Mirage Protocol

@moodyCoins

2023

Abstract

Mirage Protocol is a decentralized perpetuals exchange written in Move. It is powered by a suite of DeFi smart contract modules, connected through a global debt management system. This paper details the decisions behind the design of Mirage Protocol and gives a basic explanation of the underlying mathematics.

1 Introduction

By and large the primary drivers of DeFi so far have been trading, lending, and asset creation (Dai, sETH, etc). Mirage Protocol leverages these three facets to facilitate a completely decentralized, slippage-free, gas-efficient perpetuals exchange.

At its core, Mirage Protocol is a debt management system resting on top of a group of collateralized debt pools. Each module is given a different responsibility in managing the debt, and together they form the entirety of the protocol. The fundamental goals of the protocol are the following:

- Afford users a fully fledged perpetuals experience
- Maintain over-collateralization in all instances
- Exchange profitability

Therefore, much thought must be given to the specifics of debt management, and especially to the details of the implementation of the perpetuals exchange itself. It is important that the exchange have extremely modest and reasonable fees for the majority of its lifespan, while using pressure from various system parameters to enforce system balance in other instances.

2 Mirage Assets

The total net existence of Mirage Assets are essentially all of Mirage Protocol's debt. They are created only by depositing collateral into a Vault (Sec. 4), and are always kept over-collateralized. Creation of these assets is equivalent to the creation of new protocol debt, whereas the reverse is true of burning them. Mirage Assets are also the underlying margin of all Market trades. The ability to create and destroy Mirage Assets is how the protocol leverages it's decentralized exchange (Sec. 5).

2.1 True Interest Rate

A small amount of interest is taken at a flat rate on all debt, but this is within the Vault contract, and only alters the rebase V (Sec. 4). Other parts of the protocol are creating and destroying debt all the time. Let's call the percent rate of debt creation or destruction R_{global}, such that every year this much is created or destroyed depending on the sign. Therefore, if a vault if the protocol is charging an interest rate of R_V, the effective "true" interest rate is:

$$\text{true interest rate} := R_V + R_{\text{global}} \tag{1}$$

The goal of Mirage protocol is to keep $R_{\text{global}} < -R_V$ on long time scales. There are multiple ways the protocol can tweak parameters to bring the system back into alignment. This is essentially a statement about the protocol's revenue overall.

3 Global Debt Management

Protocol debt is managed globally at the highest level per-asset. Global debt is represented by a rebase number. Given two numbers $E, B > 0$, a rebase number R is essentially two sets of numbers: $\{x_e < E\}$ and $\{x_b < B\}$, and transformation functions $e_R(x) : x_b \to x_e$ and $b_R(x) : x_e \to x_b$ between elements of each set.

More specifically, we define global debt D as a rebase:

$$D_{\text{global}} := (E_{\text{global}}, B_{\text{global}}) \tag{2}$$
$$E_{\text{global}} := \text{all global debt} \tag{3}$$
$$B_{\text{global}} := \text{ownership of debt} \tag{4}$$

This is the singular global debt store that rests at the highest level. Contracts are given pieces of B_{global} to represent that contract's ownership of global debt. Modifying global debt is the function of various contracts throughout the protocol. Contracts like Vault and Market will be able to modify the E_{global} and

B_{global}. Since contracts own pieces of B_{global}, it is possible for one contract to indirectly modify the debt of another.

To be precise, various functions that modify the global rebase, $(\hat{F}, \hat{G}, \hat{H}, \ldots)$, are distributed to other contracts. For example, when a user creates more debt by borrowing some amount x of an asset, the new global debt D' is:

$$D'_{\text{global}} = \hat{G}(D_{\text{global}}, x) \tag{5}$$

Where $\hat{G} : R \times \mathbb{Z} \to R'$ (Eq. 53) is a function that modifies a rebase and adds new debt x to the global debt, distributing newly created ownership shares to the Vault.

An in depth discussion of rebase numbers and the transformation functions $\hat{F}$ and $\hat{G}$ is provided in appendix 5.5. The important thing to note is that access to modify global protocol debt is passed down to various distinct modules.

4 Vaults

Vaults are responsible for the minting of of Mirage Assets. Liquidity providers deposit collateral and borrow Mirage Assets up to a minimum collateralization ratio of the Vault. Vaults hold the underlying debt of the entire protocol.

To be precise, rebase functions $\hat{G}$ and $\hat{H}$ (Eq. 53) are given to Vaults to modify global debt.

4.1 Debt Management

Debt is represented in the Vault by a specific debt rebase number:

$$\begin{align} V &:= (E_V, B_V) \tag{6}\\ E_V &:= \text{global debt ownership shares} \tag{7}\\ B_V &:= \text{vault debt ownership shares} \tag{8}\end{align}$$

When a user deposits into a debt, they are receiving some piece of B_V, which is convertible to an amount of global debt ownership shares by way of Eq. 36, which defines the transformation function $e_V : B_V \to E_V$.

We can use this result to calculate the total debt owed by the user using the global debt rebase D_{global}, which defines a similar function e_D. Therefore, if a user has some amount $x \in B_V$, we can convert this into the actual amount of debt through:

$$\begin{align} \text{debt}(x \in B_V) &= e_D(e_V(x)) \tag{9}\\ &= \frac{e_V(x) \cdot E_{\text{global}}}{B_{\text{global}}} \tag{10}\\ &= \frac{\frac{x \cdot E_V}{B_V} \cdot E_{\text{global}}}{B_{\text{global}}} \tag{11}\\ &= \frac{x \cdot E_V \cdot E_{\text{global}}}{B_V \cdot B_{\text{global}}} \tag{12}\end{align}$$

This is the tangible amount of debt owed by the user, given their shares of the Vault debt.

When a user wants to borrow an asset, we use the function $\hat{G} : R \times \mathbb{Z} \to R$ to add new debt x to V:

$$V' = \hat{G}(V, e_{\hat{G}(D_{\text{global}}, x)}(x)) \tag{13}$$

The rest of the vault operations are calculated accordingly. Note that the value $e_{\hat{G}(D,x)}(x)$ is essentially just calculating the amount of base created from the $\hat{G}(D, x)$ operation.

5 Markets

A Mirage Market is a completely decentralized perpetuals exchange that allows trades to be opened at the current oracle price. Many of the concepts of a Mirage Market are different from traditional perpetuals exchange, but the overall functionality is similar for the end user.

5.1 Debt

Instead of being given $\hat{G}$ and $\hat{H}$ like the Vault, the market is given the rebase functions $\hat{F}_e$ and $\hat{F}_b$. These allow global debt modification from actions taken in the market contract. Trader payouts and losses, trading fees, and liquidations are all processed to the global debt state through $\hat{F}$.

5.2 Oracles

Mirage market uses oracles prices, fetched real-time during the contract call, to determine opening and closing prices of trades. Prices are checked against a moving average and confidence interval to ensure they are reliable. Prices are cached and in the instance of an emergency from the oracle the market participants can still close at the last cached rate.

5.3 Parameters

5.3.1 Open Interest and Skew

One of the most significant parameters used by a market is its imbalance between long and short open interest. Let's call the long open interest and short open interest OI_{long} and OI_{short} respectively, and define:

$$\mathit{skew} := OI_{\text{long}} - OI_{\text{short}} \tag{14}$$

The absolute value of *skew* is generally capped by a market defined $\mathit{skew}_{\max}$ after which markets generally only allow new orders that reduce *skew*. *skew* is an important variable and is used to determine the "net protocol leverage" which we will call L_{global}. We can calculate this as:

$$L_{\text{global}} = (\sum_{\text{markets}} \mathit{skew})/\text{total debt} \tag{15}$$

We never want this value to be large, let's just say for argument we want $L < 1$ or $\sum \mathit{skew} <$ total debt. That roughly translates to the following system parameter condition:

$$\sum \mathit{skew}_{\max} < \text{total debt} \tag{16}$$

5.3.2 Funding

Funding is calculated from the previously mentioned *skew*. The funding rate, r, has its absolute value range-bound by a market defined $r_{min} > 0$ and $r_{\max} > 0$. Funding can either be positive or negative.

When $\mathit{skew} = 0$ we want $r = r_{min}$. This is positive, which means the default state of the market is longs paying the minimum funding, which will usually be small.

The purpose of funding in the mirage market is to influence the market to reduce the *skew*, by placing increasing pressure on the skewed positions.

Let's define the "skew-factor" S as:

$$S := \frac{\mathit{skew}_{\max} \cdot \operatorname{sgn}(\mathit{skew})}{\mathit{skew}_{\max} - |\mathit{skew}|} \tag{17}$$

Then we define funding as:

$$r := \begin{cases} r_{\min} \cdot S, & |r_{\min} \cdot S| < r_{\max} \\ r_{\max} \cdot \operatorname{sgn}(S), & \text{else} \end{cases} \tag{18}$$

Note that there exists a skew' where $r_{\min} \cdot S' = r_{\max}$. We can calculate this useful value:

$$r_{\min} \cdot \frac{\mathit{skew}_{\max}}{\mathit{skew}_{\max} - |\mathit{skew}'|} = r_{\max} \tag{19}$$

$$\Rightarrow \ |\mathit{skew}'| = \mathit{skew}_{\max} \cdot (1 - \frac{r_{\min}}{r_{\max}}) \tag{20}$$

The skew-factor S is an important value and is discussed more in the Appendix. It is important to note that the way funding is collected in a mirage market is fundamentally different than how it is collected on a centralized exchange. On a centralized exchange, a user's funding payment P_{cex} is calculated with their open interest OI as:

$$P_{\text{cex}} = r \cdot \text{value}(OI) \tag{21}$$

Let's call the user's margin M. If a user has a position leveraged at ten times their margin, the net funding payment against their margin, will be $10 \cdot r \cdot \text{value}(M)$, since their leveraged position value essentially incurs ten times the funding rate. Therefore when accounting for a leverage L, we can say funding is:

$$P_{\text{cex}} = L \cdot r \cdot \text{value}(M) \tag{22}$$

In a mirage market, funding is taken at a globally fixed rate against the total margin in a market, based off of the global leverage of that side of the market, namely $L^{\text{long}}_{\text{global}}$ and $L^{\text{short}}_{\text{global}}$. Let's say we have a funding rate $r > 0$ such that longs will be paying funding. Then we can calculate the global payment by all longs as:

$$P_{\text{global}} = L^{\text{long}}_{\text{global}} \cdot r \cdot \text{value}(M^{\text{long}}_{\text{global}}) \tag{23}$$

Noting that the leverage of a market can be calculated as:

$$L^{\text{long}}_{\text{global}} = \frac{\text{value}(OI^{\text{long}}_{\text{global}})}{\text{value}(M^{\text{long}}_{\text{global}})} \tag{24}$$

Therefore the funding payment owed by all longs is:

$$P_{\text{global}} = \text{value}(OI^{\text{long}}_{\text{global}}) \cdot r \tag{25}$$

Therefore if a user is long with a positioned that is leveraged 10 times its margin, but the global leverage is only 2, then the user is only responsible for a funding payment of $2 \cdot r$ instead of $10 \cdot r$. The same is true in the reverse scenario. What this means is that **positions in a Mirage Market pay a fixed percentage funding rate against their margin**, irrelevant of position size.

5.3.3 Fees

Fees are calculated in a similar way to funding. A market defines assigns taker or maker fees to trades depending on their impact on the skew. In addition, markets bound both maker and taker fees to the ranges $(m_{\min}, m_{\max})$ and $(t_{\min}, t_{\max})$ Let us say a user wants to open a trade on the long side with open interest T. The skew then becomes:

$$\mathit{skew}' = (OI_{\text{long}} + T) - OI_{\text{short}} \tag{26}$$

Using this value, we define the ratio to the max as $\sigma' = \mathit{skew}'/\mathit{skew}_{max}$. Then we define the fee, f, for this trade as:

$$f := \begin{cases} m_{\max}, & \mathit{skew}' < 0 \text{ and } |m_{\max}/\sigma'| > m_{\max} \\ |m_{\min}/\sigma'|, & \mathit{skew}' < 0 \text{ and } |m_{\max} \cdot \sigma'| \leq m_{\max} \\ t_{\max}, & \mathit{skew}' > 0 \text{ and } |t_{\min} \cdot S'| > t_{\max} \\ |t_{\min} \cdot S'|, & \mathit{skew}' > 0 \text{ and } |t_{\max} \cdot S'| \leq t_{\max} \end{cases} \tag{27}$$

The fee's for various other situations are calculated accordingly.

5.4 Liquidations

Liquidations are processed based off of the parsed oracle price. Let's define a value called the "margin scalar" θ.

$$\theta = \frac{\text{value(margin)} - \text{maintenance margin}}{\text{position size}} \tag{28}$$

Where position size is positive and negative for long and short positions respectively. The maintenance margin, is a fixed percentage for each market and is calculated upon opening a position. Each market defines a base margin rate B, which gives us:

$$\text{maintenance margin} = B \cdot \text{value(position size)} \tag{29}$$

When a position is open we can calculate the liquidation price as:

$$\text{liquidation price} = \text{opening price} + \theta \tag{30}$$

If the market is within a position's liquidation, any account can call to the module and liquidate that position, receiving some percent fee.

5.5 Limit Orders and Order Triggers

Limit orders, take profits, stop losses, and liquidations are all processed by external keepers running open source software. Events on chain allow anyone to index market data.

Appendix

Funding Discussion

Skew Factor

It is useful to examine the skew factor S (Eq. 17) to get a better understanding of funding. Note that S becomes exponential as $skew \to skew_{\max}$. Markets want to configure the ratio $\frac{r_{\min}}{r_{\max}}$ such that $skew'$ falls somewhere after the point at which S stops estimating a linear function. We can take the derivative to see:

$$\frac{dS}{d|skew|} = \frac{skew_{\max}}{(skew_{\max} - |skew|)^2} \tag{31}$$

We can estimate how much we want our funding to be affected by the exponential nature of S. E.g. we can say we are interested in some self-defined inflection point where $\frac{dS}{d|skew|} = I > 0$:

$$\frac{skew_{\max}}{(skew_{\max} - |skew''|)^2} = I \tag{32}$$

$$\Rightarrow \ |skew''| = \frac{skew_{\max} \cdot \sqrt{I} - \sqrt{skew_{\max}}}{\sqrt{I}} \tag{33}$$

Therefore we can get a decent set of parameters for $\frac{r_{\min}}{r_{\max}}$ depending on the desired inflection point. Say we want to make sure max funding is reached at I. Using Eq. 20:

$$skew_{\max} \cdot (1 - \frac{r_{\min}}{r_{\max}}) = \frac{skew_{\max} \cdot \sqrt{I} - \sqrt{skew_{\max}}}{\sqrt{I}}$$

$$\Rightarrow \ \frac{r_{\min}}{r_{\max}} = \frac{1}{\sqrt{I \cdot skew_{\max}}} \tag{34}$$

we can also see from Eq. 20 that if $r_{\min} = 0$, then our funding curve is the entire range of the skew-factor S. All of these tools allows markets to tailor a very specific funding curve to that market's needs.

Rebase Numbers

Let's first define two sets given $E > 0$ and $B > 0$:

$$\mathcal{E}(E) := \{x_e \mid 0 < x_e \leq E\} \tag{35}$$

$$\mathcal{B}(B) := \{x_b \mid 0 < x_b \leq B\} \tag{36}$$

Now let's define a rebase number R:

$$R := (E, B) \tag{37}$$

A rebase defines two conversion functions between the sets $\mathcal{E}$ and $\mathcal{B}$

$$b_R : \mathcal{E} \to \mathcal{B} \tag{38}$$

$$e_R : \mathcal{B} \to \mathcal{E} \tag{39}$$

We define these functions as follows:

$$b_R(x_e \in \mathcal{E}) := \frac{x_e \cdot B}{E} = y_b \in \mathcal{B} \tag{40}$$

$$e_R(y_b \in \mathcal{B}) := \frac{y_b \cdot E}{B} = x_e \in \mathcal{E} \tag{41}$$

These functions take a number in the base or elastic space respectively. Here are a few important properties:

$$b_R(E) = B \tag{42}$$

$$e_R(B) = E \tag{43}$$

$$b_R(e_R(y_b)) = y_b \tag{44}$$

$$e_R(b_R(x_e)) = x_e \tag{45}$$

Let's define the set of all rebase numbers as $\mathcal{R}$ and examine functions that modify a rebase number R. For example the functions $\hat{F}_{e/b} : \mathcal{R} \times \mathbb{Z} \to \mathcal{R}$:

$$\hat{F}_e(R, x_e) := (E + x_e, B) = R' \tag{46}$$

$$\hat{F}_b(R, x_b) := (E, B + x_b) = R' \tag{47}$$

Note that the new rebase R' will have either a new set $\mathcal{E}'$ or $\mathcal{B}'$. To begin to see the usefulness of these numbers let's consider a base part $x_b \in \mathcal{B}$ and a rebase R. We know that the elastic representation of x_b is:

$$y_e = e_R(x_b) = \frac{x_b \cdot E}{B} \in \mathcal{E} \tag{48}$$

Now let's consider what happens when we modify R by adding a fixed elastic amount $c_e > 0$.

$$R' = \hat{F}_e(R, c_e) \tag{49}$$

$$\Rightarrow y'_e = e_{R'}(x_b) = \frac{x_b \cdot (E + c_e)}{B} \tag{50}$$

$$\tag{51}$$

This gives us the immediate important result that:

$$y'_e > y_e \tag{52}$$

What this means is that by altering R, the overall "elastic value" of our initial base part x_b has changed. This concept is useful for ownership. We can roughly view E as the total "owned" amount, and B as the total shares of ownership. We can define functions that modify R like:

$$\hat{G}(R, x_e) := \hat{F}_b(\hat{F}_e(R, x_e), b_R(x_e)) \tag{53}$$

$$\hat{H}(R, x_b) := \hat{F}_e(\hat{F}_b(R, x_b), e_R(x_b)) \tag{54}$$

These two functions are responsible for adding x_b or x_e to R while keeping the ratio E/B the same. For x_e, this is achieved by converting the given x_e into base using b_R, and then adding both x_e and $b_R(x_e)$ into R. We can see that after this kind of operation, the ratio stays the same:

$$e_{\hat{G}(R,x_e)}(y_b) = \frac{y_b \cdot (E + x_e)}{B + b_R(x_e)} \tag{55}$$

$$= \frac{y_b \cdot (E + x_e)}{B + \frac{x_e \cdot B}{E}} \tag{56}$$

$$= y_b \cdot \frac{E}{B} \cdot \frac{1 + \frac{x_e}{E}}{1 + \frac{x_e}{E}} \tag{57}$$

$$= y_b \cdot \frac{E}{B} \tag{58}$$

$$= e_R(y_b) \tag{59}$$

Therefore the functions G and H are such that they keep the conversion functions constant, or in other words:

$$e_{\hat{G}(R,x_e)}(y_b) = e_R(y_b) \quad \forall x_e \tag{60}$$

$$b_{\hat{H}(R,x_b)}(y_e) = b_R(y_e) \quad \forall x_b \tag{61}$$