This library implements two different methods for solving the equation $x \cdot \left(\frac{a}{b}\right)^x = \frac{c}{d}$:

- Function `solveExact(a, b, c, d)`, which opts for accuracy over performance
- Function `solveQuick(a, b, c, d)`, which opts for performance over accuracy

The solution to:

$$x \cdot \left(\frac{a}{b}\right)^x = \frac{c}{d}$$

Can be computed via:

$$x = \frac{W\Big(\log\left(\frac{a}{b}\right) \cdot \frac{c}{d}\Big)}{log\left(\frac{a}{b}\right)}$$

Where $W(z)$ computes a value of x which solves the equation $x \cdot e^x = z$.

The function $f(x) = x \cdot e^x$ reaches a global minimum at $x = -1$:

- $f'(x) = (x+1) \cdot e^x$ hence $f'(x) = 0 \iff x = -1$
- $f''(x) = (x+2) \cdot e^x$ hence $f''(-1) = 1/e > 0$

Since $f(-1) = -1/e$, the equation $x \cdot e^x = z$ has no solution for $z < -1/e$.

In order to handle the rest of the input domain, we split it into several sections.

When opting for accuracy, we split the input domain $z = \log\left(\frac{a}{b}\right) \cdot \frac{c}{d}$ into:

$$\underbrace{-1/e \ldots 0}\Big|\underbrace{0 \ldots \infty}$$

For $-1/e \le z < 0$, we approximate $W(z)$ via the Newton-Raphson converging sequence:

$$\begin{cases} y_0 = -z \\ y_{n+1} = \frac{y_n^2 + e^{y_n} \cdot z}{y_n - 1} \end{cases}$$

For $0 \le z < \infty$, we approximate $W(z)$ via the Newton-Raphson converging sequence:

$$\begin{cases} y_0 = \begin{cases} z & z < 1 \\ \log(z) & z \ge 1 \end{cases} \\ y_{n+1} = \frac{y_n^2 \cdot e^{y_n} + z}{(y_n + 1) \cdot e^{y_n}} \end{cases}$$

When opting for performance, we split the input domain $z = \log\left(\frac{a}{b}\right) \cdot \frac{c}{d}$ into:

$$\underbrace{-1/e \dots 0}\Big|\underbrace{0 \dots 1/e}\Big|\underbrace{1/e \dots 24+1/e}\Big|\underbrace{24+1/e \dots \infty}$$

For $-1/e \le z \le +1/e$, you may observe that $W(z) = \frac{W\left(\log\left(\frac{a}{b}\right)\cdot\frac{c}{d}\right)}{log\left(\frac{a}{b}\right)} = \frac{c}{d} \cdot \sum_{n=1}^{\infty} \frac{(-n)^{n-1}}{n!} \cdot \left(\log\left(\frac{a}{b}\right) \cdot \frac{c}{d}\right)^{n-1}$:

- For $-1/e \le z \le 0$, which implies that $a \le b$, we compute $W(z) = \frac{c}{d} \cdot \sum_{n=1}^{\infty} \frac{(+n)^{n-1}}{n!} \cdot \left(\log\left(\frac{b}{a}\right) \cdot \frac{c}{d}\right)^{n-1}$
- For $0 \le z \le +1/e$, which implies that $a \ge b$, we compute $W(z) = \frac{c}{d} \cdot \sum_{n=1}^{\infty} \frac{(-n)^{n-1}}{n!} \cdot \left(\log\left(\frac{a}{b}\right) \cdot \frac{c}{d}\right)^{n-1}$

As you can see, when $a = b$, both formulas can be reduced to $W(z) = \frac{c}{d}$.

Note that as the value of z approaches $-1/e$, the value of $W(z)$ approaches -1 but it never actually reaches there.

The function becomes increasingly steep, making its approximation via the Taylor series above increasingly inaccurate.

Hence for the smallest %1 of the range $-1/e \le z \le 0$:

- We precalculate a lookup table which maps 16 uniformly distributed values of z to values of $W(z)$
- During runtime, we calculate $W(z)$ as the weighted-average of $W(z_0)$ and $W(z_1)$, where $z_0 \le z < z_1$

For $1/e < z < 24 + 1/e$:

- We precalculate a lookup table which maps 128 uniformly distributed values of z to values of $W(z)$
- During runtime, we calculate $W(z)$ as the weighted-average of $W(z_0)$ and $W(z_1)$, where $z_0 \le z < z_1$

For $z \ge 24 + 1/e$, we rely on the fact that $W(z) \approx p - q + \frac{q^2+2pq-2q}{2p^2}$, where $p = \log(z)$ and $q = \log(p)$.

Since this method requires the calculation of $\log(log(z))$, it is actually applicable for as low as $z = e$.

However, for $e \le z < 24 + 1/e$, the previous method achieves better accuracy than this method.

Note that due to practical reasons, this library internally implements the calculation of $W(z)/z$ instead of $W(z)$.

This means that instead of calculating $\frac{W\left(\log\left(\frac{a}{b}\right)\cdot\frac{c}{d}\right)}{log\left(\frac{a}{b}\right)}$, it first calculates $\frac{W\left(\log\left(\frac{a}{b}\right)\cdot\frac{c}{d}\right)}{log\left(\frac{a}{b}\right)\cdot\frac{c}{d}}$, and then multiplies the result by $\frac{c}{d}$.