CURVE TRICRYPTO2 OPTIMIZATION METHODS REPORT

G.V. OVCHINNIKOV AND F.A. SKOMOROKHOV

1. Newton Y

We propose a new way of finding the new y value. Instead of using Newton's method, which is an iterative process and gives approximate solution only we derive an exact formula for new y. The final solution is exact and its quality depends on technical implementation only. The original function F may be rewritten as a polynome of K_0:

$$aK_0^3 + bK_0^2 + cK_0 + d = 0$$

with the following coefficients:

$$a = \frac{1}{27}D^3$$
$$b = -\frac{1}{9}D^3 - \frac{2}{27}D^3\gamma + \frac{1}{27xz}AD^5\gamma^2$$
$$c = \frac{1}{9}D^3 + \frac{1}{27}D^3\gamma(\gamma+4) + A\gamma^2D^2(x+z-D)$$
$$d = -\frac{1}{27}D^3(1+\gamma)^2$$

For the sake of simplicity and overflow prevention, we may divide all coefficients by D^3:

$$a = \frac{1}{27}$$
$$b = -\frac{1}{9} - \frac{2}{27}\gamma + \frac{1}{27xz}AD^2\gamma^2$$
$$c = \frac{1}{9} + \frac{1}{27}\gamma(\gamma+4) + \frac{1}{D}A\gamma^2(x+z-D)$$
$$d = -\frac{1}{27}(1+\gamma)^2$$

Now we find a root of the polynome using Cardano's formula:

$$\Delta_0 = b - \frac{3ac}{b}$$
$$\Delta_1 = 2b - \frac{9ac}{b} + \frac{27a^2d}{b^2}$$
$$C = b^{\frac{2}{3}}\sqrt[3]{\frac{\Delta_1 + \sqrt{\Delta_1^2 - 4\frac{\Delta_0^3}{b}}}{2}}$$

Date: August 2022.

Now root is obtained as:

$$K_0 = -\frac{1}{3a}(b + C + \frac{\Delta_0}{C})$$
$$y = \frac{K_0 D^3}{27xz}$$

Our decimal experiments show this solution outperforms Newton's y^* up to several orders. In integer implementation y^* given by the formula above is in most cases very close to the result of Newton Y procedure. Please note that technical implementation relies on $1E18$ basis.Here is how the landscape looks like:

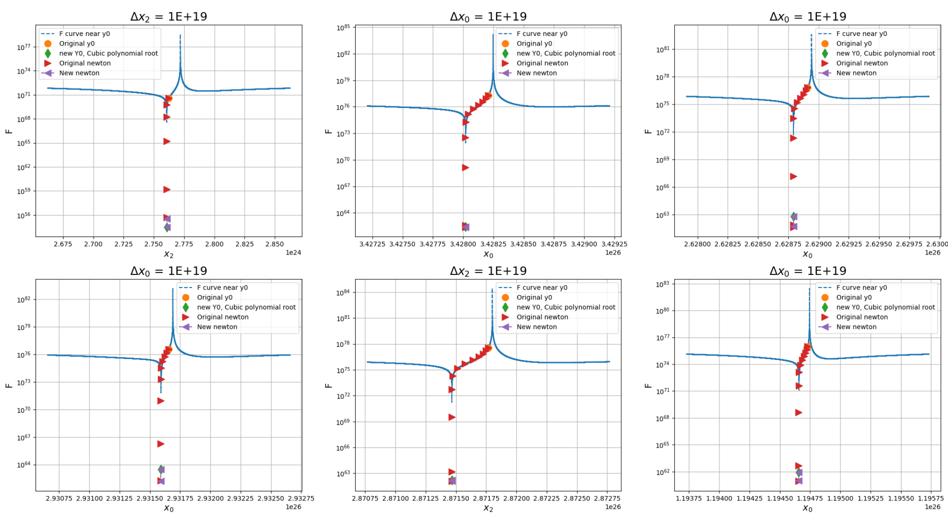

FIGURE 1. Landscape near y^*.

This plot shows a landscape new y^*. X-axis represents the corresponding coin (y or x_i if viewd as a component of vector of coin values). Y-axis shows the value of F. To make it more vivid we use logarithmic scale and take absolute values for y-axis. The orange circle here shows the original initial value for Newton Y, green diamond denotes the new value given by the polynomial root formulas above. Red and violet triangulars shows a path of Newton optimization for the original and new y0 values respectively. You can see that in general Newton Y makes only 1 step from the new y0. This may be potentially be improved to the case when only polynomial root is enough so Newton Y will only decrease the F value - this case requires more precision.

2. NEWTON D

We propose a modification of the initial D_0 in Newton_D and using the Halley method instead of the original Newton.

The new initial D_0 is the root of some polynomial. We algebraically transform the original function F to obtain a polynomial with respect to D. The following expression is obtained:

$$
\begin{aligned}
(1)\qquad & -\frac{1}{27}D^9(1+\gamma)^2 + D^6(3P + 4\gamma P + \gamma^2 P - 27A\gamma^2 P) \\
& \qquad + 27D^5 A\gamma^2 PS - D^3(81P^2 + 54\gamma P^2) + 729P^3
\end{aligned}
$$

Next we transform $D^5 = D^3 * S * D_0$, where D_0 is the original initial point in the current version of the Newton algorithm. S is more than optimal D^* in most cases, while D_0 is less. Thus we make some compensation for approximating D. Now the polynomial has the following form:

$$
\begin{aligned}
(2)\qquad & -\frac{1}{27}D^9(1+\gamma)^2 + D^6(3P + 4\gamma P + \gamma^2 P - 27A\gamma^2 P) \\
& \qquad - D^3(81P^2 + 54\gamma P^2 - 27D_0 S^2 A\gamma^2 P) + 729P^3
\end{aligned}
$$

Using $t = D^3$ substitution we may find a root of the polynomial above using Cardano's formula [1].

First, calculate coefficients of the polynome $at^3 + bt^2 + ct + d$:

$$
\begin{aligned}
a &= -\frac{1}{27}(1+\gamma)^2 \\
b &= 3P + 4\gamma P + \gamma^2 P - 27A\gamma^2 P \\
c &= -81P^2 - 54\gamma P^2 + 27D_0 S^2 A\gamma^2 P \\
d &= 729P^3
\end{aligned}
$$

Now some intermediate variables:

$$
\begin{aligned}
\Delta_0 &= b^2 - 3ac \\
\Delta_1 &= 2b^3 - 9abc + 27a^2 d \\
C &= \sqrt[3]{\frac{\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}}
\end{aligned}
$$

Now root is obtained as:

$$
\begin{aligned}
t &= -\frac{1}{3a}\left(b + C + \frac{\Delta_0}{C}\right) \\
D &= \sqrt[3]{t}
\end{aligned}
$$

This D is a new starting point for the Newton_D algorithm.

Now we use Halley's algorithm instead of Newton to obtain D^*. Please note that we apply the algorithm to (1), not to original F. The only difference is how descent direction is formulated: while Newton's update rule is

$$D_{k+1} = D_k - \frac{F}{F'}$$

, Haley has

$$D_{k+1} = D_k - \frac{2FF'}{2F'^2 - FF''}.$$

For Haley method F is defined by (1) and

[1] `https://en.wikipedia.org/wiki/Cubic_equation`

$$F' = -\frac{1}{3}D^8(1+\gamma)^2 + 6D^5(3P + 4\gamma P + \gamma^2 P - 27A\gamma^2 P)$$
$$+135D^4A\gamma^2 PS - 3D^2(81P^2 + 54\gamma P^2)$$
$$F'' = -\frac{8}{3}D^7(1+\gamma)^2 + 30D^4(3P + 4\gamma P + \gamma^2 P - 27A\gamma^2 P)$$
$$+540D^3A\gamma^2 PS - 6D(81P^2 + 54\gamma P^2)$$