MARKET-MAKERS AUTOMATIZADOS

Iker Caballero Bragagnini

Tabla de contenido

Las funciones constantes

- Los *market makers* automáticos o *automated market makers* (AMMs) son parte del ecosistema de finanzas descentralizadas que permite que activos digitales sean comerciados de una manera automática y sin necesidad de permisos, a través de fondos de liquidez y no de compradores y vendedores

    - Los AMMs usan fondos de liquidez, donde los usuarios pueden depositar criptomonedas para proporcionar liquidez. Estos fondos usan algoritmos para fijar los precios de los tokens basado en la *ratio* de activos en el fondo

        - Cuando un usuario quiere comerciar, se cambia un token por otro directamente desde el AMM, con los precios determinados por el algoritmo del fondo

        - Los AMMs proporcionan liquidez continua para un gran rango de activos, facilitando así el comercio de criptomonedas menos populares. Además, cualquiera puede proporcionar liquidez a la AMM y participar en el comercio de tokens por menores comisiones (comparado con bolsas tradicionales) y no utilizan ningún intermediario centralizado

        - Algunos de los AMMs más famosos son Uniswap, Sushiswap y Balancer

        - Los AMMs principalmente facilitan el comercio de criptomoneda a criptomoneda. Para poder comerciar con moneda fiat, los usuarios normalmente necesitan ir a una bolsa centralizada u otros servicios antes de interactuar con un AMM

    - Los AMMs son contratos inteligentes que permiten que activos digitales sean comerciados sin necesitar permiso y de manera automática, usando fondos de liquidez en vez de compradores y vendedores

        - En una plataforma de bolsa tradicional, los compradores y los vendedores ofrecen un activo a diferentes precios. Cuando otros usuarios encuentran que el precio cotizado es aceptable, entonces ejecutan una transacción y ese precio se vuelve el precio de mercado

        - No obstante, los AMMs utilizan otro enfoque para comerciar con criptomonedas. Los AMMs son una herramienta financiera única de Ethereum y de las finanzas descentralizadas, de modo que nadie controla el sistema y cualquiera puede crear soluciones y participar

        - Los comerciantes pueden comerciar contra ese contrato inteligente en vez de comerciar directamente con una contraparte como en las bolsas con *order books*

    - Antes de que existieran los AMMs, la liquidez era un reto para las bolsas descentralizadas (DEX) en Ethereum: era una tecnología nueva con una interfaz complicada, de modo que el número de compradores y vendedores era bajo y costaba encontrar personas que comerciaran regularmente

        - En las plataformas AMM, en vez de comerciar entre compradores y vendedores, los usuarios comercian contra un fondo de tokens, un fondo de liquidez (un contrato inteligente). En su núcleo, un fondo de liquidez es un fondo compartido de tokens en donde los usuarios proporcionan los tokens y el precio de estos se determina por una fórmula matemática

        - Es a través de la modificación de la fórmula que los fondos de liquidez se pueden optimizar para diferentes propósitos

        - Cualquiera con conexión a internet y en posesión de cualquier tipo ERC-20 de tokens se puede volver un proveedor de liquidez a través de proporcionar tokens a un fondo de liquidez de un AMM. Los proveedores reciben normalmente una comisión por proporcionar tokens al fondo, y estas comisiones se pagan por los comerciantes que interactúan con el fondo

        - Actualmente, los proveedores de liquidez pueden ganar rendimientos en la forma de tokens de proyecto a través del *yield farming*, que consiste en apostar o bloquear criptomonedas dentro de un protocolo *blockchain* para generar recompensas tokenizadas

    - Los AMMs se han convertido en una manera principal de comerciar con activos en un ecosistema de finanzas descentralizadas, y el elemento más importante es una fórmula matemática simple que puede tomar muchas formas

$$f\big(x(p), y(p)\big) = k$$

        - En este caso, el balance de cada uno de los tokens $x(p)$ (para un token X) e $y(p)$ (para un token Y) es una función de su precio (dado que solo se puede comerciar con estas monedas), y k es una constante

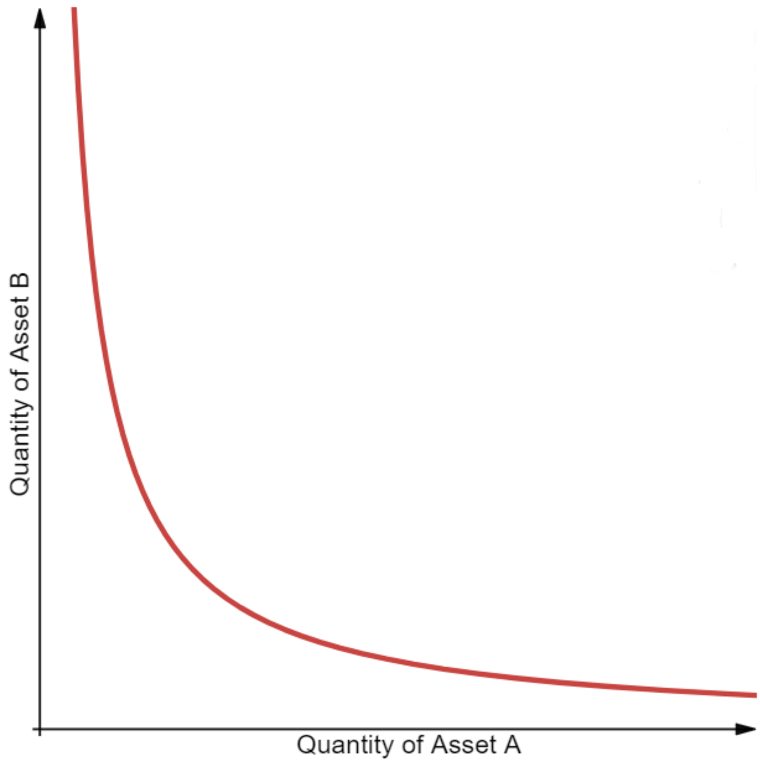
        - La presencia de la constante, representada por k, significa que hay un balance constante de activos que determina el precio de los tokens en el fondo de liquidez. Por ejemplo, si un AMM tiene ETH y BTC, toda vez que se compre ETH, el precio de ETH sube porque ahora hay menos ETH en el fondo, mientras que el precio de BTC es mayor porque quedan más

- El fondo se queda en un balance constante a través de esta afectación de los precios, donde el valor total de ETH siempre será igual al valor total de BTC en el fondo. El fondo solo aumenta de tamaño cuando nuevos proveedores se unen

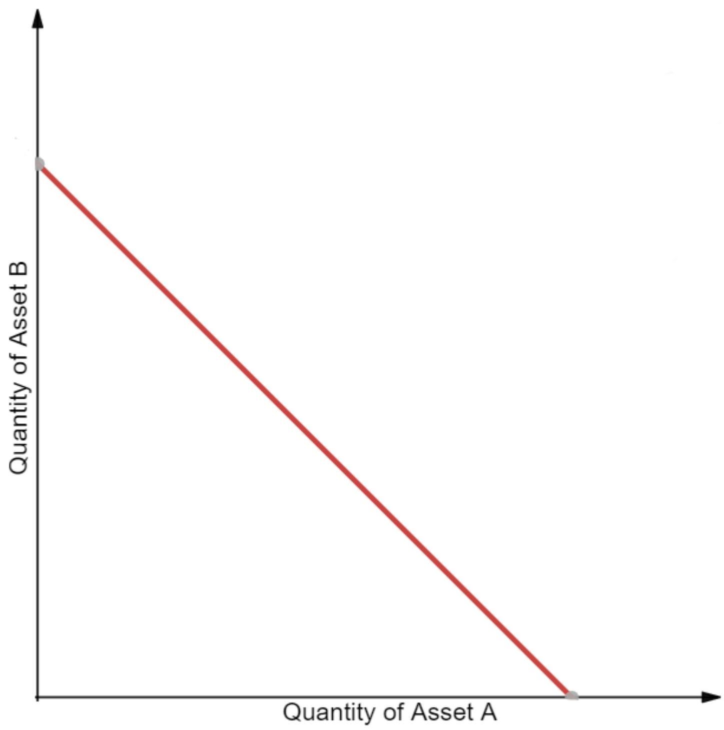
- No importa qué tan volátiles sean los precios, dado que eventualmente habrá un retorno al estado de balance que refleja de manera relativamente precisa el precio de mercado. De no ser así, el modelo incentiva a que los *traders* se aprovechen del arbitraje entre el AMM y las bolsas de criptomonedas, haciendo que se balancee todo otra vez

- La primera generación de AMMs que se hizo popular por protocolos como Bancor, Curve y Uniswap, son los *market makers* de función constante o *constant function market makers* (CFMM). Estos AMM se basan en una función constante en donde las reservas combinadas de activos se tienen que mantener constantes

  - El *market maker* de producto constante o *constant producto market maker* (CPMM) se basa en la función $xy = k$. Cuando se hace el gráfico, el resultado es una hipérbola en donde la liquidez siempre está disponible, pero a precios incrementales que tienden a infinito



- El *market maker* de suma constante o *constant sum market maker* (CSMM) es ideal para transacciones con un impacto en el precio nulo, pero no proporciona liquidez infinita. En este caso, la función es $x + y = k$, y crea una línea recta en donde, por diseño, se puede hacer que se acabe con las reservas de uno de los tokens si los precios no son iguales, y es por eso que no se usan en la práctica



- El *market maker* de media constante o *constant mean market maker* (CMMM) permite la creación de AMMs con más de dos tokens y que se ponderen por fuera de la distribución estándar de 50/50. En este modelo se usa la función de la media geométrica $(x_1 x_2 \dots x_n)^{1/n} = k$, lo cual permite una exposición variable a diferentes activos en el fondo

- Muchos de los AMMs de primera generación están limitados por la pérdida impermanente y la poca eficiencia del capital, que impacta tanto en los proveedores de liquidez como en los comerciantes

  - La pérdida impermanente es la diferencia de valor a lo largo del tiempo entre depositar tokens en un AMM comparado con tener estos tokens en cartera. Esta pérdida ocurre porque el precio de mercado de los tokens en la AMM diverge en cualquier dirección a la del mercado con el que se compara

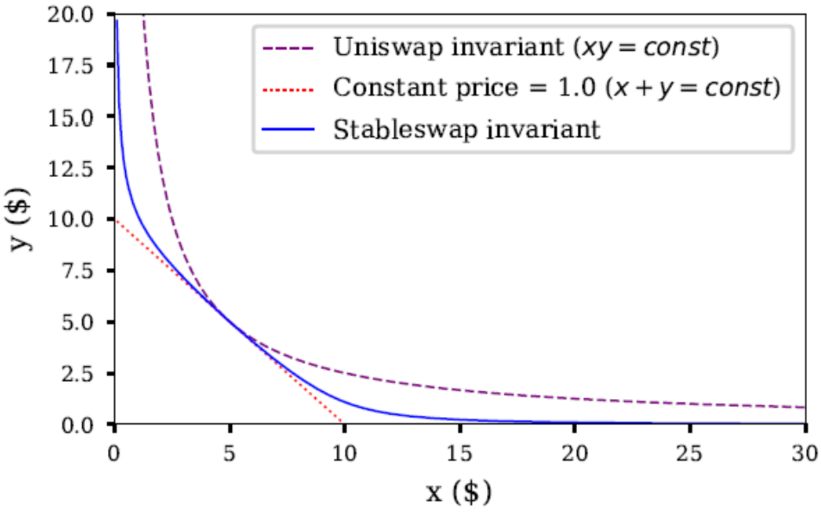
  - Como los AMMs no pueden automáticamente ajustar los tipos de cambio, requieren que haya un comerciante que aproveche el arbitraje en estos AMM con tal de que los precios se vuelvan a igualar. Este beneficio que sacan se extrae de los bolsillos de los proveedores de liquidez, creando una pérdida

  - Los diseños tradicionales de AMM requieren grandes cantidades de liquidez para conseguir el mismo nivel de impacto de precios que una bolsa con un *order book*. Esto se debe a que una porción sustancial de la liquidez de una AMM está disponible solo cuando la curva de valoración comienza a ser exponencial (como con la hipérbola), haciendo que la mayoría de liquidez nunca se usa por comerciantes racionales (debido al impacto de precio extremo experimentado)

  - En esta situación, los proveedores de liquidez no tienen control sobre los precios que se ofrecen a los comerciantes, haciendo que no sea la mejor asignación de capital. Los *market makers* en *order books* pueden controlar exactamente los precios a los que quieren vender y comprar tokens, llevando así a una mayor eficiencia del capital pero que requiere participación activa y la monitorización de la provisión de liquidez

- Las limitaciones anteriores se están superando con proyectos innovadores con nuevos patrones de diseño, tales como *market makers* automatizados híbridos, dinámicos, proactivos o virtuales

  - Existen CFMMs híbridos avanzados que combinan múltiples funciones y parámetros para conseguir comportamientos específicos, tales como una exposición al riesgo ajustada para lo proveedores de liquidez o un impacto en el precio reducido para los comerciantes



- Los AMMs, conocidos como *stableswap invariants*, combinan tanto CPMM con CSMM usando una fórmula avanzada para crear bolsillos de liquidez más densos que hacen que el impacto en el precio sea menor dentro de un rango de transacciones dado. El resultado es una hipérbola que se vuelve lineal para partes grandes de la curva y exponencial cuando se está por los límites de los ejes

- Los CFMMs híbridos permiten que haya un impacto de precios muy bajo al usar una curva que es en su mayoría lineal pero que se vuelve parabólica cuando el fondo se empuja a sus límites. Los proveedores de liquidez ganan más comisiones (aunque más bajas por transacción) porque el capital se usa más eficientemente, mientras que los comerciantes que realizan arbitraje también se pueden aprovechar de reequilibrar el fondo

- Esta curva ofrece intercambios con un impacto de precio muy bajo para tokens que ya tenían un tipo de cambio 1:1 relativamente estable. Esto significa que esta solución está predominantemente diseñada para Stablecoins, aunque también permite usar otros pares de tokens más volátiles con liquidez concentrada similar

- Los *market makers* dinámicos automatizados o *dynamic automated market makers* (DAMM) sonmodelos que permiten usar *data feeds* de precios de los tokens junto a su volatilidad implícita para ayudar a distribuir dinámicamente la liquidez a lo largo del precio de la curva

        - Incorporando múltiples variables dinámicas al algoritmo, es posible crear un *market maker* más robusto que se adapta a las condiciones de mercado cambiantes

        - Durante los periodos de baja volatilidad, uno puede concentrar la liquidez cerca del precio de mercado e incrementar la eficiencia del capital, y expandirse durante los periodos de volatilidad alta con tal de proteger a los comerciantes del deterioro

    - Un modelo de *market maker* proactivo o *proactive market maker* (PMM) imita los comportamientos de *market making* de los humanos de un *limit order book* central

        - Este protocolo utiliza precios de mercado precisos a través de un *data feed* para mover proactivamente la curva de precios de cada activo en respuesta a cambios de mercado, incrementando así la liquidez cerca del precio de mercado actual

        - Esto permite una manera de comerciar más eficiente y reduce las pérdidas de deterioro de los proveedores de liquidez

    - Un modelo de *market maker* automatizado virtual o *virtual automated market maker* (VAMM) permite minimizar el impacto en los precios, mitigar la pérdida impermanente y permite que haya exposición a un único token para los activos sintéticos

        - Estos utilizan la misma fórmula que un CPMM, pero en vez de confiar en el fondo de liquidez, los comerciantes depositan colateral en un contrato inteligente

        - Comerciando activos sintéticos y no el activo subyacente, uno puede ganar exposición a movimientos de precios de una gran variedad de criptoactivos de una manera eficiente. No obstante, aquellos que mantienen una posición abierta en un activo sintético tienen riesgo de que su colateral se liquide si el precio se mueve en su contra

Los fondos de liquidez Uniswap V2

- Uniswap v2 es una bolsa descentralizada basada en un sistema de contratos inteligentes de la *blockchain* de Ethereum y otras redes. Ahora se estudia el *trading* y el *spot price* de este tipo de fondo de liquidez

    - Esta está formada de fondos de liquidez que permiten el *market making* automatizado: permiten que los *traders* compren y vendan activos contra el protocolo sin la necesidad de una tercera parte

- Cada fondo de liquidez de Uniswap v2 consiste de reservas de dos tokens ERC-20 depositados por los proveedores de liquidez, los cuales se benefician de las comisiones que el protocolo carga a los *traders*

- Las comisiones que se recogen se comparten proporcionalmente entre todos los proveedores de liquidez

- Los fondos de liquidez de Uniswap v2 se basan en la fórmula del producto constante vista anteriormente para la cantidad de reservas de dos tokens en el fondo

$$xy = k$$

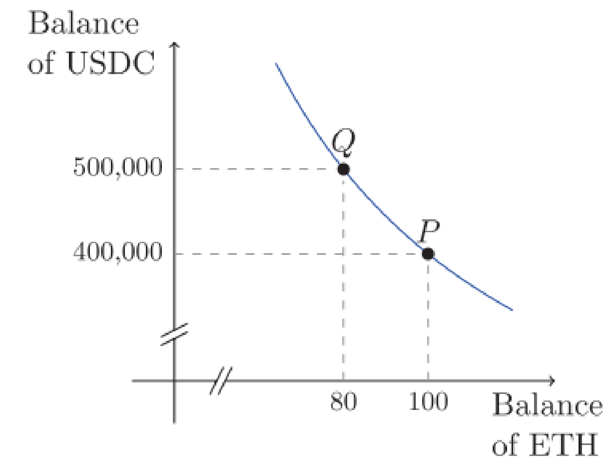
  - Esta fórmula juega un papel muy importante porque determina las cantidades de cada token en cualquier transacción y determina el precio de un token en términos del otro token (del fondo)

  - Además, el precio de un activo en un fondo de liquidez de Uniswap v2 sigue el precio del mercado actual debido a los arbitrajistas externos, los cuales detectarán inconsistencias en los precios y el precio interno del fondo y comerciarán con tal de aprovecharse del arbitraje. Esto, en consecuencia, deja los precios iguales al final

  - Al parámetro k se le conoce como el parámetro de liquidez del fondo, mientras que x e y son las cantidades del token X y el token Y

- Asumiendo que el fondo de liquidez no tiene comisiones, se puede analizar la fórmula para obtener varias fórmulas importantes
  - Si un *trader* quiere comprar una cantidad a del token X, entonces se debe depositar una cantidad b del token Y para mantener el producto constante al parámetro de liquidez. Por lo tanto, después de la transacción la fórmula sería la siguiente:

$$(x - a)(y + b) = k$$



$P = (100, 400000)$: Pool state before the trade.
$Q = (80, 500000)$: Pool state after the trade.

- Considerando la fórmula después de la transacción, se puede obtener la fórmula para a (la cantidad que se recibe si se pagan b tokens Y) y para b (la cantidad que se tiene que pagar por a tokens X)

$$xy + xb - ay - ab = k \;\; \Rightarrow \;\; k + xb - ay - ab = k$$

$$\Rightarrow \; xb - ay - ab = 0 \;\; \Rightarrow \begin{cases} xb - (b + y)a = 0 \\ -ay - (a - x)b = 0 \end{cases}$$

$$\Rightarrow \begin{cases} a = \dfrac{xb}{y + b} \\ b = \dfrac{ya}{x - a} \end{cases}$$

- Un concepto fundamental de los AMMs es el *spot price*, que es el precio que se paga por token X que se recibe del AMM después de depositar una cantidad infinitesimal del token Y

  - Si un *trader* deposita b tokens Y y recibe una cantidad de a tokens A, entonces el precio que el *trader* paga por cada token X es b/a, que también se conoce como el precio efectivo que paga el *trader* por cada unidad del token Y en términos del token X. Por lo tanto, el *spot price* se define de la siguiente manera:

$$p = \lim_{b \to 0} p_e(b) = \lim_{b \to 0} \frac{b}{a}$$

  - Aplicando las fórmulas vistas anteriormente, es posible ver que el el *spot price* con un estado (x, y) del fondo es igual a y/x, que coincide con la pendiente de la línea que pasa a través del origen y el punto (x, y)

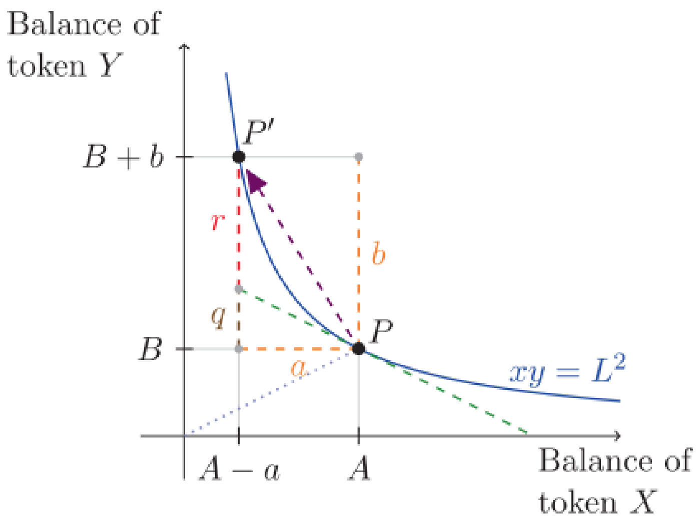
$$p_e(b) = \frac{b}{a} = \frac{b}{\frac{xb}{y+b}} = \frac{y}{x} + \frac{b}{x}$$

$$\Rightarrow \; p = \lim_{b \to 0} p_e(b) = \lim_{b \to 0} \left(\frac{y}{x} + \frac{b}{x} \right) = \frac{y}{x}$$

- Debido a que $b > 0$, entonces $p_e(b) > p$, por lo que el precio efectivo pagado por el *trader* es mayor al *spot price*. La diferencia entre ambos precios indica que los fondos con mucha liquidez tienen precios efectivos y *spot prices* más parecidos

$$p_e(b) - p = \frac{b}{x} > 0$$

- El *spot price* se puede interpretar como el precio que el fondo de liquidez ofrece a los *traders* en un momento particular, aunque estos tengan que pagar más que ese precio para ello



Segment from $(0,0)$ to P. Its slope is $\frac{B}{A}$, that is, the spot price at P.

Portion of the tangent line to the curve defined by $xy = L^2$ at P. Its slope is $-\frac{B}{A}$, which is the opposite of the spot price at P.

b: Amount of token Y deposited.

a: Amount of token X received.

q: Amount of token Y that would have been paid if the effective price had been equal to the spot price.

r: Price impact (equal to $b - q$).

- La diferencia entre la cantidad b que el *trader* paga y la cantidad que hubiera pagado si hubiera comprado la cantidad a a un precio igual al *spot price* se conoce como impacto en el precio o *slippage*

$$Slippage = b - ap = a\left(\frac{b}{a} - p\right) = a(p_e(b) - p) = \frac{ab}{x}$$

- Si el estado de un fondo antes de la transacción es $P = (x, y)$ y después de esta es $P' = (x', y') = (x - a, y + b)$, entonces el precio efectivo es el siguiente:

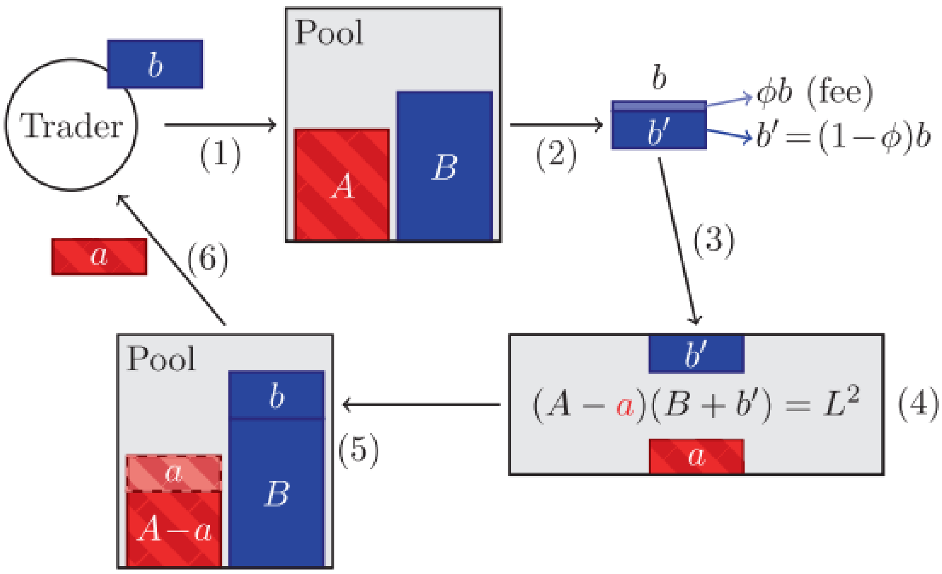
$$p_e = \frac{y' - y}{x - x'}$$

- Como el punto (x', y') pertenece a la curva $xy = k$, se puede entender y' como una función de x', por lo que p_e puede entenderse como una función de x' (donde x e y son fijos), y se puede obtener que el *spot price* es el opuesto a la pendiente de la tangente a la curva en el punto (x, y)

$$p = \lim_{x' \to x} p_e(x') = \lim_{x' \to x} \frac{y' - y}{x - x'} = -\lim_{x' \to x} \frac{y' - y}{x' - x}$$

- Las ecuaciones anteriores son válidas si no se tienen comisiones, pero en general si se tienen. Estas se tienen que añadir a las reservas del fondo y pagarse a los proveedores de liquidez de modo que estos pagos sean proporcionales

  - El protocolo de Uniswap v2 le cobra una comisión al *trader* sobre la cantidad que este deposite, y entonces la cantidad restante es la que se comercia



(1) The trader deposits an amount b of token Y into the pool.
(2) The AMM charges the fee on the amount b.
(3) The remaining amount $b' = (1-\phi)b$ is actually traded.
(4) The AMM computes the amount a of token X that has to be given to the trader, following the curve described by $xy = L^2$ and using b' as the incoming amount of token Y.
(5) The balance of token X in the pool decreases by a, and the balance of token Y in the pool increases by b.
(6) An amount a of token X is given to the trader.

- Siendo x e y los balances y $\phi \in [0,1)$ la comisión, si un *trader* quiere depositar una cantidad b de tokens Y, entonces la cantidad depositada efectiva sería $(1-\phi)b$ y se procedería como si no hubiera comisiones:

$$(x-a)(y+(1-\phi)b) = k$$

$$\Rightarrow\ xy + (1-\phi)bx - ay - (1-\phi)ab = k$$

$$\Rightarrow\ k + (1-\phi)bx - ay - (1-\phi)ab = k$$

$$\Rightarrow\ (1-\phi)bx - ay - (1-\phi)ab = 0$$

$$\Rightarrow \begin{cases} (1-\phi)bx - \big((1-\phi)b + y\big)a = 0 \\ -ay - (a-x)(1-\phi)b = 0 \end{cases}$$

$$\Rightarrow \begin{cases} a = \dfrac{(1-\phi)bx}{y+(1-\phi)b} \\ b = \dfrac{ya}{(1-\phi)(x-a)} \end{cases}$$

- Es importante ver que después de cada transacción, el parámetro de liquidez k se tiene que actualizar, siendo cada vez un poco más grande en cada transacción siempre que la comisión sea positiva

$$x' = x - a \quad \& \quad y' = y + b$$

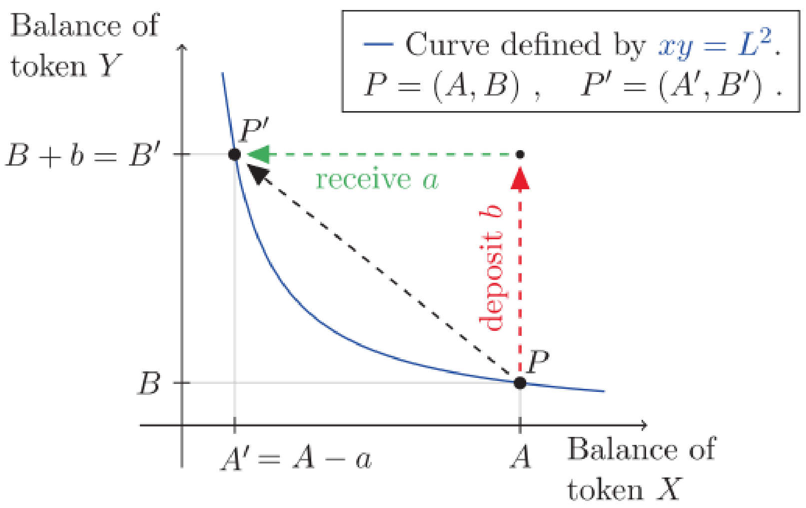
$$\Rightarrow \quad k' = (x - a)(y + b) = xy + bx - ay - ab = xy + (x - a)b - ay$$

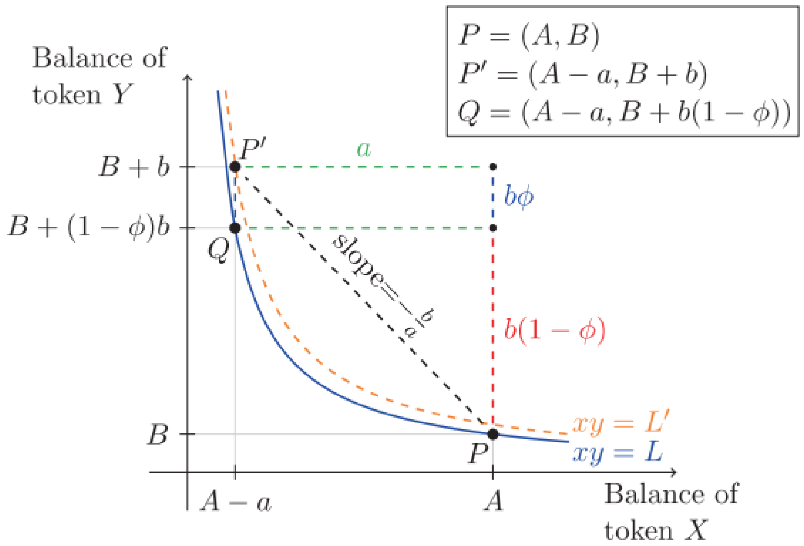
$$\Rightarrow \quad k' = k + (x - a)b - (x - a)(1 - \phi)b = k + \phi b(x - a)$$

  - El nuevo *spot price* también cambiará después de cada transacción si la curva cambia, de modo que será el siguiente:

$$p' = \frac{y + b}{x - a}$$

  - En resumen, si el fondo de liquidez no tiene comisiones, su parámetro de liquidez se mantiene constante después de cada *trade* y los nuevos estados (puntos) se localizan en la misma curva. No obstante, si hay comisiones, entonces el parámetro k incrementa cada vez más y los estados se sitúan en curvas diferentes





- Cuando una transacción se ejecuta, los balances de los tokens en el fondo cambian, y como consecuencia de esta transacción, el precio también cambia. Por lo tanto, es interesante realizar estos análisis para mostrar cómo el *spot price* varía cuando las transacciones ocurren

    - Si se considera un fondo de x tokens X e y tokens Y con una comisión de ϕ si un *trader* quisiera comprar a del token X, entonces se pueden obtener los siguientes cálculos:

        - Para obtener la cantidad de tokens Y que se tienen que pagar, se puede usar la fórmula anteriormente vista para b y calcular el precio medio por unidad:

$$b_1 = \frac{ay}{(1-\phi)(x-a)} \Rightarrow Ave.Price_1 = \frac{b_1}{a}$$

        - Si el *trader* quisiera volver a comprar esa nueva cantidad, ahora se tendrían que usar los valores actualizados de $x' = x - a$ y $y' = y + b_1$, obteniendo el siguiente resultado:

$$b_2 = \frac{ay'}{(1-\phi)(x'-a)} = \frac{a(y+b_1)}{(1-\phi)(x-2a)} > b_1$$

$$\Rightarrow Ave.Price_2 = \frac{b_2}{a} > Ave.Price_1$$

    - Es posible extender el ejemplo anterior para estudiar qué ocurre si en vez de comprar a en una transacción se compra $a/2$ en dos transacciones

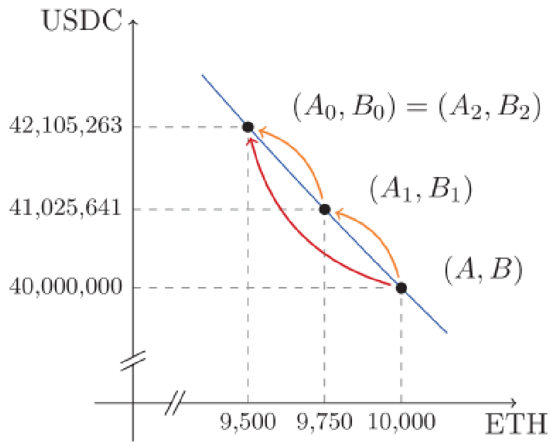
        - Si no hay comisiones, entonces se puede calcular el depósito necesario b_0 para una sola transacción y los depósitos b_1 y b_2 necesarios para realizar la transacción en dos transacciones. De este modo, se puede ver como no hay diferencia entre ejecutar la estrategia en 1 o 2 transacciones

$$(for\ 1\ trans.) \qquad b_0 = \frac{(a_1 + a_2)y}{x - a_1 - a_2}$$

$$(for\ 2\ trans.) \qquad b_1 = \frac{a_1 y}{x - a_1} \quad \& \quad b_2 = \frac{a_2(y + b_1)}{x - a_1 - a_2}$$

$$\Rightarrow\ b_1 + b_2 = \frac{a_1 y}{x - a_1} + \frac{a_2(y + b_1)}{x - a_1 - a_2} = \frac{(a_1 + a_2)y}{x - a_1 - a_2} = b_0$$

- Como el fondo no tiene comisiones, entonces el parámetro k es constante y los estados del fondo se mantienen en la misma curva, haciendo que el punto que se obtiene haciendo una transacción es lo mismo que dividir la transacción en 2



(A, B): Initial pool state.
(A_0, B_0): Pool state after one trade of 500 ETH.
(A_1, B_1): Pool state after one trade of 250 ETH.
(A_2, B_2): Pool state after two consecutive trades of 250 ETH each.

- En cambio, en el caso de que haya comisiones, entonces dividir la transacción en 2 es menos provechoso que realizarla entera

$$(for\ 1\ trans.) \qquad b_0 = \frac{(a_1 + a_2)y}{(1 - \phi)(x - a_1 - a_2)}$$

$$(for\ 2\ trans.) \qquad b_1 = \frac{a_1 y}{(1 - \phi)(x - a_1)} \quad \& \quad b_2 = \frac{a_2(y + b_1)}{(1 - \phi)(x - a_1 - a_2)}$$

$$\Rightarrow\ b_1 + b_2 = \frac{\phi a_1 a_2 y}{(1 - \phi)^2 (x - a_1)(x - a_1 - a_2)}$$

$$> \frac{(a_1 + a_2)y}{(1-\phi)(x - a_1 - a_2)} = b_0$$

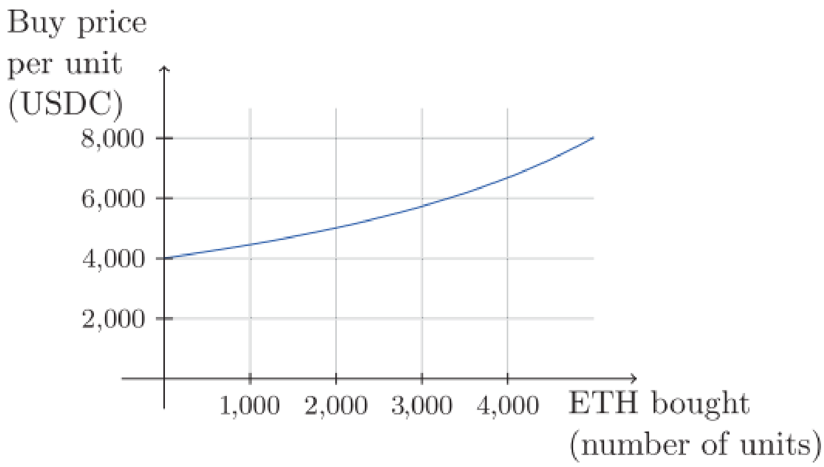
  - En consecuencia, $b_0 \leq b_1 + b_2$ y la igualdad se mantiene si, y solo si, $\phi = 0$

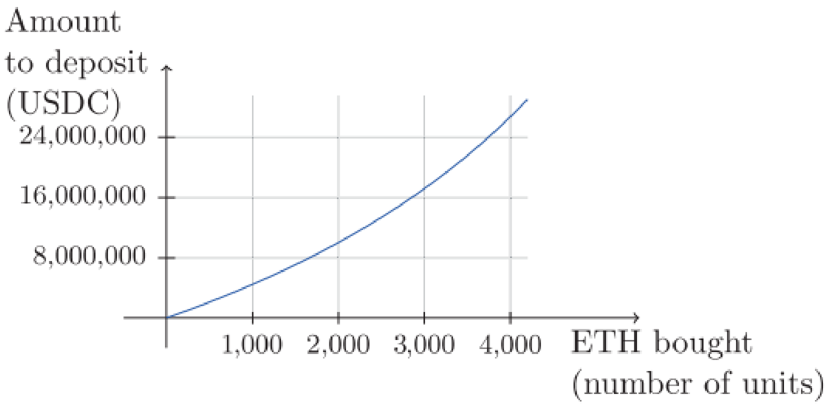
- En la práctica, los *traders* normalmente compran tokens una sola vez y los tokens cuestan más con cada compra, por lo que el tamaño de la transacción impacta en el precio de compra y venta promedio

  - La cantidad del token Y que se tiene que depositar para pagar cada token X, o el precio promedio del token Y por cada token X, se puede expresar a partir de la fórmula para b:

$$b = \frac{ay}{(1-\phi)(x-a)} \;\Rightarrow\; \frac{b}{a} = \frac{y}{(1-\phi)(x-a)}$$

  - Como se puede ver, cuanto mayor es a (la cantidad que se quiere comprar), mayor es el precio por unidad b/a, mientras que la cantidad b de tokens Y a depositar crece con a, por lo que ambas funciones son convexas. La función de b tiene una pendiente mayor que la de b/a debido a que a crece y divide a, haciendo que el efecto no sea tan pronunciado pero que sea creciente

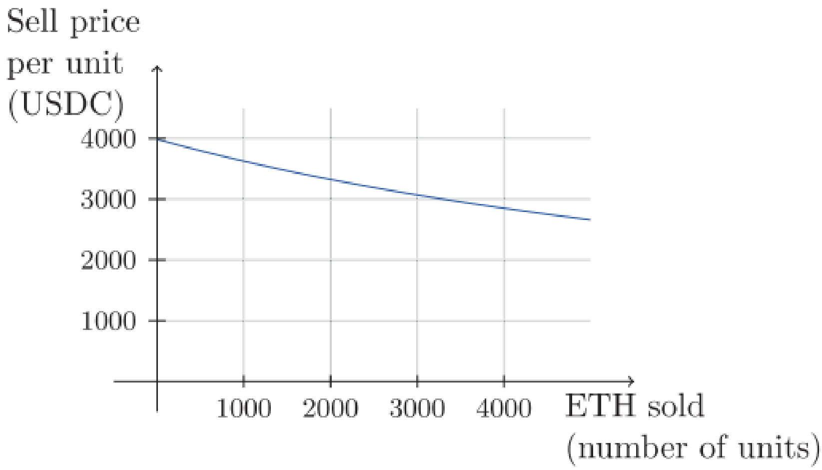




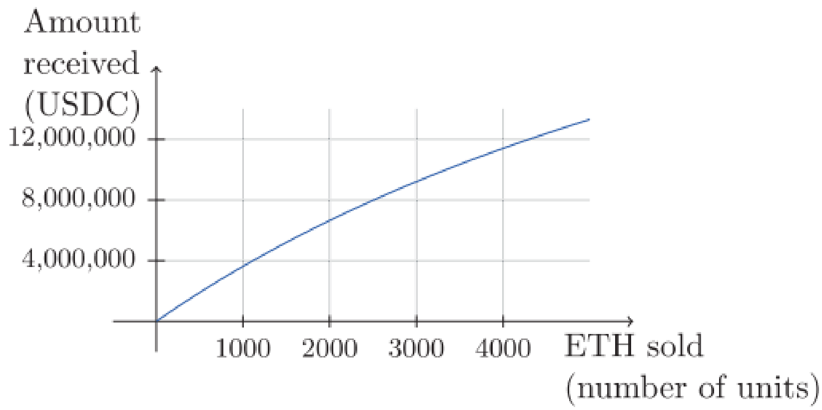
- La cantidad del token X que se tiene que recibir por cada token Y pagado, o el precio promedio del token X por cada token Y, se puede expresar a partir de la fórmula para a:

$$a = \frac{(1-\phi)bx}{y+(1-\phi)b} \Rightarrow \frac{a}{b} = \frac{(1-\phi)x}{y+(1-\phi)b}$$

- Como se puede ver, cuanto mayor es a (la cantidad que se quiere comprar), menor es el precio por unidad a/b, mientras que la cantidad b de tokens Y a recibir crece con a, por lo que la primera función es convexa pero la última es cóncava. La función de b tiene una pendiente positiva y la de a/b es negativa debido a que b divide (relación negativa con a), pero la última tiene mayor magnitud porque en a/b, el movimiento de b se compensa entre el numerador y en el denominador



Amount of USDC received



- Igual que con los precios de venta y de compra, el impacto del tamaño de la transacción también se puede cuantificar para la *ratio* del crecimiento del precio efectivo que paga el *trader*

    - En este caso, el *spot price* actual sería de $p = y/x$, y se quiere calcular el crecimiento el *spot price* cuando se quiere comprar una cantidad a. A partir de las ecuaciones anteriormente vistas, es posible derivar el nuevo *spot price*:

$$x' = x - a \qquad y' = y + b = y + \frac{ay}{(1-\phi)(x-a)}$$

$$\Rightarrow p' = \frac{y'}{x'} = \frac{y + \frac{ay}{(1-\phi)(x-a)}}{x-a}$$

    - Siendo $r = a/x$ la proporción del token X que el *trader* está comprando (con respecto al balance total en el fondo), es posible obtener una expresión de p' en función de p y r:

$$p' = \frac{y + \frac{ay}{(1-\phi)(x-a)}}{x-a} =$$

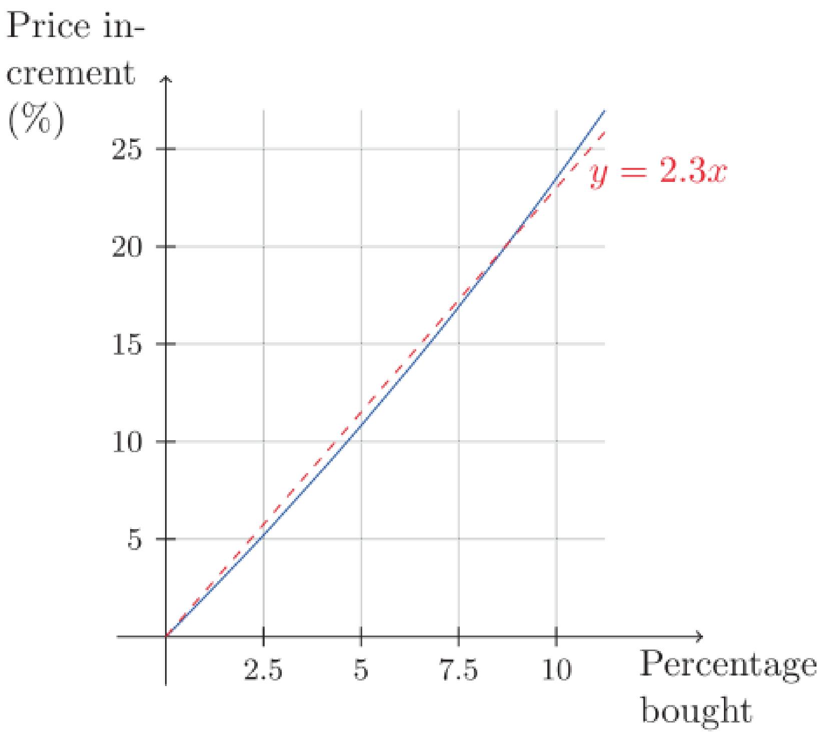
$$= \frac{y/x + \frac{ay/x}{(1-\phi)(x/x - a/x)}}{x/x - a/x} = \frac{p + \frac{p}{(1-\phi)(r^{-1}-1)}}{1-r} =$$

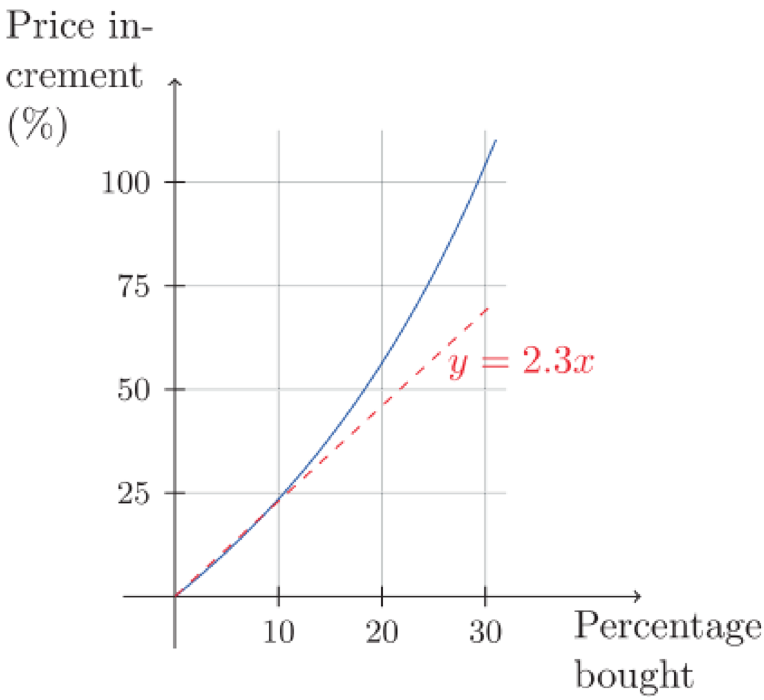
$$= p\frac{1 + (r^{-1}-1)(1-\phi)}{(1-r)(r^{-1}-1)(1-\phi)}$$

- Por lo tanto, la *ratio* de crecimiento del precio es la siguiente, la cual no depende del estado del fondo:

$$\frac{p'}{p} = \frac{1 + (r^{-1}-1)(1-\phi)}{(1-r)(r^{-1}-1)(1-\phi)}$$

- Este crecimiento se puede graficar como una función lineal para porcentajes relativamente pequeños, pero la aproximación lineal falla cuanto uno más se aleja, dado que comienza a tener un comportamiento exponencial





- Ahora que se ha visto el impacto en el precio de las transacciones, uno puede estudiar cómo se añade liquidez a un fondo de liquidez Uniswap v2 y cómo se puede sacar esa liquidez
  - Cuando un proveedor quiere proporcionar liquidez a un fondo, necesita depositar una cantidad de tokens X e Y en una proporción concreta determinada por el estado del fondo
    - Por realizar eso obtendrán tokens LP o *liquidity pool tokens*, que representan una parte del fondo del que son propietarios. El proveedor de liquidez también ganará comisiones de *trading* acorde a esta proporción, y esta proporción variará en el tiempo (cuando entren o salgan proveedores del fondo)
    - Si un proveedor de liquidez quiere proporcionar liquidez al fondo, entonces tiene que depositar una cantidad a de tokens X y b de tokens Y que satisfagan que la proporción es igual a la proporción de reservas que hay en el fondo

$$\frac{b}{A} = \frac{x}{y}$$

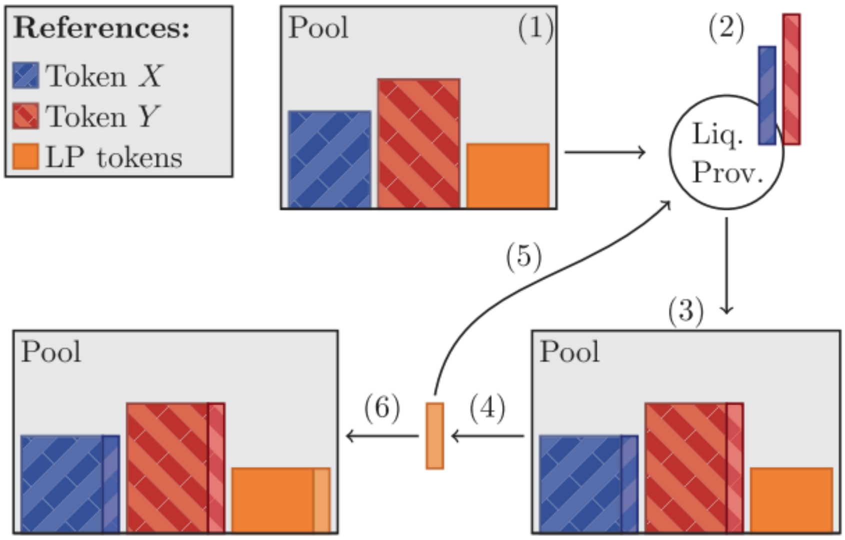
    - Por lo tanto, el *spot price* después del depósito es el mismo que antes debido a la equivalencia anterior

$$p = \frac{y}{x} \quad \Rightarrow \quad p' = \frac{y+b}{x+a} = \frac{y}{x}$$

- Cuando un proveedor proporciona liquidez a un fondo, se le envía tokens LP, los cuales actúan como recibo de la provisión por parte del proveedor de liquidez y representan una porción del fondo

    - La cantidad de tokens LP dentro del fondo es dinámica. El AMM sigue cuantos tokens LP se han creado y cuántos se han dado en retorno por la liquidez

    - Cuando un nuevo usuario proporciona liquidez, el AMM acuña una cantidad apropiada de tokens LP que se envían al nuevo proveedor y representa la porción de la cual es propietario



(1) The pool has a certain amount of reserves and tracks the amount of LP tokens in circulation.
(2) The liquidity provider deposits certain amounts of tokens X and Y into the pool. These amounts must be proportional to the balances of tokens X and Y in the pool.
(3) The tokens deposited are added to the pool reserves.
(4) The AMM mints an amount of LP tokens, which is proportional to the amount of tokens deposited.
(5) The newly minted LP tokens are given to the liquidity provider.
(6) The amount of LP tokens in circulation is updated.

    - Concretamente, si M es la cantidad de tokens LP existentes y un *trader* quiere proporcionar una liquidez de a tokens X y b tokens Y, entonces se puede encontrar obtener la cantidad que recibiría el *trader* es qM y que se actualizaría a $(1+q)M$ después de la provisión de liquidez

$$\frac{b}{a} = \frac{y}{x} \Rightarrow \frac{b}{y} = \frac{a}{x}$$

$$if\ \ q = \frac{a}{x} = \frac{b}{y} \quad \Rightarrow \quad M + qM = (1+q)M$$

- Los AMM Uniswap v2 no requieren que el proveedor encuentre las cantidades exactas de depósito. En verdad, dadas las cantidades C_x de tokens X y C_y de token Y que el proveedor tiene disponibles para depositar, el protocolo calcula las cantidades máximas de cada token que se pueden depositar para mantener la proporción del fondo igual, devolviendo los tokens restantes y dando los tokens LP correspondientes

  - Esta característica es muy útil debido a que el estado del fondo cambia constantemente y no es predecible, debido a que un AMM es descentralizado. Si un proveedor de liquidez intenta calcular en un cierto momento las cantidades que se necesitan depositar, cuando se procese la transacción, el estado podría ser diferente (se han procesado otras transacciones primero) y la proporción necesaria también

  - En el caso en que $C_y/C_x = y/x$, las cantidades y los tokens LP que se reciben son los siguientes:

$$\frac{C_y}{C_x} = \frac{y}{x} \Rightarrow \quad q = \frac{C_x}{x} \quad \Rightarrow qM = \frac{C_x}{x}M = \frac{C_y}{y}M$$

  - En el caso en que $\frac{C_y}{C_x} > y/x$, las cantidades y los tokens LP que se reciben son los siguientes, teniendo en cuenta un diferencial Δ_y para obtener las proporciones adecuadas:

$$\frac{C_y - \Delta_y}{C_x} = \frac{y}{x} \quad \Rightarrow \quad \Delta_y = C_y - \frac{yC_x}{x} \quad \Rightarrow \quad q = \frac{C_x}{x} = \frac{C_y - \Delta_y}{y}$$

$$\Rightarrow \ qM = \frac{C_x}{x}M = \frac{C_y - \Delta_y}{y}M < \frac{C_y}{y}M$$

  - En el caso en que $\frac{C_y}{C_x} > y/x$, las cantidades y los tokens LP que se reciben son los siguientes, teniendo en cuenta un diferencial Δ_x para obtener las proporciones adecuadas:

$$\frac{C_y}{C_x - \Delta_x} = \frac{y}{x} \quad \Rightarrow \quad \Delta_x = C_x - \frac{xC_y}{y} \quad \Rightarrow \quad q = \frac{C_x - \Delta_x}{x} = \frac{C_y}{y}$$

$$\Rightarrow \ qM = \frac{C_y}{y}M = \frac{C_x - \Delta_x}{x}M < \frac{C_x}{x}M$$

  - Por lo tanto, el proveedor de liquidez recibirá una cantidad de tokens LP igual al mínimo entre ambas cantidades vistas:

$$\min\left\{\frac{C_x}{x}M, \frac{C_y}{y}M\right\}$$

- Finalmente, una de las características más útiles para los *traders* que interactúan con AMMs son los agregadores DEX
  - Un agregador DEX es un servicio basado en *blockchain* que funciona como un explorador de precios y liquidez ofrecido por diferentes bolsas descentralizadas o DEXs y ayuda a los *traders* a encontrar el mejor precio para sus transacciones
    - Además, los agregadores están equipados con un algoritmo que permite a los usuarios dividir la transacción en varias contra diferentes AMMs, de modo que se puede obtener el mejor precio posible para cada transacción
    - Claramente, este servicio es prometedor, por lo que vale la pena entender la necesidad de dividir una transacción en dos o más partes
  - Suponiendo que se quiere comerciar una cantidad de T en dos fondos de liquidez diferentes para los mismos tokens X e Y, uno quiere encontrar la cantidad de B que se debería pagar en cada fondo para recibir la máxima cantidad de A posible
    - Para $j \in \{1,2\}$, las reservas del fondo j son x_j e y_j, y las comisiones son ϕ_1 y ϕ_2 para los fondos 1 y 2, respectivamente. Si se comercia una cantidad de b en el primer fondo y una cantidad de $T - b$ (la cantidad restante) en el otro, entonces se pueden obtener las siguientes cantidades a_j de tokens que se obtendría en cada fondo:

$$a_1 = \frac{(1-\phi_1)by_1}{x_1 + (1-\phi_1)b} \qquad a_2 = \frac{(1-\phi_2)(T-b)y_2}{x_2 + (1-\phi_2)(T-b)}$$

    - Con tal de maximizar la cantidad de a que se recibe cuando se comercia con T tokens Y, entonces se tiene que encontrar el máximo de la suma de ambas cantidades, que es una función $f: [0, T] \to \mathbb{R}$. Maximizando la función, se obtienen los siguientes resultados:

$$\max_b f(b) = \max_b \frac{(1-\phi_1)by_1}{x_1 + (1-\phi_1)b} + \frac{(1-\phi_2)(T-b)y_2}{x_2 + (1-\phi_2)(T-b)}$$

$$f'(b) = \frac{x_1 y_1 \overline{\phi_J}[x_2 + \overline{\phi_2}(T-b)]^2 - x_2 y_2 [x_1 + \overline{\phi_1} b]^2}{[x_1 + \overline{\phi_1} b]^2 [x_2 + \overline{\phi_2}(T-b)]^2} = 0$$

$$where \ \ \overline{\phi_J} = 1 - \phi_j \ \ \ for \ \ j \in \{1,2\}$$

- El numerador es un polinomio de grado 2, y como $f'(b) = 0$ solo si este numerador es nulo para $b \in [0, T]$, entonces se puede resolver esta ecuación de segundo grado con la fórmula cuadrática. Las raíces b_1 y b_2 permitirán obtener la cantidad de tokens Y que se debe asignar a cada uno de los fondos para obtener la máxima cantidad de tokens X

- Como se puede ver, es posible optimizar las cantidades que se quieren, pero es necesario tener toda la información sobre el estado de los dos fondos y realizar los cálculos de la optimización. Los agregadores hacen todo esto por el usuario, por lo que son valiosos

Los fondos de liquidez Uniswap V3: funcionamiento básico

- Uniswap V3 es muy diferente de Uniswap V2: aunque los fondos de liquidez de Uniswap V3 también tienen únicamente dos tokens y una fórmula de producto constante, esta versión introduce el concepto de liquidez concentrada

  - La idea principal de esta característica es que los proveedores de liquidez puedan proporcionar liquidez en un rango de precios escogidos, lo cual implica que las reservas de cada posición sean las justas para que sean compatibles con el *trading* en ese rango

    - Cuando el precio se mueve fuera de este rango, la posición del proveedor de liquidez se cambia enteramente a uno de los dos tokens, dependiendo de si el precio se mueve hacia arriba o hacia abajo

  - Una consecuencia de la liquidez concentrada es que las posiciones son altamente personalizadas, dado que los proveedores de liquidez pueden escoger no solo la cantidad a depositar sino el rango de precios en el que quieren proporcionar la liquidez

    - Esto implica que las posiciones en un fondo Uniswap V3 son *non-fungible* o no intercambiables por naturaleza. Por lo tanto, a los proveedores de liquidez se les debe dar tokens no intercambiables a cambio de su depósito, y estos tokens tendrán que mantener un registro con los detalles de la posición particular

    - Además, debido a las características de personalización de la provisión de liquidez, las comisiones deben recogerse y almacenarse de manera separada como tokens individuales en vez de ser reinvertidos automáticamente como liquidez (en las reservas del fondo)

- Como se ha mencionado anteriormente, los proveedores de liquidez proporcionan liquidez en un rango de precios cerrado. No obstante, los límites superiores e inferiores no pueden ser arbitrarios, sino que se tienen que escoger de un subconjunto finito de valores reales positivos
    - Los elementos del subconjunto de números reales se conocen como *ticks* y se indexan por un número entero (llamado *tick index*) de la siguiente manera:

$$p(i) = (1.0001)^i$$

        - Cada *tick* es un 0.01% del siguiente *tick* (un punto base o *base point*)
        - Uniswap V3 utiliza enteros de 24 bits con signo para los *tick indexes*, de modo que el precio máximo y mínimo son los siguientes:

$$p(-2^{23}) = (1.0001)^{-2^{23}} \approx 5.07 \times 10^{-365}$$

$$p(2^{23}-1) = (1.0001)^{2^{23}-1} \approx 1.97 \times 10^{364}$$

    - Uniswap V3 se basa en las fórmulas de producto constante de Uniswap V2 igualmente. En este último fondo, los balances x e y de dos tokens A y B satisfacen la relación $xy = L^2$, donde L es el parámetro de liquidez del fondo
        - El *spot price* p del token X en términos del token Y viene dado por la *ratio* y/x. Por lo tanto, los balances x e y en términos de L y $\sqrt{p}$ se pueden expresar de la siguiente manera:

$$xy = L^2 \;\Rightarrow\; \sqrt{x}\sqrt{y} = L \;\Rightarrow\; \sqrt{x} = \frac{x}{\sqrt{x}} = \frac{L}{\sqrt{y}} \;\Rightarrow\; x = \frac{L}{\sqrt{p}}$$

$$xy = L^2 \;\Rightarrow\; \sqrt{x}\sqrt{y} = L \;\Rightarrow\; \sqrt{y} = \frac{y}{\sqrt{y}} = \frac{L}{\sqrt{x}} \;\Rightarrow\; y = L\sqrt{p}$$

        - Por lo tanto, L y $\sqrt{p}$ se pueden usar para tener registro del estado del fondo de liquidez (solo se necesitan aplicar las fórmulas anteriores), y esto es lo que Uniswap V3 hace. Además, uno puede ver lo siguiente:

$$\sqrt{p}(i) = (1.0001)^{i/2}$$

- Dado cualquier precio p, el *tick index* asociado a p se define como el *tick index* del mayor *tick* t que satisfazca $t \leq p$. Por lo tanto, el *tick index* i asociado al precio p se calcula de la siguiente manera:

$$p = (1.0001)^i \;\Rightarrow\; \log_{10} p = i \log_{10} 1.0001$$

$$\Rightarrow\; i = \frac{\log_{10} p}{\log_{10} 1.0001} \;\Rightarrow\; i = \lfloor \log_{1.0001} p \rfloor = \left\lfloor 2 \log_{1.0001} \sqrt{p} \right\rfloor$$

- Como ejemplo, se considera un fondo Uniswap V3 con tokens ETH y DAI con los siguientes balances:

Tokens	DAI	ETH
Balances	40,000	10

- En este caso, $p = 40000/10 = 4000$ DAI/ETH, y este precio está asociado al *tick index*:

$$i = \lfloor \log_{1.0001} 4000 \rfloor = 82944$$

- Además, el *tick index* 82944 está asociado con el precio 3999.742678, mientras que el *tick index* 82945 está asociado con el precio 4000.142653

$$1.0001^{82944} \approx 3999.742678$$

$$1.0001^{82945} \approx 4000.1426753$$

- Si un proveedor de liquidez quiere proporcionar en rango de precio $[3700{,}4300]$, entonces se pueden encontrar los *tick indexes* que son cercanos a los límites del intervalo aplicando la fórmula vista anteriormente:

$$\lfloor \log_{1.0001} 3700 \rfloor = 82164 \qquad \lfloor \log_{1.0001} 4300 \rfloor = 83667$$

- Ahora, se calculan los *ticks* que corresponden a cada uno de los *tick indexes* y al *tick index* inmediatamente a la derecha, dado que los precios en los que uno está interesado estarán entre esos *ticks*

$$p(82164) = 1.0001^{82164} \approx 3699.634$$

$$p(82165) = 1.0001^{82165} \approx 3700.004$$

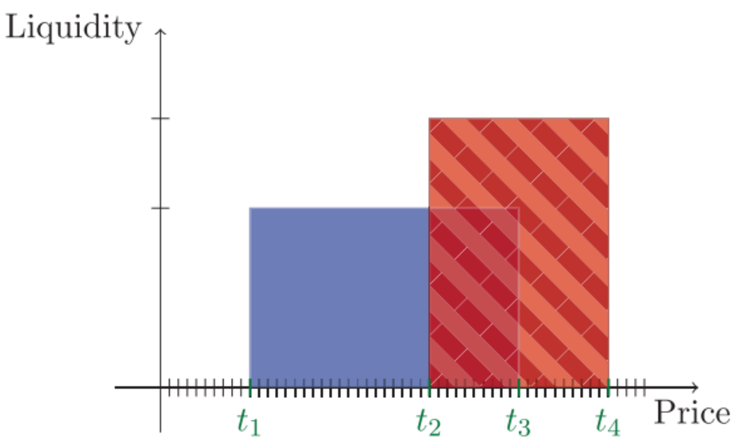
$$p(83667) = 1.0001^{83667} \approx 4299.619$$

$$p(83668) = 1.0001^{83668} \approx 4300.049$$

Tick index	82,164	82,165	83,667	83,668
Tick (approx.)	3,699.634	3,700.004	4,299.619	4,300.049

   - Por lo tanto, el proveedor de liquidez puede escoger proveer liquidez en el intervalo de precios $[3700.004, 4300.049]$

- En Uniswap V3, los *ticks* pueden tener dos estados diferentes, el estado inicializado y el no inicializado. El estado inicializado se refiere a aquellos *ticks* que definen la frontera entre las posiciones corrientes



   - En el ejemplo gráfico, t_1, t_2, t_3 y t_4 se inicializan porque son los que delimitan las posiciones actuales, mientras que todos los otros *ticks* no están inicializados. La liquidez que se proporciona en cada intervalo de *ticks* se determina por el total de proveedores de liquidez en ese rango de *ticks*

   - Estos son los *ticks* de los que uno se tiene que interesar, debido a que el parámetro de liquidez L puede cambiar cuando el precio cruza uno de estos *ticks*. Esto se debe a que es necesario considerar un conjunto de posiciones x y y diferentes cuando se cruza un *tick* inicializado (depende de los proveedores de liquidez en ese rango de precios)

   - La introducción de *ticks* inicializados permite que el protocolo evite hacer cálculos y actualizar las variables cada vez que el precio cruce un *tick*, dado que solo necesita hacerlo cuando el precio cruza un *tick* inicializado

   - Si un *tick* se usa para una nueva posición y este no se ha inicializado aún, este se inicializa, mientras que el *tick* de una sola posición se deja de inicializar cuando esta posición desaparece. Con tal de registrar aquellos *ticks* inicializados y no inicializados, se introdujo un *tick bitmap*, donde cada posición del *bitmap* corresponde a un *tick index* (el valor en la posición es 1 cuando se inicializa y 0 cuando no)

- Anteriormente no se ha puesto restricción en los *tick index*, por lo que cualquier entero de 24 bits con signo podía ser un *tick index*. No obstante, Uniswap V3 generalmente no permite la elección arbitraria de *tick indexes*, sino que introduce el concepto de *tick spacing*, que es una medida de separación entre los *tick indexes* permitidos

  - Concretamente, solo se permiten *tick indees* que son múltiplos del *tick spacing* que se permite. Si este fuera 5, solo se permitirían índices que sean múltiples de 5 como $\ldots, -10, -5, 0, 5, 10, \ldots$

  - En el ejemplo visto anteriormente, si se tuviera un espaciado igual a 5, se consideraría una tabla diferente y se obtendría un intervalo de precio de $[3700.04, 4300.909]$

Tick index	82,160	82,165	83,665	83,670
Tick (approx.)	3,698.155	3,700.004	4,298.759	4,300.909

  - El parámetro de *tick spacing* se define y se fija cuando el fondo se crea. Claramente, si este es pequeño, los proveedores de liquidez pueden escoger rangos más precisos pero que causará que el *trading* sea más caro en términos de comisiones de gas, dado que cada se tienen que fijar nuevos valores para ciertas variables cada vez que el precio cruce un *tick* inicializado

- Uno de los conceptos más importantes a entender es el funcionamiento del *trading* en un fondo Uniswap V3 y la diferencia entre las reservas virtuales y las reales

  - Suponiendo que el proveedor de liquidez proporciona liquidez en un rango de precios $[p_a, p_b]$, donde los precios son en el precio del token X en términos del token Y, la idea principal de Uniswap V3 es que la liquidez de la posición del proveedor usará sus activos para permitir el *trading* entre precios p_a y p_b

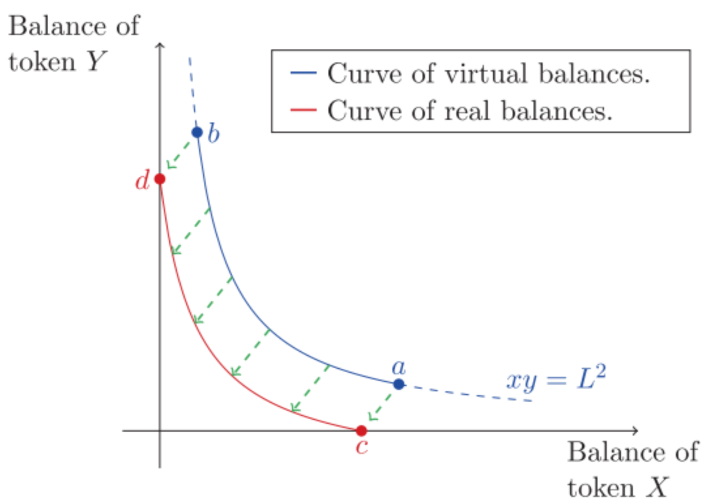
    - Cuando el precio se va fuera de este rango, los activos de la posición ya no se usan para el *trading*, por lo que los balances de la posición no se modifican (a través del intercambio con el comprador/vendedor) hasta que el precio vuelva a entrar en el intervalo

    - Además, si el precio cae por debajo de p_a, entonces la posición del proveedor de liquidez se convierte completamente en el token X, mientras que si va por encima de p_b, se convierte completamente en el token Y

- Por ejemplo, se puede pensar que la cantidad completa del token A de la posición se ha vendido cuando el precio del token X ha incrementado (ya que el precio se expresa en términos de X), de modo que solo tendría el token Y como restante. En ambos casos, el proveedor de liquidez se queda con el activo que es menos valioso

o Aunque el AMM Uniswap V3 se basa en la fórmula de producto constante, se tienen que hacer ciertas modificaciones para permitir que el balance de uno de los tokens se vuelva cero cuando el precio llega a una de las fronteras del intervalo

- En este caso, se tiene que aplicar un tipo de translación de la curva para que pueda tocar un eje del gráfico (para que, en ese punto, la reserva del token representado por el eje contrario sea nula)



- Con tal de encontrar los parámetros de translación que se necesitan aplicar, se calculan los balances de los tokens en el fondo en los estados a y b. Siendo x_a y y_a los balances en el estado a y x_b y y_b los balances en el estado b, se pueden obtener los siguientes resultados:

$$x_a y_a = L^2 \;\Rightarrow\; p_a = \frac{y_a}{x_a} \;\Rightarrow\; a = \left(x_a = \frac{L}{\sqrt{p_a}}, y_a = L\sqrt{p_a}\right)$$

$$x_b y_b = L^2 \;\Rightarrow\; p_b = \frac{y_b}{x_b} \;\Rightarrow\; b = \left(x_b = \frac{L}{\sqrt{p_b}}, y_b = L\sqrt{p_b}\right)$$

- Cuando el precio es p_a, es necesario que el balance del token Y sean iguales a cero o que el estado/punto b toque el eje horizontal, por lo que se tiene que hacer una translación de y_a hacia abajo. Cuando el precio es p_b, se necesita un balance nulo del token X o que a toque el eje vertical, y en este caso se necesita una translación de x_b hacia la izquierda

- Debido a esto, se hace una translación de la curva definida por $xy = L^2$ (llamada curva de balances virtuales) por el vector $(-x_b, -y_a) = (-L/\sqrt{p_b}, -L\sqrt{p_a})$, lo cual permite obtener la curva de balances reales. La expresión se obtiene aplicando la translación a la función expresado aislando una de las variables:

$$x - x_b = \frac{L^2}{y - y_a} \;\Rightarrow\; x - \frac{L}{\sqrt{p_b}} = \frac{L^2}{y - L\sqrt{p_a}}$$

$$\Rightarrow \left(x + \frac{L}{\sqrt{p_b}}\right)\left(y + L\sqrt{p_a}\right) = L^2$$

- Observando que se aplica la translación anterior a los puntos o estados a y b, se pueden obtener los siguientes puntos:

$$a = \left(\frac{L}{\sqrt{p_a}}, L\sqrt{p_a}\right) \;\Rightarrow\; c = \left(\frac{L}{\sqrt{p_a}} - \frac{L}{\sqrt{p_b}}, 0\right)$$

$$b = \left(\frac{L}{\sqrt{p_b}}, L\sqrt{p_b}\right) \;\Rightarrow\; d = \left(0, L\sqrt{p_b} - L\sqrt{p_a}\right)$$

- Con la notación previa, las reservas virtuales o balances virtuales de los tokens X e Y son los números reales positivos x_v y y_b que satisfacen la fórmula $x_v y_v = L^2$ y el *spot price* $p = y_v/x_v$ para un momento concreto. Esto significa que las reservas virtuales definen un punto o estado (x_v, y_v) que se localiza en la curva de balances virtuales

  - Se puede pensar que en cualquier momento en el que el precio esté en el rango $[p_a, p_b]$, las transacciones en el fondo se llevan a cabo siguiente la fórmula del producto constante $x_v y_v = L^2$, aunque las reservas reales del fondo (los balances reales de los tokens) no coincidan con los valores de las reservas virtuales

  - Es importante observar que las reservas reales (x y y) y virtuales (x_v y y_v) de la posición de un proveedor de liquidez se relacionan de la siguiente manera:

$$x_v - \frac{L}{\sqrt{p_b}} = x \;\Leftrightarrow\; x_v = x + \frac{L}{\sqrt{p_b}}$$

$$y_v - L\sqrt{p_b} = y \iff y_v = y + L\sqrt{p_b}$$

$$\Rightarrow \left(x + \frac{L}{\sqrt{p_b}}\right)\left(y + L\sqrt{p_b}\right) = L^2 \iff x_v y_v = L^2$$

- Si el *spot price* en un momento concreto es p, como $p = y_v/x_v$, entonces:

$$x_v = \frac{L}{\sqrt{p}} \quad \& \quad y_v = L\sqrt{p}$$

- Es importante poder calcular los balances de la posición e un proveedor de liquidez cuando el precio es p

Price range	**Real balance of token X**	**Real balance of token Y**
$p \le p_a$	$\frac{L}{\sqrt{p_a}} - \frac{L}{\sqrt{p_b}}$	0
$p_a \le p \le p_b$	$\frac{L}{\sqrt{p}} - \frac{L}{\sqrt{p_b}}$	$L\sqrt{p} - L\sqrt{p_a}$
$p \ge p_b$	0	$L\sqrt{p_b} - L\sqrt{p_a}$

- Si $p = p_a$, el balance del token Y es nulo y aislando x se puede obtener el siguiente estado o punto:

$$P_v = (x_v, y_v) \Rightarrow P_v = \left(\frac{L}{\sqrt{p}}, L\sqrt{p}\right)$$

$$\Rightarrow P_{real} = \left(\frac{L}{\sqrt{p_a}} - \frac{L}{\sqrt{p_b}}, L\sqrt{p_a} - L\sqrt{p_a}\right) = \left(\frac{L}{\sqrt{p_a}} - \frac{L}{\sqrt{p_b}}, 0\right)$$

- Si $p \in [p_a, p_b]$, los balances de los tokens X e Y no son nulos y se puede obtener el siguiente estado o punto:

$$P_v = (x_v, y_v) \Rightarrow P_v = \left(\frac{L}{\sqrt{p}}, L\sqrt{p}\right)$$

$$\Rightarrow P_{real} = \left(\frac{L}{\sqrt{p}} - \frac{L}{\sqrt{p_b}}, L\sqrt{p} - L\sqrt{p_a}\right)$$

- Si $p = p_b$, el balance del token X es nulo y aislando y se puede obtener el siguiente estado o punto:

$$P_v = (x_v, y_v) \Rightarrow P_v = \left(\frac{L}{\sqrt{p}}, L\sqrt{p}\right)$$

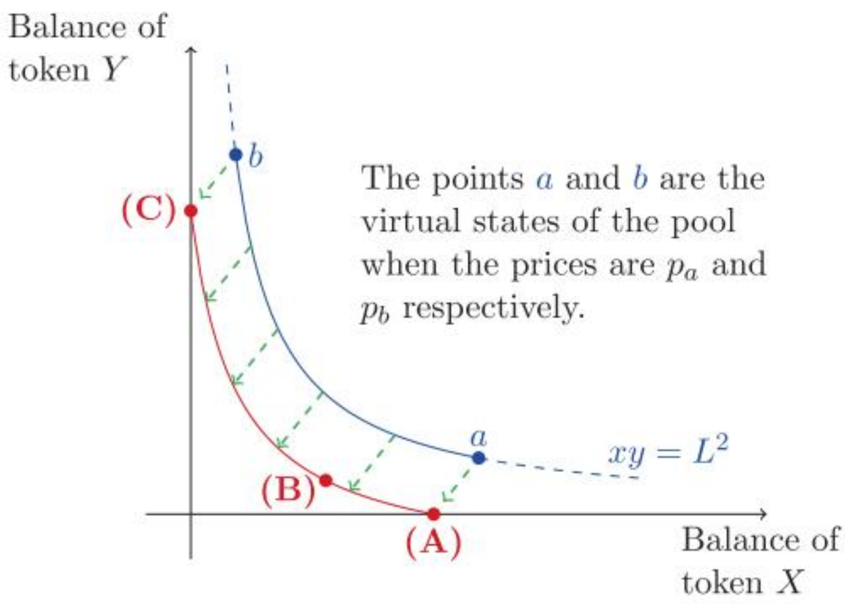
$$\Rightarrow P_{real} = \left(\frac{L}{\sqrt{p_b}} - \frac{L}{\sqrt{p_b}}, L\sqrt{p_b} - L\sqrt{p_a}\right) = \left(0, L\sqrt{p_b} - L\sqrt{p_a}\right)$$

- Suponiendo que el proveedor de liquidez quiere proporcionar liquidez en el rango de precios $[p_a, p_b]$, entonces con un parámetro L fijo, la curva de balances virtuales es fija y la de balances reales está únicamente determinada por esta. Por lo tanto, dado un precio p, los balances reales de la posición del proveedor de liquidez están determinados

    - Es importante ver que como las comisiones de *trading* de Uniswap V3 se almacenan por separado, el parámetro L es constante (no como en Uniswap V2, que cambian), de modo que las curvas anteriormente vistas no cambian y el estado de la posición o punto siempre será un punto en la curva de balances reales

    - En consecuencia, si el proveedor de liquidez quiere comenzar una posición con p_a, p_b y L, dado un precio p, se necesitan depositar las cantidades de tokens X e Y vistas en la tabla anterior

    - Si el precio actual p está por debajo del rango $[p_a, p_b]$, entonces el proveedor de liquidez solo tiene que depositar tokens X en una cantidad $x = \frac{L}{\sqrt{p_a}} - \frac{L}{\sqrt{p_b}}$; si el precio está en el rango, entonces deposita tokens X e Y en cantidades $x = \frac{L}{\sqrt{p}} - \frac{L}{\sqrt{p_b}}$ y $y = L\sqrt{p} - L\sqrt{p_a}$; y si el precio está por encima del rango, entonces solo se depositan tokens Y en una cantidad $y = L\sqrt{p_b} - L\sqrt{p_a}$



- En la práctica, un proveedor de liquidez no quiere escoger un parámetro L, sino un rango de precios $[p_a, p_b]$ y una cantidad concreta de tokens X e Y a depositar. En este caso, se pueden usar las fórmulas de la tabla anterior para encontrar el valor del parámetro L y usar este para calcular las cantidades de tokens que se necesitan depositar dependiendo del precio p

- El funcionamiento de este mecanismo se puede explicar a través de un ejemplo sencillo: se supone que un proveedor de liquidez quiere proporcionar liquidez con tokens ETH y USDC, que el precio de ETH en términos de USDC (p) es de 4000 USDC. Además, se supone que se quiere depositar liquidez en un rango de precios de $(3700.004, 4300.049)$ y que se quiere depositar 2 ETH

  - Con tal de encontrar cuantos USDC se necesitan depositar, primero se calcula el valor del parámetro L usando las fórmulas de las tablas anteriores:

$$x = \frac{L}{\sqrt{p}} - \frac{L}{\sqrt{p_b}}$$

$$\Rightarrow \quad 2 \approx \frac{L}{\sqrt{4000}} - \frac{L}{\sqrt{4300.049}} = L\left(\frac{1}{\sqrt{4000}} - \frac{1}{\sqrt{4300.049}}\right)$$

$$\Rightarrow \quad L \approx \frac{2}{\frac{1}{\sqrt{4000}} - \frac{1}{\sqrt{4300.049}}} \approx 3561.138$$

  - Ahora, se usa el valor de L para poder calcular la cuantía necesaria de USDC:

$$y = L\sqrt{p} - L\sqrt{p_a} \approx 3561.138(\sqrt{4000} - \sqrt{3700.004})$$

$$\approx 8610.458$$

- Es posible calcular el valor de la posición $V(p)$ de un proveedor de liquidez en términos del precio p. Si x_p y y_p son las reservas reales de la posición del proveedor de liquidez cuando el precio es p, entonces $V(p) = x_p p + y_p$

  - Cuando la posición se configura, el valor del parámetro de liquidez L y el rango de precios $[p_a, p_b]$ en el que la liquidez se proporciona son fijos

  - Usando las fórmulas de la tabla anterior, se pueden obtener las siguientes para los valores:

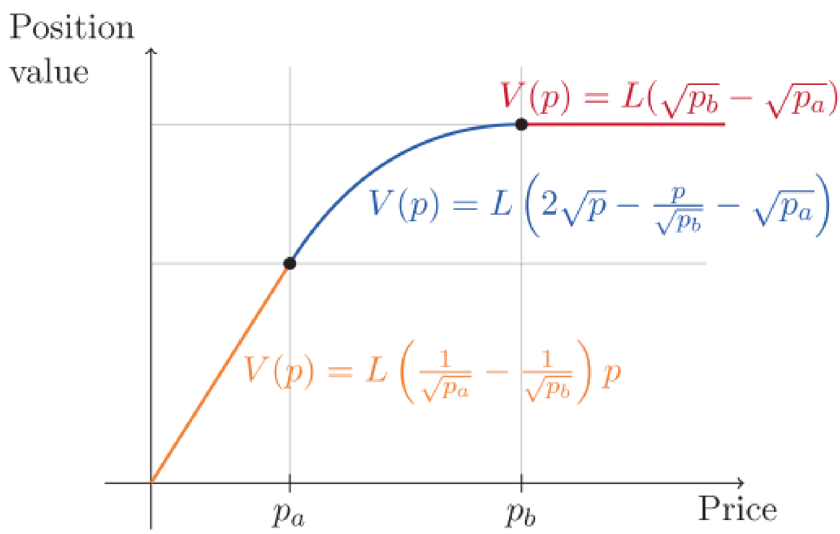
$$V(p) = \left(\frac{L}{\sqrt{p_a}} - \frac{L}{\sqrt{p_b}}\right)p = \left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right)Lp \quad if \;\; p \leq p_a$$

$$V(p) = \left(\frac{L}{\sqrt{p}} - \frac{L}{\sqrt{p_b}}\right)p + L\sqrt{p} - L\sqrt{p_a} =$$

$$= L\left(\sqrt{p} - \frac{p}{\sqrt{p_b}} - \sqrt{p_a}\right) \quad if \;\; p \in [p_a, p_b]$$

$$V(p) = L\sqrt{p_b} - L\sqrt{p_a} = L(\sqrt{p_b} - \sqrt{p_a}) \quad if \;\; p \geq p_b$$

  - A partir de las fórmulas anteriores, es posible realizar un gráfico del valor de la posición dependiendo del precio p#



- Considerando un parámetro de liquidez $L \approx 3561.138$ y un rango de precios $(3700.004{,}4300.049)$, si el proveedor de liquidez deposita 2 ETH y aproximadamente 8610.458 USDC con un precio de 4000 USDC por ETH, se obtiene el siguiente valor de la posición:

$$V(4000) = 2\,ETH \cdot 4000\frac{USDC}{ETH} + 8610.458\,USDC =$$

$$= 16610.458\,USDC$$

  - Si el precio sube a 4200, a partir de las fórmulas anteriores, se obtiene el balance real de tokens en la posición del proveedor de liquidez

$$x = \frac{3561.138}{\sqrt{4200}} - \frac{3561.138}{\sqrt{4300.049}} \approx 0.643$$

$$y = 3561.138\sqrt{4200} - 3561.138\sqrt{3700.004} \approx 14172.435$$

  - Por lo tanto, el valor de la posición ahora se actualiza al siguiente:

$$V(4200) \approx 0.643\,ETH \cdot 4200\frac{USDC}{ETH} + 14172.435\,USDC =$$

$$= 16873.035\,USDC$$

  - Si el proveedor de liquidez hubiera mantenido en su cartera virtual en vez de depositar los tokens en el fondo, su valor hubiera sido mayor al valor de $V(4200)$. Por lo tanto, el proveedor se enfrenta a una pérdida impermanente

$$2 \cdot 4200 + 8610.458 = 17010.458\,USDC > V(4200)$$

Los fondos de liquidez Uniswap V3: valor y combinación de posiciones

- Anteriormente se ha estudiado el valor de la posición del proveedor de liquidez en términos del precio p, de modo que ahora es necesario analizar las posibles pérdidas impermanentes que se pueden producir
  - Se considera un fondo de liquidez Uniswap V3 con tokens X e Y y un proveedor de liquidez que quiere proporcionarla en un rango $[p_a, p_b]$.El precio p_0 es el precio de entrado (el precio al cual el depósito de liquidez se ha realizado, x_0 y y_0 son la cantidad de tokens depositadas a ese precio y L es el parámetro de liquidez en ese momento

- El número p es un número real positivo y x_p y y_p son las reservas reales de los tokens X e Y que corresponden a la posición del proveedor de liquidez cuando el precio del token X en términos del token Y es p

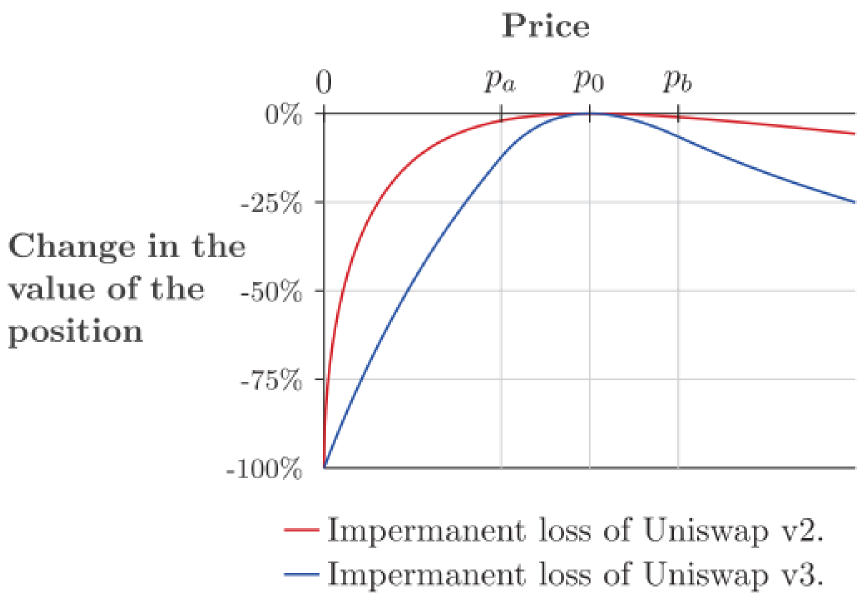
- El valor $V(p)$ es el valor de la posición del proveedor (en términos del token Y), y $W(p)$ es el valor conjunto de las cantidades x_0 y y_0 de los tokens X e Y (en términos del token Y). Este último será el valor de la posición del proveedor de liquidez si no hubiera depositado los tokens en el fondo

$$V(p) = x_p p + y_p \qquad W(p) = x_0 p + y_0$$

- La fracción de la pérdida impermanente se calcula de la siguiente manera:

$$IL(p) = \frac{V(p)}{W(p)} - 1$$

- El análisis de la pérdida impermanente se puede dividir en tres subcasos cada uno dependiendo del precio p_0. El primer caso consiste en ver qué ocurre si $p_0 \in [p_a, p_b]$ y plantear los casos para p (el precio actual o posterior a p_0)



- Si $p_0 \in [p_a, p_b]$, entonces las cantidades a depositar en el momento $t = 0$ son las siguientes:

$$x_0 = L\left(\frac{1}{\sqrt{p_0}} - \frac{1}{\sqrt{p_b}}\right) \quad \& \quad y_0 = L(\sqrt{p_0} - \sqrt{p_a})$$

- Si $p \in [p_a, p_b]$, entonces se pueden obtener los siguientes resultados a partir de las fórmulas de la tabla anteriormente vistas y de la fórmula de la pérdida impermanente:

$$x_p = L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{p_b}}\right) \quad \& \quad y_p = L\left(\sqrt{p} - \sqrt{p_a}\right)$$

$$\Rightarrow IL(p) = \frac{L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{p_b}}\right)p + L\left(\sqrt{p} - \sqrt{p_a}\right)}{L\left(\frac{1}{\sqrt{p_0}} - \frac{1}{\sqrt{p_b}}\right)p + L\left(\sqrt{p_0} - \sqrt{p_a}\right)} - 1 =$$

$$= \frac{2\sqrt{p} - \frac{p}{\sqrt{p_b}} - \sqrt{p_a}}{\frac{p}{\sqrt{p_0}} - \frac{p}{\sqrt{p_b}} + \sqrt{p_0} - \sqrt{p_a}} - 1$$

- Si $p \leq p_a$, entonces se pueden obtener los siguientes resultados a partir de las fórmulas de la tabla anteriormente vistas y de la fórmula de la pérdida impermanente:

$$x_p = L\left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right) \quad \& \quad y_p = 0$$

$$\Rightarrow IL(p) = \frac{L\left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right)p}{L\left(\frac{1}{\sqrt{p_0}} - \frac{1}{\sqrt{p_b}}\right)p + L\left(\sqrt{p_0} - \sqrt{p_a}\right)} - 1 =$$

$$= \frac{\frac{p}{\sqrt{p_a}} - \frac{p}{\sqrt{p_b}}}{\frac{p}{\sqrt{p_0}} - \frac{p}{\sqrt{p_b}} + \sqrt{p_0} - \sqrt{p_a}} - 1$$

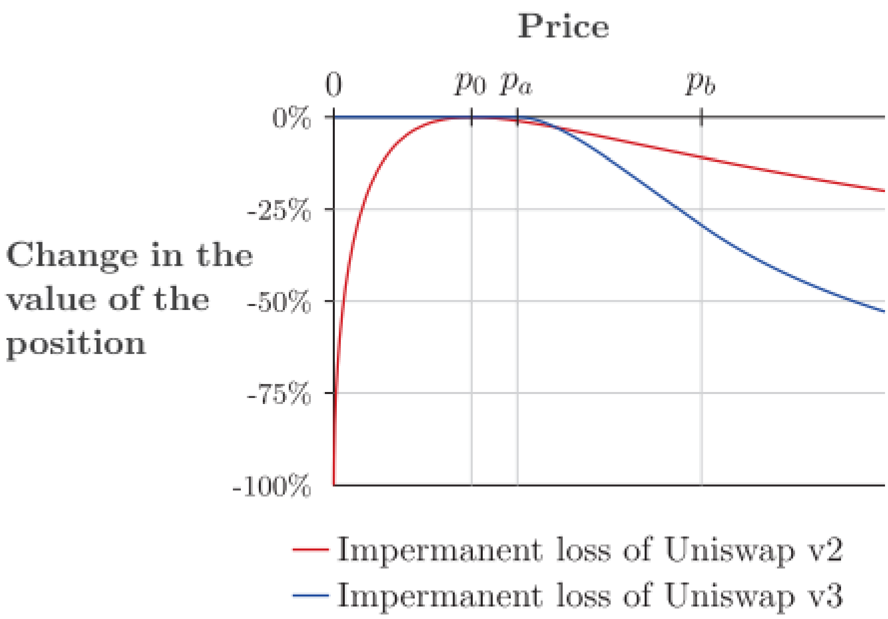
- Si $p \geq p_b$, entonces se pueden obtener los siguientes resultados a partir de las fórmulas de la tabla anteriormente vistas y de la fórmula de la pérdida impermanente:

$$x_p = 0 \quad \& \quad y_p = L\left(\sqrt{p_b} - \sqrt{p_a}\right)$$

$$\Rightarrow IL(p) = \frac{L\left(\sqrt{p_b} - \sqrt{p_a}\right)}{L\left(\frac{1}{\sqrt{p_0}} - \frac{1}{\sqrt{p_b}}\right)p + L\left(\sqrt{p_0} - \sqrt{p_a}\right)} - 1 =$$

$$= \frac{\sqrt{p_b} - \sqrt{p_a}}{\frac{p}{\sqrt{p_0}} - \frac{p}{\sqrt{p_b}} + \sqrt{p_0} - \sqrt{p_a}} - 1$$

- El segundo caso consiste en ver qué ocurre si $p_0 \leq p_a$ y plantear los casos para p (el precio actual o posterior a p_0). En este caso, la pérdida impermanente de Uniswap V3 es más grande cuando el precio está alejado de p_a, pero se estabiliza hacia cero cuando p está cerca de p_a



  - Si $p_0 \leq p_a$, entonces las cantidades a depositar en el momento $t = 0$ son las siguientes:

$$x_0 = L\left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right) \quad \& \quad y_0 = 0$$

  - Si $p \in [p_a, p_b]$, entonces se pueden obtener los siguientes resultados a partir de las fórmulas de la tabla anteriormente vistas y de la fórmula de la pérdida impermanente:

$$x_p = L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{p_b}}\right) \quad \& \quad y_p = L\left(\sqrt{p} - \sqrt{p_a}\right)$$

$$\Rightarrow IL(p) = \frac{L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{p_b}}\right)p + L\left(\sqrt{p} - \sqrt{p_a}\right)}{L\left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right)p} - 1 =$$

$$= \frac{2\sqrt{p} - \frac{p}{\sqrt{p_b}} - \sqrt{p_a}}{\frac{p}{\sqrt{p_a}} - \frac{p}{\sqrt{p_b}}} - 1$$

- Si $p \leq p_a$, entonces se pueden obtener los siguientes resultados a partir de las fórmulas de la tabla anteriormente vistas y de la fórmula de la pérdida impermanente:

$$x_p = L\left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right) \quad \& \quad y_p = 0$$

$$\Rightarrow IL(p) = \frac{L\left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right)p}{L\left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right)p} - 1 = 1 - 1 = 0$$

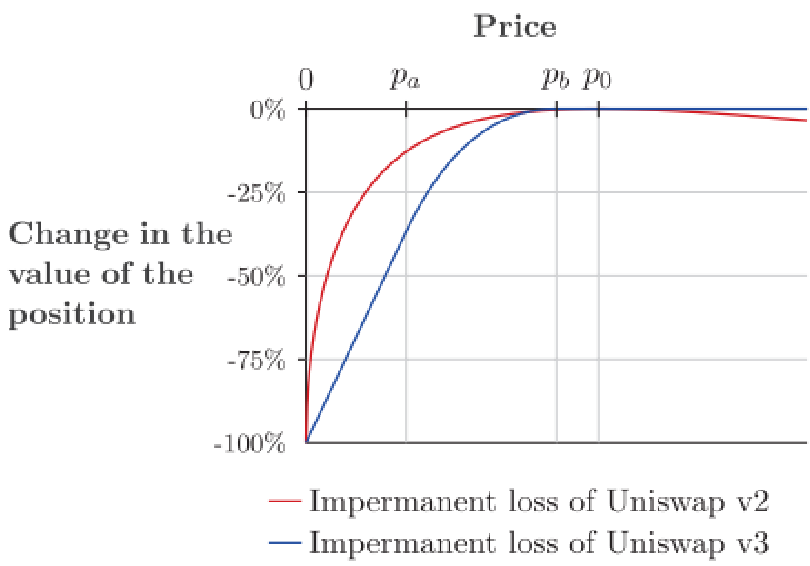
- Si $p \geq p_b$, entonces se pueden obtener los siguientes resultados a partir de las fórmulas de la tabla anteriormente vistas y de la fórmula de la pérdida impermanente:

$$x_p = 0 \quad \& \quad y_p = L\left(\sqrt{p_b} - \sqrt{p_a}\right)$$

$$\Rightarrow IL(p) = \frac{L\left(\sqrt{p_b} - \sqrt{p_a}\right)}{L\left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right)p} - 1 = \frac{\sqrt{p_b} - \sqrt{p_a}}{\frac{p}{\sqrt{p_a}} - \frac{p}{\sqrt{p_b}}} - 1 =$$

$$= \frac{\sqrt{p_b} - \sqrt{p_a}}{p\,\frac{\sqrt{p_a} - \sqrt{p_a}}{\sqrt{p_a}\sqrt{p_b}}} - 1 = \frac{\sqrt{p_a}\sqrt{p_b}}{p} - 1$$

- El tercer caso consiste en ver qué ocurre si $p_0 \geq p_b$ y plantear los casos para p (el precio actual o posterior a p_0). En este caso, la pérdida impermanente de Uniswap V3 es relativamente mayor cuando el precio está lejos de p_b pero se estabiliza hacia cero cuando está cerca de p_b



- Si $p_0 \geq p_b$, entonces las cantidades a depositar en el momento $t = 0$ son las siguientes:

$$x_0 = 0 \quad \& \quad y_0 = L\left(\sqrt{p_b} - \sqrt{p_a}\right)$$

- Si $p \in [p_a, p_b]$, entonces se pueden obtener los siguientes resultados a partir de las fórmulas de la tabla anteriormente vistas y de la fórmula de la pérdida impermanente:

$$x_p = L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{p_b}}\right) \quad \& \quad y_p = L\left(\sqrt{p} - \sqrt{p_a}\right)$$

$$\Rightarrow IL(p) = \frac{L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{p_b}}\right)p + L\left(\sqrt{p} - \sqrt{p_a}\right)}{L\left(\sqrt{p_b} - \sqrt{p_a}\right)} - 1 =$$

$$= \frac{2\sqrt{p} - \frac{p}{\sqrt{p_b}} - \sqrt{p_a}}{\sqrt{p_b} - \sqrt{p_a}} - 1$$

- Si $p \leq p_a$, entonces se pueden obtener los siguientes resultados a partir de las fórmulas de la tabla anteriormente vistas y de la fórmula de la pérdida impermanente:

$$x_p = L\left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right) \quad \& \quad y_p = 0$$

$$\Rightarrow IL(p) = \frac{L\left(\frac{1}{\sqrt{p_a}} - \frac{1}{\sqrt{p_b}}\right)p}{L\left(\sqrt{p_b} - \sqrt{p_a}\right)} - 1 = \frac{\frac{p}{\sqrt{p_a}} - \frac{p}{\sqrt{p_b}}}{\sqrt{p_b} - \sqrt{p_a}} - 1 =$$

$$= \frac{\left(\frac{\sqrt{p_b} - \sqrt{p_a}}{\sqrt{p_a}\sqrt{p_b}}\right)p}{\sqrt{p_b} - \sqrt{p_a}} - 1 = \frac{p}{\sqrt{p_a}\sqrt{p_b}} - 1$$

- Si $p \geq p_b$, entonces se pueden obtener los siguientes resultados a partir de las fórmulas de la tabla anteriormente vistas y de la fórmula de la pérdida impermanente:

$$x_p = 0 \;\;\&\;\; y_p = L\left(\sqrt{p_b} - \sqrt{p_a}\right)$$

$$\Rightarrow IL(p) = \frac{L\left(\sqrt{p_b} - \sqrt{p_a}\right)}{L\left(\sqrt{p_b} - \sqrt{p_a}\right)} - 1 = 1 - 1 = 0$$

- Este caso es un poco más complicado cuando se comparan Uniswap V3 y V2 si el token Y es una *stablecoin*, dado que, en este caso, un fondo Uniswap V2 requeriría que se cambiaran la mitad de los tokens Y en tokens X antes de hacer el depósito. Por lo tanto, si el precio del token X en relación a Y (la *stablecoin*) aumenta, la posición de un fondo Uniswap V3 no cambiaría, pero la de un fondo Uniswap V2 tendría menos tokens X y más tokens Y, dando una pérdida impermanente respecto al valor inicial (pero con una posición más valiosa que Uniswap V3, ya que se tienen más tokens X comparado con tener solo tokens Y como en V3)

- Una cuestión importante es entender cómo múltiples posiciones diferentes se combinan en Uniswap V3. Se tiene que mostrar como el parámetro de liquidez L dado un precio p que no sea un *tick* inicializado es la suma de los parámetros de liquidez (L en un *tick* inicializado no se puede definir bien debido a que a la derecha y a la izquierda de este pueden ser diferentes)

  - Siendo $n \in \mathbb{N}$, se supone que existen n posiciones en un fondo de liquidez V3. Para cada $j \in \{1,2,\dots,n\}$, $\left[a_j, b_j\right]$ es el rango de precio de la posición j y L_j es su parámetro de liquidez. Definiendo $T = \{a_1, a_2, \dots, a_n\} \cup \{b_1, b_2, \dots, b_n\}$, $p_0 \in \mathbb{R}^+$ tal que $p_0 \notin T$ y $A = \left\{j \in \{1,2,\dots,n\} \middle| p_0 \in \left[a_j, b_j\right]\right\}$, entonces el parámetro de liquidez del fondo de liquidez Uniswap V3 al precio p_0 es $\sum_{j \in A} L_j$

    - Si $A = \emptyset$, entonces p_0 no pertenece a ninguno de los rangos de precios de las posiciones en el fondo y el parámetro de liquidez a un precio p_0 es 0, que coincide con $\sum_{j \in A} L_j$ (no tiene sumandos). Por lo tanto, uno tiene que asumir que $A \neq \emptyset$

- Si $j \notin A$, entonces la posición j no proporciona liquidez al precio p_0, y por tanto, no se necesita considerar la posición j. Por lo tanto, $I = \bigcap_{j \in A} \left[a_j, b_j\right]$. Claramente, I es un intervalo y $p_0 \in I$

- Para cada $j \in A$, la ecuación $x_j(p) y_j(p) = L_j^2$ es válida para precios dentro de un intervalo $\left[a_j, b_j\right]$ y, en particular, para precios que pertenecen al intervalo I, dado que $I \subseteq \left[a_j, b_j\right]$

- Debido a la fórmula para el *spot price*, se pueden obtener los siguientes resultados para toda $j \in A$ y $p \in I$:

$$\frac{y_j(p)}{x_j(p)} = p \;\; \Rightarrow \;\; y_j = p x_j(p) \;\; \& \;\; x_j(p) = \frac{L}{\sqrt{p}}$$

- Debido a que el *trading* se hace usando la ecuación de las reservas virtuales y como las posiciones indexadas $j \in A$ proporcionan liquidez en el intervalo I, entonces $p \in I$ y las reservas virtuales que se necesitan considerar son las siguientes:

$$\sum_{j \in A} x_j(p) \;\; \& \;\; \sum_{j \in A} y_j(p)$$

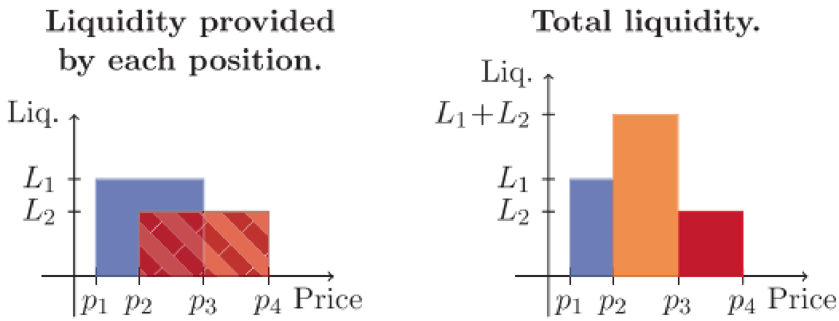
- Cuando $p \in I$, el *trading* que se considera es equivalente al *trading* del AMM de producto constante con parámetro de liquidez $\sum_{j \in A} L_j$. Por lo tanto, este resultado es válido para un precio p_0 tal que $p_0 \in I$

$$\left(\sum_{j \in A} x_j(p)\right)\left(\sum_{j \in A} y_j(p)\right) = \left(\sum_{j \in A} x_j(p)\right)\left(\sum_{j \in A} p x_j(p)\right) =$$

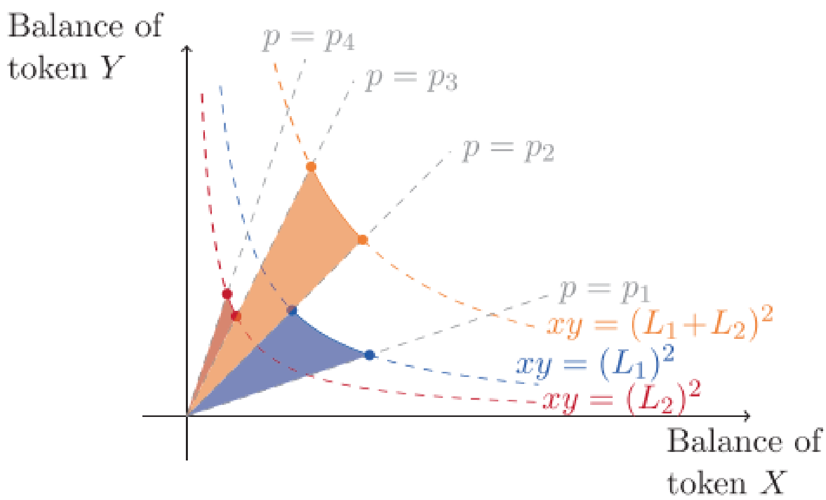
$$= p\left(\sum_{j \in A} x_j(p)\right)^2 = p\left(\sum_{j \in A} \frac{L_j}{\sqrt{p}}\right)^2 = \sum_{j \in A} L_j$$

  - Esto se puede ejemplificar con el caso de un fondo Uniswap V3 con dos posiciones en los intervalos de precio $[p_1, p_3]$ y $[p_2, p_4]$ con parámetros de liquidez L_1 y L_2 respectivamente

    - Claramente, los precios p_1, p_2, p_3 y p_4 se escogen dentro de los *ticks* permitidos. En la intersección de ambos intervalos, el parámetro de liquidez será $L_1 + L_2$



- Si se dibujan las curvas de balances virtuales, las líneas grises representando los estados con un cierto precio p_j y las diferentes curvas indican diferentes curvas dependiendo de las posiciones



Los fondos de liquidez Uniswap V3: variables y comisiones

- Las variables que interesan para el *smart contract* son las que aparecen en la siguiente tabla, donde aquellas que se indican con “(T)” dependen del *tick index* y, por tanto, se tiene una variable de esas por cada *tick* inicializado

Variable	**Notation**	**Type**
Current tick index	i_c	
Square root of the current price	$\sqrt{p}$	
Total liquidity	L_{tot}	
Net liquidity	$\Delta L(i)$	(T)
Gross liquidity	$L_g(i)$	(T)
Total of collected fees	f_g^0, f_g^1	
Fees collected from "outside"	$f_o^0(i), f_o^1(i)$	(T)

- En Uniswap V2, se cobra una comisión de *trading* en el token que entra al fondo para cada transacción que se hace, y las comisiones recogidas se quedan en el fondo, causando que el parámetro de liquidez L incremente. Los proveedores de liquidez recogen su parte de las comisiones cuando sacan su liquidez del fondo

    - En Uniswap V3, las comisiones se cargan en los tokens que entran, pero estas se almacenan de manera separada y se mantienen fuera del fondo como los tokens individuales en las que se pagan

    - Esto significa que las comisiones no se reinvierten y que los proveedores de liquidez ganan una porción de las comisiones solo cuando el precio se mueve en el rango de precios en donde estos proporcionan liquidez

    - Para poder realizar esto, el *smart contract* de Uniswap V3 necesita hacer un seguimiento de varias variables

- Las variables más básicas son el *tick* actual, la raíz cuadrada del precio actual y la liquidez total

    - El *tick* actual es una variable índice que da el *tick index* que corresponde al precio corriente

$$i_c = \lfloor \log_{1.0001} p \rfloor$$

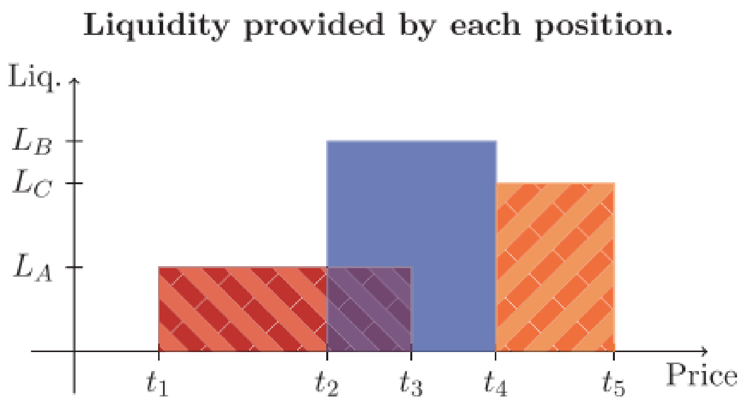
    - En vez de registrar los balances del fondo como en Uniswap V2, V3 registra $\sqrt{p}$ y la liquidez total L_{tot} en un precio específico porque con ambas variables se pueden calcular las reservas virtuales x_v y y_v, ahorrando cálculos (a través de las fórmulas anteriormente vistas)

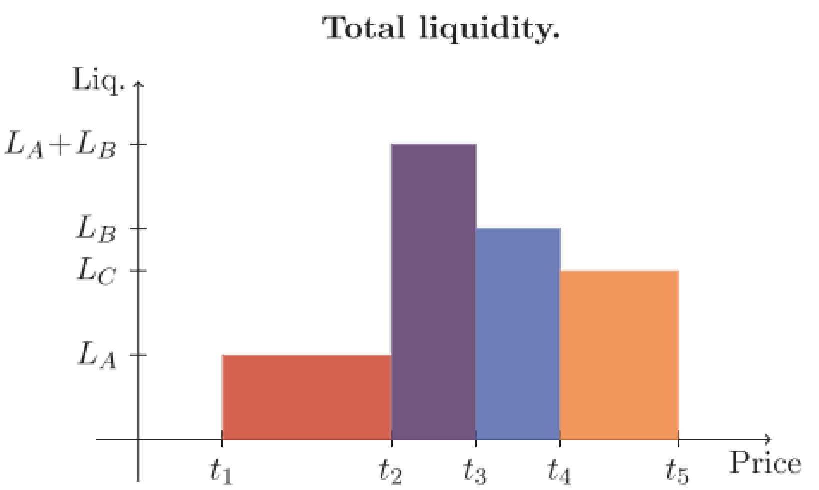
- Para cualquier *tick index* inicializado i, la variable de liquidez neta $\Delta L(i)$ mide el cambio en el parámetro de liquidez L_{tot} cuando se cruza un *tick* inicializado con índice i. Por lo tanto, cuando el precio cruza ese precio con índice i, la cuantía $\Delta L(i)$ se suma o se sustrae de L_{tot} para obtener el parámetro de liquidez total del nuevo intervalo de precio

    - Suponiendo que se tienen *ticks* inicializados t_1, t_2, t_3, t_4 y t_5 con $t_1 < t_2 < t_3 < t_4 < t_5$, y siendo i_j el *tick index* de t_j para $j \in \{1,2,3,4,5\}$, se supone que hay tres proveedores de liquidez que proporcionan liquidez en los intervalos $[t_1, t_3]$, $[t_2, t_4]$ y $[t_4, t_5]$ con parámetros de liquidez L_A, L_B y L_C, respectivamente

    - Suponiendo que los precios se mueven de izquierda a derecha, cuando se cruza t_1, se añade una cantidad de liquidez L_A, haciendo que $\Delta L(i_1) = L_A$; cuando se cruza t_2, se obtiene

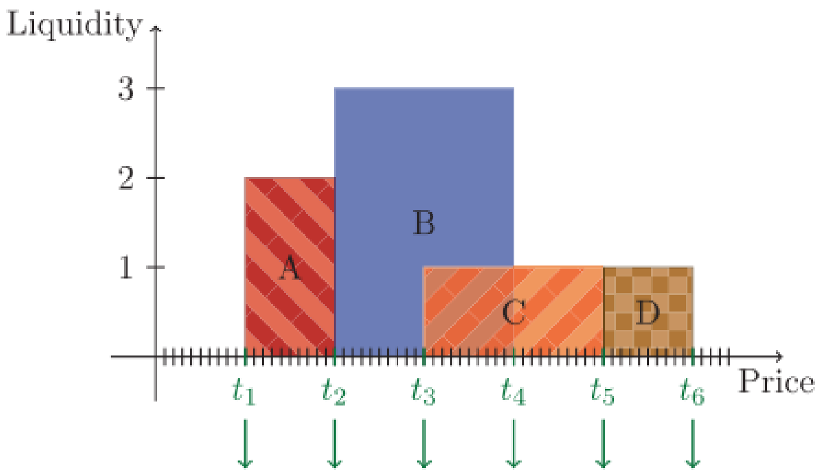
$\Delta L(i_2) = L_B$; cuando se cruza t_3 se obtiene $\Delta L(i_3) = -L_A$; cuando se cruza t_4 se obtiene $\Delta L(i_4) = L_C - L_B$; y cuando se cruza t_5 se obtiene $\Delta L(i_5) = -L_C$





- De este modo, para poder actualizar el valor del parámetro de liquidez todo el rato, solo se necesita calcular $L_{tot} + \Delta L(i)$ si se cruza de izquierda a derecha o $L_{tot} - \Delta L(i)$ si se cruza de derecha a izquierda

- Para cualquier *tick index* inicializado i, la liquidez bruta en un *tick* i es la suma de los parámetros de liquidez de todas las posiciones referenciando el *tick* indexado por i. Esta variable se usa para registrar los *ticks* realmente necesarios



Initialized tick	t_1	t_2	t_3	t_4	t_5	t_6
Net liquidity	2	1	1	-3	0	-1
Gross liquidity	2	5	1	3	2	1

- Si la liquidez bruta de un *tick* inicializado t se vuelve nula cuando la posición se quita, eso significa que t no se referencia por ninguna posición, por lo que se puede no inicializar. Esto no se puede hacer simplemente observado la liquidez neta de un *tick*, dado que la liquidez neta de un *tick* t puede ser nula pero pueden haber posiciones referenciando a ese *tick*

- Por ejemplo, si hay exactamente dos posiciones referenciando a un mismo *tick* t dadas por los intervalos de $[p_1, t]$ y $[t, p_2]$, respectivamente, y ambas posiciones tienen el mismo parámetro de liquidez, entonces si i es el *tick index* i de t, se obtiene la liquidez $\Delta L(i) = 0$. Claramente, aunque $\Delta L(i) = 0$, se tiene que registrar el *tick* t porque hay dos posiciones que lo referencian y se necesitan registrar comisiones que gana cada una de las posiciones separadamente

- El protocolo mantiene un registro de todas las comisiones que se recogen. Estas comisiones no se depositan en el fondo, sino que se guardan por separado en el mismo token en el que se han cobrado

    - Por lo tanto, se necesitan dos variables f_g^0 y f_g^1, las cuales registran la cantidad total de comisiones recogidas por unidad de liquidez

    - De otro modo, sería difícil registrar la cantidad de comisiones que un proveedor de liquidez particular tiene si la cantidad total de liquidez cambia, y en este caso, sería necesaria registrar la cantidad de comisiones propiedad de cada una de las posiciones

- Cualquier *tick* t divide el conjunto de números reales positivos en dos intervalos $[0, t)$ y $[t, \infty)$. Dado cualquier *tick* t y precio actual p, se define el intervalo exterior que corresponde al *tick* t con respecto al precio p como el único de ambos intervalos que no contiene p (este intervalo cambia dependiendo de los cambios de precio)

  - Para un *tick index* inicializado i, el *smart contract* registra dos variables denotadas por $f_o^0(i)$ y $f_o^1(i)$ que registran la cuantía total de comisiones por unidad de liquidez que se han recolectado en cada uno de los dos tokens en el intervalo exterior correspondiente al *tick* con índice i

$$[0, t) \;\; if \;\; i_c \geq i \qquad or \qquad [t, \infty) \;\; if \;\; i_c < i$$

  - Dado un *tick* inicializado t con índice i, si el precio cruza t, entonces el intervalo exterior cambia, y, por tanto, las variables $f_o^0(i)$ y $f_o^1(i)$ se tienen que actualizar consecuentemente. Esto se realiza fijando, para cada $j \in \{0,1\}$, la variable $f_o^j(i)$ como $f_g^j - f_o^j(i)$, dado que la suma de la cuantía de comisiones recogida en los intervalos $[0, t)$ y $[t, \infty)$ es igual al total recogido

- También es necesario mencionar que cuando el nuevo *tick indez* i se inicializa, los valores de $f_o^j(i)$ para $j \in \{0,1\}$ se definen de la siguiente manera, donde i_c es el *tick index* actual:

$$f_o^j(i) = \begin{cases} f_g^j & if \;\; i \leq i_c \\ 0 & if \;\; i > i_c \end{cases}$$

  - Esto es equivalente a decir que todas las comisiones se recogieren por debajo del *tick* cuyo índice es i, que claramente puede no ser verdad. No obstante, como los valores solo se usan para cálculos relativos, la inexactitud no modifica los resultados interesantes

  - Por lo tanto, las variables $f_o^0(i)$ y $f_o^1(i)$ no tienen por qué representar las cuantías reales de comisiones recogidas, sino que son valores auxiliares que se necesitan para el cálculo de las comisiones que cada posición ha ganado

  - Una definición arbitraria como la anterior es necesaria para $f_o^0(i)$ y $f_o^1(i)$ porque no hay manera de determinar la cuantía de comisiones que corresponden a cada uno de los dos intervalos cuando el nuevo *tick* con índice i se inicializa

- Como se ha mencionado, Uniswap V3 cobra una comisión al token que se introduce en el fondo por cada transacción que se realiza, por lo que es necesario entender el funcionamiento de las comisiones

- El *smart contract* de Uniswap V3 cobra una comisión de *trading* al token que se introduce en el fondo por cada transacción que se realiza, pero también permite fijar una comisión de protocolo, que es una fracción de la comisión de *trading* que se guarda para el protocolo y no se da a los proveedores de liquidez

    - Tanto los porcentajes de la comisión de *trading* como de la comisión del protocolo se fijan cuando se crea un nuevo fondo de liquidez. El porcentaje de la comisión del protocolo es cero de manera predeterminada, pero se puede definir como cualquier número dentro del rango de valores permitidos

    - Por simplicidad, se asume que la comisión del protocolo es nula, lo cual no afecta al entendimiento de como Uniswap V3 funciona. La única consecuencia es que la cantidad de comisiones de *trading* cambia, pero la manera en que se recogen y se registran las diferentes variables y cómo se paga a los proveedores de liquidez no se modifica

    - Con tal de poder calcular la cantidad de comisiones que ciertas posiciones recogen, es necesario introducir los conceptos de comisiones desde arriba, desde abajo y dentro de un rango

- Dado un *tick* t con índice i, y dado $j \in \{0,1\}$, se quiere calcular cuantas comisiones en el token j se recogerán (por unidad de liquidez) desde arriba y desde abajo del *tick* t (en los intervalos $[t, \infty)$ y $[0, t)$ respectivamente). Las comisiones desde arriba se denotan por $f_a^j(i)$ y desde abajo se denotan por $f_b^j(i)$

    - Siendo $j \in \{0,1\}$, se pueden definir las comisiones desde arriba y desde debajo de la siguiente manera:

$$f_a^j(i) = \begin{cases} f_g^j - f_o^j(i) & if \;\; i_c \geq i \\ f_o^j(i) & if \;\; i_c < i \end{cases}$$

$$f_b^j(i) = \begin{cases} f_o^j(i) & if \;\; i_c \geq i \\ f_g^j - f_o^j(i) & if \;\; i_c < i \end{cases}$$

    - Como ejemplo se puede considerar un fondo V3 con tokens X e Y y con unos *ticks* t_1, t_2 y t_3 tales que $t_1 < t_2 < t_3$. En la tabla, se muestra como las variables f_g^j, f_o^j, f_a^j y f_b^j (para $j \in \{0,1\}$) se actualizan cuando algunas transacciones hipotéticas se ejecutan y el precio se mueve (dado que p es función de las reservas)

Input data		**Global fee**		**Fee variables at tick t_2**					
Price movement	**Collected fee**	f_g^0	f_g^1	f_o^0	f_a^0	f_b^0	f_o^1	f_a^1	f_b^1
Initial price: t_1	–	0	0	0	0	0	0	0	0
$t_1 \rightarrow t_2$	10 *Y*	0	**10**	0	0	0	0	0	**10**
t_2 tick cross	–	0	10	0	0	0	**10**	0	10
$t_2 \rightarrow t_3$	15 *Y*	0	**25**	0	0	0	10	**15**	10
$t_2 \leftarrow t_3$	4 *X*	**4**	25	0	**4**	0	10	15	10
t_2 tick cross	–	4	25	**4**	4	0	**15**	15	10
$t_1 \leftarrow t_2$	3 *X*	**7**	25	4	4	**3**	15	15	10

- Cuando se realiza una transacción y se cobra la comisión, se tienen que actualizar las comisiones globales f_g^0 y f_g^1 necesitan actualizarse (solo una, dado que la comisión se cobra en solo uno de los dos fondos de tokens). Por lo tanto, los valores de las variables f_a^j y f_b^j para $j \in \{0,1\}$ pueden cambiar también

- Además, los valores de las variables f_o^0 y f_o^1 se mantienen no modificadas cuando una transacción se realiza (si no se cruza ningún *tick* inicializado) dado que estas variables representan las comisiones recogidas fuera del precio del intervalo en el que está actualmente el precio, y los tamaños de las comisiones recogidas fuera del intervalo corriente no cambian cuando la transacción se realiza dentro de este intervalo

- Por otra parte, cuando se cruza un *tick*, las variables f_o^0 y f_o^1 necesitan actualizarse. Los valores de las variables f_a^j y f_b^j para $j \in \{0,1\}$ siguen siendo los mismos cuando el *tick* se cruza debido a que los valores de las variables f_o^0 y f_o^1 se compensan con la modificación en la fórmula de las variables anteriores

- Realmente, las variables f_a^j y f_b^j para $j \in \{0,1\}$ no se registran ni se actualizan en el contrato de Uniswap V3 porque no hay necesidad de ello (solo son variables auxiliares que hacen más fácil el cálculo de la posición). No obstante, se incluyen para entender mejor el funcionamiento del contrato

- Siendo $j \in \{0,1\}$, l y u los *ticks* con $l < u$, i_l el *tick index* de l y i_u el *tick index* de u, se define $f_r^j(l,u)$ para $j \in \{0,1\}$ de la siguiente manera:

$$f_r^j(l,u) = f_g^j - f_b^j(i_l) - f_a^j(i_u) \quad for \;\; j \in \{0,1\}$$

- La variable $f_r^j(l,u)$ representa las comisiones en el token *j* que se han recogido (por cada unidad de liquidez) dentro del rango de precio $[l,u]$

- Es importante observar que como f_r^j se define usando f_a^j y f_b^j, que a su vez se definen a partir de f_o^j, estas variables no se pueden usar para obtener valores reales de lo que realmente representan. No obstante, son útiles para calcular la cantidad de comisiones que una posición ha recogido

- En Uniswap V3, una posición consiste de una dirección (de donde los fondos provienen) y dos *tick indezes* que definen las fronteras del intervalo de precios. Cuando el proveedor de liquidez crea la posición en este intervalo, el *smart contract* asocia la posición (la tupla triple) un parámetro L sino también los valores $f_r^0(l,u)$ y $f_r^1(l,u)$ en el momento del depósito

  - Denotando los valores iniciales de $f_r^0(l,u)$ y $f_r^1(l,u)$ en el momento del depósito como F_0 y F_1, cuando el proveedor de liquidez quiere sacar sus comisiones, el contrato toma las cantidades actualizadas $f_r^0(l,u)$ y $f_r^1(l,u)$ y da al proveedor unas cantidades $L(f_r^0(l,u)-F_0)$ para el token X y $L(f_r^1(l,u)-F_1)$ para el token Y

  - Después, el contrato actualiza las variables donde se almacenan las comisiones como $f_r^0(l,u)$ y $f_r^1(l,u)$ para poder calcular las cantidades de comisiones que la posición recoge a partir de este momento

- Previamente se ha mencionado que las variables f_o^j, f_a^j y f_b^j no se podían usar para determinar exactamente la cantidad de comisiones recogidas en los intervalos que siguen, debido a que cuando el nuevo *tick index* i se inicializa, la variable $f_o^j(i)$ se define arbitrariamente que no mide la cantidad de comisiones recogida anteriormente

  - No obstante, después del estado inicial, y como la variable f_g^j se incrementa continuamente con nuevas comisiones que se cobran las variables f_o^j, f_a^j y f_b^j también se actualizan con los incrementos de las comisiones (cuando las comisiones se recogen en los intervalos correspondientes)

  - Por lo tanto, las diferencias $f_r^0(l,u)-F_0$ y $f_r^1(l,u)-F_1$ siguen y registran los incrementos de las cuantías de las comisiones (por unidad de liquidez) desde la última vez que el proveedor de liquidez sacó las comisiones

Los fondos de liquidez Uniswap V3: transacciones

- En Uniswap V3, las transacciones se realizan en una manera similar a Uniswap V2, siguiendo una fórmula de producto constante. No obstante, con tal de realizar transacciones en este contrato, se necesita tener en cuenta la lista de *ticks* inicializados dado que el parámetro de liquidez puede cambiar cuando el precio cruza un *tick* inicializado
  - Se define ϕ como la comisión de *trading*, p como el precio actual que se representa por el *tick index* i_c, $\mathcal{I}$ como el conjunto de *tick indexes* inicializados, $i_u = \min\{i \in \mathcal{I} | i > i_c\}$ como el *tick index* inicializado más cercano a i_c por la derecha y $i_l = \max\{i \in \mathcal{I} | i \leq i_c\}$ como el *tick index* inicializado más cercano a i_c por la izquierda)
    - Se observa que $i_c \in [i_l, i_u)$ y (i_l, i_u) no contienen ningún *tick index* inicializado (dado que se está entre dos *ticks* inicializados más cercanos a i_c), por lo que el parámetro de liquidez L no cambia dentro de este intervalo
  - Siendo t_l y t_u los *ticks* cuyos índices son i_l y i_u respectivamente, $p \in [t_l, t_u)$ dado que $i_c \in [i_l, i_u)$. Cuando una transacción se va a realizar, el protocolo revisa si esta transacción se puede realizar dentro del intervalo del *tick* actual $[t_l, t_u]$ (si la liquidez del intervalo es la suficiente para realizar la transacción)
    - Si es suficiente, entonces la transacción se ejecuta, y las comisiones correspondientes se recogen. De otro modo, si la liquidez no es suficiente para realizar la transacción completa, entonces solo se realiza una porción de la transacción hasta que el precio llega a la frontera del intervalo $[t_l, t_u]$, y la transacción continua en el siguiente intervalo (a la izquierda o derecha dependiendo de la frontera que se toque) con el resto de la transacción
    - Se tiene que tener en cuenta que los intervalos que se consideran se forman por *ticks* inicializados consecutivos, dado que los *ticks* que no son inicializados pueden ser ignorados
- Siendo L el parámetro de liquidez del intervalo $[t_l, t_u]$ y dado que los balances virtuales son $x_v = L/\sqrt{p}$ y $y_v = L\sqrt{p}$, se pueden estudiar cuatro casos para poder estudiar cómo se ejecutan las transacciones detalladamente
  - El primer caso que se puede estudiar es el caso en el que el agente quiere obtener una cantidad a del token X

- Siendo x_r el balance real del token X (en el intervalo $[t_l, t_u]$), se puede ver que se puede obtener la cuantía de la siguiente manera:

$$x_r = L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{t_u}}\right)$$

- Por lo tanto, si $a \leq x_r$, entonces la transacción se puede ejecutar dentro del intervalo $[t_l, t_u]$. En este caso, se ejecuta la transacción, y como los balances virtuales del token X después de la transacción serán $x_v - a$, el precio se actualiza de la siguiente manera:

$$x_v - a = \frac{L}{\sqrt{p'}} \Rightarrow \sqrt{p'} = \frac{L}{x_v - a} \Rightarrow p' = \left(\frac{L}{x_v - a}\right)^2$$

- De otro modo, si $a > x_r$, entonces la transacción no se puede llevar a cabo completamente dentro del intervalo $[t_l, t_u]$. En este caso, se calcula un primer paso de la transacción, donde una cuantía x_r del token X se comercia, la cuantía necesaria del token Y se calcula y se deposita, y se aplica una cierta comisión en el token Y, resultando en el siguiente balance virtual con precio nuevo igual a t_u:

$$x_v - x_r = \frac{L}{\sqrt{p}} - L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{t_u}}\right) = \frac{L}{\sqrt{t_u}} \Rightarrow x_v = \frac{L}{\sqrt{t_u}} \Rightarrow p' = t_u$$

- Debido a que $a > x_r$, hay una parte del token X que se tiene que incluir en la transacción (la cantidad restante $a - x_r$). Por lo tanto, es necesario cruzar el *tick* t_u y la transacción continua en el siguiente intervalo, que es $[t_u, t']$, donde t' se define como $t' = \min\{i \in \mathcal{I} | t(i) > t_u\}$

- Cuando t_u se cruza, las variables i_c, L_{tot}, $f_o^0(i_u)$ y $f_o^1(i_u)$ se actualizan. Después de esto, la cantidad restante del token X se comercia considerando el intervalo $[t_u, t']$. Las consideraciones anteriores aplican para esta cuantía restante de la transacción, pero para la cantidad $a - x_r$

$$If \;\; x'_r \geq a - x_r \;\; \Rightarrow \;\; p'' = \left(\frac{L'}{x'_v - a + x_r}\right)^2$$

$$If \;\; x'_r < a - x_r \;\; \Rightarrow \;\; continues\ next\ interval$$

- El segundo caso que se puede estudiar es el caso en el que el agente quiere obtener una cantidad del token Y depositando una cantidad a del token X

  - Siendo $a' = (1-\phi)a$ y x_{max} el valor máximo posible del balance real del token X en el intervalo $[t_l, t_u]$ (que no puede superar para hacer que el precio p esté dentro del rango), entonces se obtienen las siguientes cantidades:

$$x_r = L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{t_u}}\right) \qquad x_{max} = L\left(\frac{1}{\sqrt{t_l}} - \frac{1}{\sqrt{t_u}}\right)$$

  - Por lo tanto, si $x_r + a' \leq x_{max}$ o $a' \leq x_{max} - x_r$, la transacción se puede completar dentro del intervalo $[t_l, t_u]$. En este caso, una comisión ϕa se cobra y la cantidad restante del token X se comercia, y como el balance virtual del token X después de la transacción es $x_v + a'$ y el precio es el siguiente:

$$x_v - a' = \frac{L}{\sqrt{p'}} \Rightarrow \sqrt{p'} = \frac{L}{x_v - a'} \Rightarrow p' = \left(\frac{L}{x_v - a'}\right)^2$$

  - De otro modo, si $x_r + a' > x_{max}$ o $a' > x_{max} - x_r$, entonces la transacción no se puede completar dentro de ese intervalo $[t_l, t_u]$. Igual que en el caso anterior, en esta situación, un primer paso de la transacción se calcula y se obtiene la cantidad del token X que se comercia y el balance virtual después de la transacción

$$x_{max} - x_r = L\left(\frac{1}{\sqrt{t_l}} - \frac{1}{\sqrt{t_u}}\right) - L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{t_u}}\right) = L\left(\frac{1}{\sqrt{t_l}} - \frac{1}{\sqrt{p}}\right)$$

$$x_v + x_{max} - x_r = \frac{L}{\sqrt{p}} + L\left(\frac{1}{\sqrt{t_l}} - \frac{1}{\sqrt{p}}\right) = \frac{L}{\sqrt{t_l}} \Rightarrow p' = t_u$$

  - Debido a que $a' > x_{max} - x_r$, hay una parte del token X que se tiene que incluir en la transacción. Como la cantidad del token X que se ha comerciado en el primer paso es $x_{max} - x_r$, la cantidad del token X que el *trader* tiene que usar para este paso es la siguiente:

$$Traded\ quantity\ in\ [t_l, t_u]:\ \frac{1}{1-\phi}(x_{max} - x_r)$$

$$Fee\ collected\ in\ [t_l, t_u]:\ \frac{\phi}{1-\phi}(x_{max} - x_r)$$

- Por lo tanto, la cantidad restante del token X que está disponible para el siguiente paso de la transacción (sin comisiones aplicadas aún) es la siguiente:

$$a - \frac{x_{max} - x_r}{1-\phi} = \frac{a'}{1-\phi} - \frac{x_{max} - x_r}{1-\phi} = \frac{1}{1-\phi}(a' - x_{max} + x_r)$$

- Por lo tanto, es necesario cruzar el *tick* t_l y la transacción continua en el siguiente intervalo, que es $[t', t_l]$, donde t' se define como $t' = \max\{i \in \mathcal{I} | t(i) < t_l\}$. Cuando t_l se cruza, las variables i_c, L_{tot}, $f_o^0(i_l)$ y $f_o^1(i_l)$ se actualizan.

- Después de esto, la cantidad restante del token X se comercia considerando el intervalo $[t', t_l]$. Las consideraciones anteriores aplican para esta cuantía restante de la transacción, pero para la cantidad restante

$$If \ \ x_r' + \frac{1}{1-\phi}(a' - x_{max} + x_r) \le x_{max}$$

$$\Rightarrow \ p'' = \left(\frac{L'(1-\phi)}{x_v'(1-\phi) - a' + x_{max} - x_r}\right)^2$$

$$If \ \ x_r' + \frac{1}{1-\phi}(a' - x_{max} + x_r) \le x_{max}$$

$$\Rightarrow \ \ continues \ next \ interval$$

- El tercer caso que se puede estudiar es el caso en el que el agente quiere obtener una cantidad b del token Y

    - Siendo y_r el balance real del token Y (en el intervalo $[t_l, t_u]$), se puede ver que se puede obtener la cuantía de la siguiente manera:

$$y_r = L\left(\sqrt{p} - \sqrt{t_l}\right)$$

    - Por lo tanto, si $b \le y_r$, entonces la transacción se puede ejecutar dentro del intervalo $[t_l, t_u]$. En este caso, se ejecuta la transacción, y como los balances virtuales del token Y después de la transacción serán $y_v - b$, el precio se actualiza de la siguiente manera:

$$y_v - b = L\sqrt{p'} \ \ \Rightarrow \ \sqrt{p'} = \frac{y_v - b}{L} \ \Rightarrow \ p' = \left(\frac{y_v - b}{L}\right)^2$$

- De otro modo, si $b > y_r$, entonces la transacción no se puede llevar a cabo completamente dentro del intervalo $[t_l, t_u]$. En este caso, se calcula un primer paso de la transacción, donde una cuantía y_r del token Y se comercia, la cuantía necesaria del token X se calcula y se deposita, y se aplica una cierta comisión en el token X, resultando en el siguiente balance virtual con precio nuevo igual a t_l:

$$y_v - y_r = L\sqrt{p} - L\left(\sqrt{p} - \sqrt{t_l}\right) = L\sqrt{t_l} \;\; \Rightarrow \;\; y_v = L\sqrt{t_l}$$

$$\Rightarrow p' = t_l$$

- Debido a que $b > y_r$, hay una parte del token Y que se tiene que incluir en la transacción (la cantidad restante $b - y_r$). Por lo tanto, es necesario cruzar el *tick* t_l y la transacción continua en el siguiente intervalo, que es $[t', t_l]$, donde t' se define como $t' = \max\{i \in \mathcal{I} | t(i) < t_l\}$

- Cuando t_l se cruza, las variables i_c, L_{tot}, $f_o^0(i_l)$ y $f_o^1(i_l)$ se actualizan. Después de esto, la cantidad restante del token Y se comercia considerando el intervalo $[t', t_l]$. Las consideraciones anteriores aplican para esta cuantía restante de la transacción, pero para la cantidad $b - y_r$

$$If \;\; y_r' \geq b - y_r \;\; \Rightarrow \;\; p'' = \left(\frac{y_v' - b + y_r}{L'}\right)^2$$

$$If \;\; y_r' < b - y_r \;\; \Rightarrow \;\; continues \; next \; interval$$

- Finalmente, el último caso que se puede estudiar es el caso en el que el agente quiere obtener una cantidad del token X depositando una cantidad b del token Y

  - Siendo $b' = (1 - \phi)b$ y y_{max} el valor máximo posible del balance real del token Y en el intervalo $[t_l, t_u]$ (que no puede superar para hacer que el precio $p$esté dentro del rango), entonces se obtienen las siguientes cantidades:

$$y_r = L\left(\sqrt{p} - \sqrt{t_l}\right) \qquad y_{max} = L\left(\sqrt{t_u} - \sqrt{t_l}\right)$$

  - Por lo tanto, si $y_r + b' \leq y_{max}$o $b' \leq y_{max} - y_r$, la transacción se puede completar dentro del intervalo $[t_l, t_u]$. En este caso, una comisión ϕb se cobra y la cantidad restante del token Y se comercia, y como el balance virtual del token Y después de la transacción es $y_v + b'$ y el precio es el siguiente:

$$y_v - b' = L\sqrt{p'} \;\Rightarrow\; \sqrt{p'} = \frac{y_v - b'}{L} \;\Rightarrow\; p' = \left(\frac{y_v - b'}{L}\right)^2$$

- De otro modo, si $y_r + b' > y_{max}$ o $b' > y_{max} - y_r$, entonces la transacción no se puede completar dentro de ese intervalo $[t_l, t_u]$. Igual que en el caso anterior, en esta situación, un primer paso de la transacción se calcula y se obtiene la cantidad del token Y que se comercia y el balance virtual después de la transacción

$$y_{max} - y_r = L\left(\sqrt{t_u} - \sqrt{t_l}\right) - L\left(\sqrt{p} - \sqrt{t_l}\right) = L\left(\sqrt{t_u} - \sqrt{p}\right)$$

$$y_v + y_{max} - y_r = L\sqrt{p} + L\left(\sqrt{t_l} - \sqrt{p}\right) = L\sqrt{t_l} \;\Rightarrow\; p' = t_l$$

- Debido a que $b' > y_{max} - y_r$, hay una parte del token Y que se tiene que incluir en la transacción. Como la cantidad del token Y que se ha comerciado en el primer paso es $y_{max} - y_r$, la cantidad del token Y que el *trader* tiene que usar para este paso es la siguiente:

$$Traded\ quantity\ in\ [t_l, t_u]\colon\ \frac{1}{1-\phi}(y_{max} - y_r)$$

$$Fee\ collected\ in\ [t_l, t_u]\colon\ \frac{\phi}{1-\phi}(y_{max} - y_r)$$

- Por lo tanto, la cantidad restante del token X que está disponible para el siguiente paso de la transacción (sin comisiones aplicadas aún) es la siguiente:

$$b - \frac{y_{max} - y_r}{1-\phi} = \frac{b'}{1-\phi} - \frac{y_{max} - y_r}{1-\phi} = \frac{1}{1-\phi}(b' - y_{max} + y_r)$$

- Por lo tanto, es necesario cruzar el *tick* t_u y la transacción continua en el siguiente intervalo, que es $[t_u, t']$, donde t' se define como $t' = \min\{i \in \mathcal{I} | t(i) > t_u\}$. Cuando t_u se cruza, las variables i_c, L_{tot}, $f_o^0(i_u)$ y $f_o^1(i_u)$ se actualizan.

- Después de esto, la cantidad restante del token Y se comercia considerando el intervalo $[t_u, t']$. Las consideraciones anteriores aplican para esta cuantía restante de la transacción, pero para la cantidad restante

$$If\ \ y_r' + \frac{1}{1-\phi}(b' - y_{max} + y_r) \le y_{max}$$

$$\Rightarrow \ p'' = \left(\frac{L'(1-\phi)}{y_v'(1-\phi) - b' + y_{max} - y_r}\right)^2$$

$$If \ \ y_r' + \frac{1}{1-\phi}(b' - y_{max} + y_r) \leq y_{max}$$

$$\Rightarrow \ \ continues \ next \ interval$$

- Cabe destacar que en cada uno de los cuatro casos se ha asumido implícitamente que existe un *tick* t' inicializado consecutivo que permite definir el siguiente intervalo $[t_u, t']$ o el anterior $[t', t_l]$
  - Aunque esto suele ser el caso, puede ser que ese *tick* no exista. Esto implicaría que no hay liquidez por encima de t_u o por debajo de t_l, y, por tanto, que no se puede continuar en el intervalo siguiente o anterior
  - Por lo tanto, el algoritmo pararía, y la transacción se completaría solo de manera parcial

Los fondos de liquidez Uniswap V3: análisis de provisión de capital

- Comparado con Uniswap V2, los fondos Uniswap V3 proporcionan liquidez con mayor eficiencia de capital, lo que significa que con la misma cantidad de tokens depositados, uno puede obtener un mayor parámetro de liquidez en el intervalo escogido. Aunque esto es beneficioso para los *traders* (por el poco impacto en el precio), para los proveedores de liquidez tiene diferentes implicaciones, or lo que se analizan dos situaciones con dos suposiciones diferentes
  - Primero, se supone que se ejecutan exactamente las mismas transacciones en un fondo Uniswap V2 y en un fondo Uniswap V3. En este caso, la cantidad de comisiones recogidas es la misma y el total de comisiones recogidas se comparte entre todos los proveedores de liquidez
    - En Uniswap V2, las comisiones se comparten de manera proporcional a los depósitos que los proveedores de liquidez realizan, mientras que en Uniswap V3, esta proporcion también depende del intervalo que cada proveedor escoge
    - Específicamente, los proveedores ganan comisiones solo por aquellas transacciones (o porción de transacciones) que se realiza entro del intervalo que escogen, y en este caso, su proporción de comisiones es igual al parámetro de liquidez proporcionado dividido entre la liquidez total (más alto que en Uniswap V2)

        - Por lo tanto, aunque la cantidad de comisiones serán las mismas, los proveedores de liquidez pueden obtener más o menos comisiones con respecto a su cuota en Uniswap V2

    - Para la segunda situación…

        - …

- EFICIENCIA CAPITAL
- INDEPENDENCIA POSICIÓN

Los fondos de liquidez aritméticos

- Aunque hasta se han visto fórmulas de producto constante, tales como las de los fondos de liquidez Uniswap, también se pueden concebir otros tipos de diseño como el aritmético

    - Aunque los fondos de liquidez que tienen funciones constantes permiten el comercio fácil y descentralizado, estos imponen estructuras rígidas que no permiten a sus proveedores de liquidez adaptar su estrategia basándose en dinámicas de mercado

        - Por lo tanto, estos proveedores de liquidez incurren en pérdidas predecibles debido a la naturaleza de la función de *trading* del fondo y su falta de insumos estratégicos, normalmente provenientes del *slippage* (como en Uniswap V2) y de selección adversa

    - El diseño de fondos de liquidez aritméticos (ALPs) permite introducir funciones de impacto y funciones de cotización o de fijación de precios que permiten a los proveedores ajustar dinámicamente los precios y gestionar la liquidez basándose en condiciones de mercado

        - Las funciones de impacto son aquellas que definen cómo afectan las transacciones al tipo de cambio o precio marginal p, similar a cómo la transacción afectaría al precio medio en finanzas tradicionales

        - Las funciones de cotización o de fijación de precio p determinan el *bid-ask spread* alrededor del tipo de cambio o precio marginal, permitiendo que los fondos de liquidez reflejen el riesgo, la demanda esperada y los precios futuros

        - Los proveedores de liquidez en estos ALPs operan sobre periodos de *trading* fijos, y en cada periodo se puede comprometer liquidez a una estrategia predefinida la cual no se puede cambiar en medio del periodo. Esta estructura permite que los proveedores optimicen sus rendimientos mientras que se evitan disrupciones por extracciones y depósitos en medio del periodo

- Se puede demostrar que los fondos de liquidez de función constante son un subconjunto de los ALPs, lo que significa que, especificando una función de cotización y de impacto concretas, se puede replicar cualquier fondo de función constante

  - No obstante, también se puede demostrar que estos fondos de liquidez con función constantes son subóptimos en términos de maximización de riqueza de los proveedores de liquidez, dado que les falta flexibilidad para ajustar la estrategia de valoración dinámicamente

- Una medida que se puede considerar para medir las pérdidas financieras esperadas debido a los factores predecibles en los fondos de liquidez de función constante, como el *slippage* y los costes de oportunidad

  - Un ALPs proporciona flexibilidad que puede neutralizar la pérdida predecible a través de permitir que los proveedores de liquidez respondan estratégicamente a flujos de *trading* predecibles

  - Con tal de garantizar que los ALPs no incentivan a las oportunidades de arbitraje libres de riesgo, es posible derivar condiciones para las funciones de impacto y de cotización que mantengan estable los precios marginales del fondo, por lo que se evitaría el arbitraje cíclico

- Con este diseño se puede desarrollar una estrategia para los proveedores de liquidez que maximicen rendimientos a través de fijar unas funciones basadas en suposiciones sobre la demanda, la volatilidad del precio y las preferencias de riesgo

  - En este caso, la solución al problema es una solución cerrada que expresa el *bid-ask spread* en función de la sensibilidad del mercado y los niveles objetivo de liquidez. Esto permite que los proveedores maximicen sus beneficios dentro de un cierto periodo de tiempo

  - Además, también se puede implementar numéricamente para poder resolver el problema de implementación usando exponenciación de matrices, haciendo posible el cálculo de cuotas eficientemente en *blockchains*

- Este tipo de fondos de liquidez se ha intentado validar empíricamente a través del uso de datos de Uniswap y de Binance, a través de simular un

ALP con estrategias de provisión de liquidez óptimas y compararlo a un fondo con función constante

	Average	Standard deviation
ALP ($\kappa = 0.1$)	-0.004%	0.719%
ALP ($\kappa = 0.05$)	0.717%	2.584%
Buy and Hold	0.001%	0.741%
Uniswap v3	-1.485%	7.812%

- Los resultados muestran como los ALP permiten que los proveedores de liquidez obtengan mejores rendimientos promedios y menos volatilidad de los beneficios que con Uniswap

- Debido a su buen funcionamiento, se han pensado en posibles extensiones como fondos de múltiples tokens y modelos predictivos para poder usar señales de *trading* en tiempo real para los proveedores de liquidez

- El modelaje matemático que permite presentar resultados formales sobre las estrategias óptimas y sobre el diseño de los ALPs se basa en la teoría de control óptimo y en ciertas suposiciones

    - …

La ejecución y la especulación con AMMs

- Los AMMs en finanzas descentralizadas permiten el *trading* sin la necesidad de intermediarios, pero de la misma manera que en un mercado tradicional, el *trader* se enfrenta a la decisión de cómo comerciar con estos

    - La mayoría de AMMs utilizan funciones de producto constantes, tales como los Uniswap V2 y los Uniswap V3. Es posible plantar diversas estrategias de *trading* óptimo y de estrategias de arbitraje con este tipo de AMMs

        - El foco está puesto sobre todo en los costes de ejecución, que están dirigidos principalmente por la concavidad de la función de *trading*

        - Los costes de ejecución son lineales en el tamaño de la transacción (por cada unidad adicional, el precio se mueve linealmente debido a la función), pero no son lineales en la profundidad de la liquidez (cantidad de reservas de cada activo) y en el tipo de cambio o precio p

- Los costes de la convexidad nacen de la estructura de los fondos de liquidez con funciones de producto constante, donde los costes de ejecución incrementan cuando el tamaño de la transacción crece (cuando se va a los extremos) y están inversamente relacionados con la profundidad del fondo

    - Estos costes impactan a los *traders* (los *liquidity takers*) que necesitan balancear entre el riesgo de fluctuación del precio p y los costes de ejecución cuando se hacen grandes transacciones o se arbitra entre mercados centralizados y descentralizados

- Con tal de poder entender lo que ocurre en escenarios en donde se opera entre mercados centrados y descentralizados, se puede plantear un problema de control estocástico que capture el objetivo de los *liquidity takers* de balancear costes de ejecución con el precio en tiempo continuo

    - En un primer modelo, se puede plantear que los precios se forman primero en los CEX, debido a la cantidad tan grande de liquidez que hay. En este caso, los *liquidity takers* usan estos precios como precios de referencia para comerciar con los DEXs, ajustando su velocidad de *trading* basada en los costes de convexidad del fondo

    - En un segundo modelo, se puede plantear que los precios se forman primero en los DEX, debido a la gran actividad de *liquidity takers* y de proveedores de liquidez. Este modelo captura los escenarios de precios de activos eficientemente valorados en la DEX misma

    - El último modelo deriva los precios de ambos lugares, permitiendo que los *liquidity providers* consideren la cointegración que puede existir entre los precios de los CEX y los DEX. Este ajuste se parece al *trading* de activos cotizados entre múltiples bolsas

    - Para cada modelo de *trading*, se pueden derivar PDEs semilineales, pero como estas son muy intensivas de resolver, entonces se proponen aproximaciones donde los costes de convexidad se tratan como una función constante a piezas. Esto lleva a una estrategia de forma cerrada que aproxima la estrategia óptima mientras que son asequibles computacionalmente