reserve via

$$y = \frac{-sc\underline{\lambda}x' - \sqrt{s^2c^2\underline{\lambda}^2x'^2 - (1 - \underline{\lambda}s^2)\left[(1 - \underline{\lambda}c^2)x'^2 - r^2\right]}}{1 - \underline{\lambda}s^2} + b$$
$$x = \frac{-sc\underline{\lambda}y' - \sqrt{s^2c^2\underline{\lambda}^2y'^2 - (1 - \underline{\lambda}c^2)\left[(1 - \underline{\lambda}s^2)y'^2 - r^2\right]}}{1 - \underline{\lambda}c^2} + a,$$

where $\underline{\lambda} := 1 - 1/\lambda^2$, $x' := x - a$, and $y' := y - b$. We use this to execute swaps.

Implementation Table 3 lists the most important functions that implement the above calculations. They are all defined in the file `GyroCEMMMath.sol`.

To save computational resources, we store the values of $\tau(\alpha)$ and $\tau(\beta)$ so that they only need to be computed once. They are passed to the functions that need them in a `DerivedParams` structure. To further save resources and

Function	Purpose
`calculateInvariant`	(Re-)computes the invariant r from the current reserves $t = (x, y)$
`calcOutGivenIn`	Computes the amount that leaves the pool when a certain amount enters it, after fees.
`calcInGivenOut`	Computes the amount that needs to enter the pool when a certain amount should leave it, after fees.
`calcXGivenY`	Computes the amount of asset x given a certain amount of asset y and a certain invariant. Used by `calcOutGivenIn` and `calcInGivenOut`.
`calcYGivenX`	The same functionality for y given x.
`liquidityInvariantUpdate`	New invariant when liquidity is added/removed in a "balanced" fashion (without affecting the price). This avoids fully re-calculation of the invariant.

Table 3: Most important functions

to avoid numerical inaccuracies relating to square root calculations, we do *not* compute these values in the contract; rather, they are passed as inputs to the contstructor and then only *validated* on-chain.