1 Background

The Golang source for the `log()` function[1] states that the natural log is computed in three parts

1.1 Argument Reduction

Find k and f such that

$$x = 2^k \left(1 + f\right)$$

where $\frac{\sqrt{2}}{2} < 1 + f < \sqrt{2}$.

1.2 Approximation of $\log(1 + f)$

Let $s = \frac{f}{2+f}$, based on

$$\begin{aligned} \log(1+f) &= \log(1+s) - \log(1-s) \\ &= 2s + \frac{2}{3}s^3 + \frac{2}{5}s^5 + \cdots \\ &= 2s + sR\left(s\right) \end{aligned}$$

Use a special Reme[2] algorithm on $[0, 0.1716]$ to generate a polynomial of degree 14 to approximate R.

This approximation is given by[3]

$$R(z) \approx L_1 s^2 + L_2 s^4 + L_3 s^6 + L_4 s^8 + L_5 s^{10} + L_6 s^{12} + L_7 s^{14}$$

with maximum error $2^{-58.45}$.

1.3 Combining

Finally

$$\begin{aligned} \log(x) &= k\log(2) + \log(1+f) \\ &= kH + \left(f - \left(\frac{f^2}{2} - \left(s\left(\frac{f^2}{2} + R\right) + kL\right)\right)\right) \end{aligned}$$

where

$$\log(2) = H + L$$

splits as high and low parts

$$\begin{aligned} H &= 2^{-1} \cdot \texttt{1.62e42fee00000}_{16} \\ L &= 2^{-1} \cdot \texttt{0.00000001a39ef35793c76}_{16} = 2^{-33} \cdot \texttt{1.a39ef35793c76}_{16} \\ \log(2) &= 2^{-1} \cdot \texttt{1.62e42fefa39ef}_{16} \end{aligned}$$

such that the high part satisfies nH is always exact for $|n| < 2000$.

[1] The Golang source actually mentions that this method and some of the comments were borrowed from original C code, from FreeBSD's `/usr/src/lib/msun/src/e_log.c`

[2] misspelling, should be Remez

[3] the z is likely s^2 which likely means the polynomial is intentionally degree 7 in z

2 Reducing Value

First note that

$$2^{-1} < 2^{-\frac{1}{2}} < 2^{0} < 2^{\frac{1}{2}} < 2^{1} \quad \text{i.e.} \quad \frac{1}{2} < \frac{\sqrt{2}}{2} < 1 < \sqrt{2} < 2.$$

In Golang the function `math.Frexp` takes a float x and returns an exponent e and fractional value m such that

$$x = 2^{e} m, \qquad 2\left|m\right| \in [1, 2)\,.$$

Since $\log(x)$ defined requires $x > 0$ we know $m > 0$. If $\frac{1}{\sqrt{2}} < m < 1$ then we set $k = e$ and $1 + f = m$. If $\frac{1}{2} \leq m \leq \frac{1}{\sqrt{2}}$ then we set $1 + f = 2m \in \left[1, \sqrt{2}\right]$ and $k = e - 1$ since $2^{e} m = 2^{e-1}(2m)$. This would appear to only give

$$\frac{1}{\sqrt{2}} < 1 + f \leq \sqrt{2}$$

but we also know $1 + f \neq \sqrt{2}$ since an irrational can't be represented in floating point.

3 From f to s

We write $1 + f = \dfrac{1 + s}{1 - s}$ so that

$$\begin{aligned} \log(1 + f) &= \log(1 + s) - \log(1 - s) \\ &= 2s + \frac{2}{3}s^{3} + \frac{2}{5}s^{5} + \cdots \end{aligned}$$

Solving for s in the above yields $s = \dfrac{f}{2 + f} = 1 - \dfrac{2}{2 + f}$ and

$$\frac{1}{\sqrt{2}} < 1 + f < \sqrt{2} \Longrightarrow -\left(3 - 2\sqrt{2}\right) < s < 3 - 2\sqrt{2} \approx 0.171573 < 0.1716.$$

3.1 Why the bounds?

The bound $\frac{1}{\sqrt{2}} < 1 + f < \sqrt{2}$ was just given but never justified. In fact, this interval is required to be able to use an approximation of $R(s)$. First, in order to factor $x = 2^{k}(1 + f)$ uniquely, if $1 + f \in [\alpha, \beta]$ we must have

$$\log_2 \beta = \log_2 \alpha + 1 \Longleftrightarrow \beta = 2\alpha.$$

Second, since defining $R(s)$ requires both $\log(1 \pm s)$ to be defined, we need $-1 < s < 1$.

Third, the symmetry $R(s) = R(-s)$ means we limit to some $-A < s < A$ (for $0 < A \leq 1$). This in turn means that

$$\frac{1}{B} < 1 + f < B, \qquad B = \frac{1 + A}{1 - A} \Longleftrightarrow A = \frac{B - 1}{B + 1}.$$

Hence we put $1 + f \in \left[B^{-1}, B\right]$ which forces

$$\log_2 B = \log_2\left(B^{-1}\right) + 1 \Longrightarrow 2\log_2 B = 1 \Longrightarrow B = 2^{\frac{1}{2}} = \sqrt{2} \Longrightarrow A = 3 - 2\sqrt{2}.$$

4 Remez Algorithm

We seek to approximate

$$R(s) = \frac{2s^{2}}{3} + \frac{2s^{4}}{5} + \frac{2s^{6}}{7} + \cdots = \frac{\log(1 + s) - \log(1 - s)}{s} - 2, \quad R(0) = 0$$

with a degree 14 polynomial that gives equi-oscillating errors. By construction $R(s) = R(-s)$, hence we need to be able to approximate $R(s)$ for $s \in \left[0, 3 - 2\sqrt{2}\right] \subset [0, 0.1716]$.

We manually force an equi-oscillating error at node points $s_0, s_1, \ldots, s_7$ by setting

$$R(s_j) = L_2 s_j^2 + L_4 s_j^4 + \cdots + L_{14} s_j^{14} + (-1)^j E = P(s_j) \pm E$$

and solving for the 7 unknown coefficients and the error E. This gives a system

$$\begin{bmatrix} s_0^2 & \cdots & s_0^{14} & 1 \\ s_1^2 & \cdots & s_1^{14} & -1 \\ \vdots & \ddots & & \vdots \\ s_7^2 & \cdots & s_7^{14} & -1 \end{bmatrix} \begin{bmatrix} L_2 \\ \vdots \\ L_{14} \\ E \end{bmatrix} = \begin{bmatrix} R(s_0) \\ R(s_1) \\ \vdots \\ R(s_7) \end{bmatrix}.$$

After solving, we exchange $s_0, s_1, \ldots, s_7$ for new points which maximize $|R(s) - P(s)|$ locally. The method terminates once the exchange process results in no change at all (or a minimal change).

An alternative approach only swaps a single s_j at a time. It finds the absolute extreme

$$s^* = \operatorname*{argmax}_{s \in \left[0, 3 - 2\sqrt{2}\right]} |R(s) - P(s)|$$

and then swap s^* with the nearest value among the $\{s_j\}$ and only terminates once the absolute extreme occurs among the $\{s_j\}$.

In either situation, if there is no exchange left to do

$$\max_{s \in \left[0, 3 - 2\sqrt{2}\right]} |R(s) - P(s)| = |R(s^*) - P(s^*)| = |\pm E| = E$$

and this equi-oscillating error will occur a maximal number of times in our interval.