BALANCER SOLIDITY NUMERICAL APPROXIMATIONS

FERNANDO MARTINELLI

1. Original formulas to be approximated

When a user sends to Balancer tokens j to get tokens i, we can calculate q_i (the amount of tokens i) a user gets when sending q_j (the amount of tokens j):

$$q_i = \left(1 - \left(\frac{Q_j}{Q_j + q_j}\right)^{\frac{w_j}{w_i}}\right).Q_i \tag{1}$$

It is also very useful for traders to know how much they need to send of the input token q_j to get a desired amount of output token q_i. We can calculate the amount q_j as a function of q_i similarly as follows:

$$q_j = \left(\left(\frac{Q_i}{Q_i - q_i}\right)^{\frac{w_i}{w_j}} - 1\right).Q_j \tag{2}$$

Since solidity does not have fixed point algebra or more complex functions like fractional power we use the following binomial approximation:

$$(1+x)^{\alpha} = 1 + \alpha x + \frac{(\alpha)(\alpha-1)}{2!}x^2 + \frac{(\alpha)(\alpha-1)(\alpha-2)}{3!}x^3 + \cdots = \sum_{k=0}^{\infty}\binom{\alpha}{k}x^k \tag{3}$$

which converges for $|x| < 1$
We derive the solidity approximation of the first equation using the binomial approximation above:

$$q_i = \left(1 - (1+x)^{\alpha}\right).Q_i \tag{4}$$

where

$$x = \frac{Q_j}{Q_j + q_j} - 1 = \frac{-q_j}{Q_j + q_j} \tag{5}$$

and

Date: July 2019.

$$\alpha = \frac{w_j}{w_i} \tag{6}$$

q_j is by design limited to 10% of Q_j. That is, no trade can exchange/sell more than 10% of Balancer's balance of those tokens. This prevents excessive slippage loss for traders.

$|x|$ is then by design always lower than 0.1.

Since calculations in solidity are done in integers, the order of the operations we choose is fundamental for the calculation to be correct. E.g. $(99/100) * 100 = 0$. This happens because 99/100 is truncated to 0 and then multiplied by 100.

To avoid dividing two numbers that are close to each other (which truncates all the precision as in the example above), we multiply by Q_i all terms in the binomial expansion used for approximating q_i:

$$q_i = Q_i + Q_i\alpha x + Q_i\frac{(\alpha)(\alpha-1)}{2!}x^2 + Q_i\frac{(\alpha)(\alpha-1)(\alpha-2)}{3!}x^3 + \cdots \tag{7}$$

To make the solidity implementation simpler and more elegant using recursive functions, we can rewrite q_i as:

$$q_i = \sum_{k=0}^{n} T_k \tag{8}$$

where:

$$T_0 = Q_i \tag{9}$$

and

$$T_k = \frac{(\alpha - (k-1))\, x}{k} T_{k-1} \tag{10}$$

The binomial approximation described above is especially accurate for small values of α. When $\alpha > 1$ we split the calculation into two parts for increased accuracy:

$$q_i = \left(1 - \left(\frac{Q_j}{Q_j + q_j}\right)^{int\left(\frac{w_j}{w_i}\right)} \left(\frac{Q_j}{Q_j + q_j}\right)^{w_j \% w_i}\right) .Q_i \tag{11}$$