Accuracy vs Efficiency of Numerical Methods

How to program a Calculator

Jake Darby

Abstract

This document will discuss and analyse various numerical methods for computing functions commonly found on calculators. The aim of this paper is to, for each set of functions, compare and contrast several algorithms in regards to their effiency and accuracy.

4 Trigonometric Functions

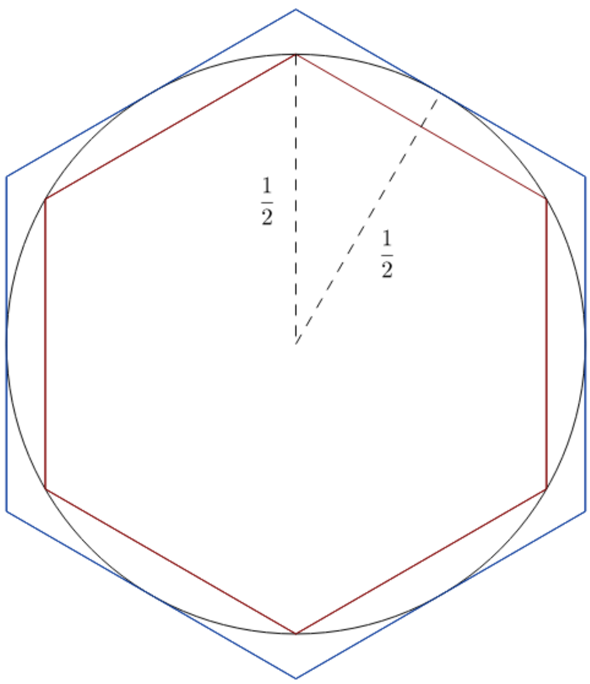
This section will focus on trigonometric functions, which are commonly used cyclic functions. These functions have been studied for hundreds of years, and can be challenging to calculate. We will discuss several methods of calculating them below before comparing methods.
TODO: Extend and Eloquate introduction

4.1 Calculating π

Several of the methods in this section require that we already know the value of π, for example when we are applying several trig identities. Here we will briefly discuss several methods for calculating the value of π, so that we may use this value in later subsections.

The first method to consider is the method used by ancient mathematicians, such as the Greeks and Chinese. We know that if the radius of the circle is $\frac{1}{2}$, then the circumference of the circle is π, and the value is between the perimeters of the inner and outer polygon perimeters. The internal perimeter is $p_n = n\sin(\frac{\pi}{n})$ and the external perimeter is $P_n = n\tan(\frac{\pi}{n})$.

Figure 4.1.1: Ancient method of calculating π



As we know the values of $\tan(\frac{\pi}{6})$ and $\sin(\frac{\pi}{6}$, then we can calculate P_6 and p_6. It has be shown that $P_{2n} = \frac{2p_nP_n}{p_n+P_n}$ and $p_{2n} = \sqrt{p_nP_{2n}}$, which allows us to create an iterative method to approximate π, by taking the mid-point of the successive polygon perimeters.

Other common historical methods for approximating π are to use infinite series. One such method uses the series expansion of $\tan^{-1}$, which is discussed in detail below, where $\tan^{-1}(1) = \frac{\pi}{4}$. This gives the following approximation using N terms:

$$\pi = 4\sum_{n=0}^{N}\frac{(-1)^n}{2n+1} = \sum_{n=0}^{N}\frac{8}{(4n+1)(4n+3)} \tag{4.1.1}$$

This sequence converges very slowly, with sublinear convergence, to the correct value. More modern methods have typically revolved around finding more rapidly converging infinite series, examples include Ramanujan's series:

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{n=0}^{\infty} \frac{(4n)!(1103 + 26390n)}{(k!)^n 396^{4n}} \tag{4.1.2}$$

or the Chudnovsky algorithm:

$$\frac{1}{\pi} = 12 \sum_{n=0}^{\infty} \frac{(-1)^n (6n)!(13591409 + 545140134n)}{(3n)!(n!)^3 640320^{3n+\frac{3}{2}}} \tag{4.1.3}$$

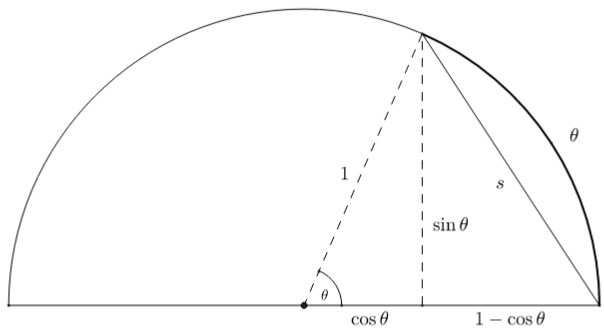
This final series is extremely rapidly convergant to the value of $\frac{1}{\pi}$, for example just the first term gives π accurate to 13 decimal places while we can get π accurate to 1000 decimal places with summing just 71 terms. Compared to Equation **??** which takes the summation of 500 terms to acheive the same 1000 digits of accuracy.

To get large degrees of accuracy for π is extremely computer intensive and using the `mpfr` requires the number of bits of precision and number of terms to be set. This makes calculating π to a large number of decimal places, for example 1000000, computationally infeasible on a regular home computer. Therefore for our purposes we will use the precalculated value of π to 1000000 decimal places as listed on `http://www.exploratorium.edu/pi/pi_archive/Pi10-6.html`

4.2 Geometric Method

The first method I will be discussing is a method based on geometric properties that are derived on a circle, and we will start by considering values of $\cos$ in the range $[0, \frac{\pi}{2}]$. To do this we will consider the following figure of the unit circle:

Figure 4.2.1: Diagram showing angles to be dealt with

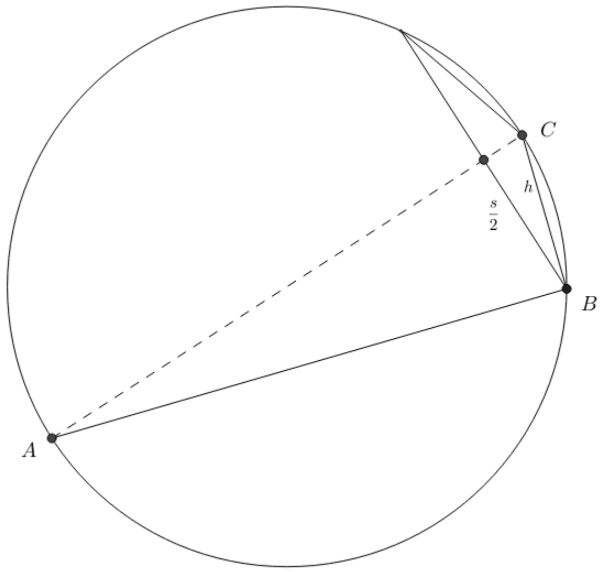


Here theta will be given in radians, and we can note that the labelled arc has length θ due the formula for the circumference of a circle. By using the following derivation we can find a formula for θ in terms of s:

$$\begin{aligned} s^2 &= \sin^2\theta + (1-\cos\theta)^2 \\ &= (\sin^2\theta + \cos^2\theta) + 1 - 2\cos\theta \\ &= 2 - 2\cos\theta && \text{By using } \sin^2\theta + \cos^2\theta = 1 \\ \cos\theta &= 1 - \frac{s^2}{2} \end{aligned}$$

We will now consider a second diagram which will allow us to calculate an approximate value of s.

Figure 4.2.2: Diagram detailing how to calculate s

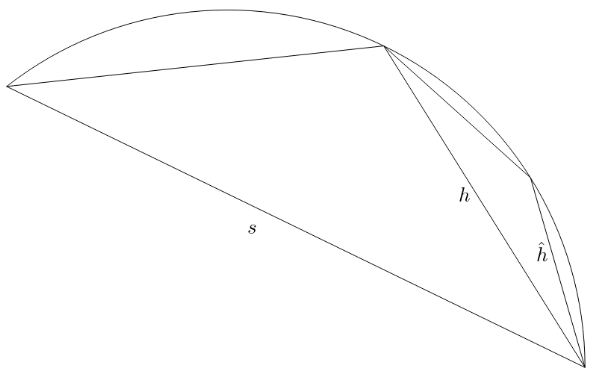


We will first note that by an elementary geometry result we can know that the angle ABC is a right-angle; also we can consider that h is an approximation of $\frac{\theta}{2}$, which will become relevant later. Now because AC is a diameter of our circle then it's length is 2 and thus, by utilising Pythagarus' Theorem, we get that the length of AB is $\sqrt{AC^2 - BC^2} = \sqrt{4 - h^2}$.

From here we consider the area of triangle ABC, which can be calculated as $\frac{1}{2} \cdot h \cdot \sqrt{4 - h^2}$ and as $\frac{1}{2} \cdot 2 \cdot \frac{s}{2}$; by equating these two, squaring both sides and re-arranging we get that $s^2 = h^2(4 - h^2)$. Now we have the basis for a method that will allow us to calculate $\cos\theta$.

To complete our method we will consider introducing a new line that is to h what h is to s as shown in the diagram below:

Figure 4.2.3: Detailing the recursive steps



It is easy to see that if we repeat the steps above we get that $h^2 = \hat{h}^2(4 - \hat{h}^2)$, and it also follows that $\hat{h} \approx \frac{\theta}{4}$. Using this we can take an initial guess of $h_0 := \frac{\theta}{2^k}$, for some $k \in \mathbb{N}$, and then calculate $h_{n+1}^2 = h_n^2(4 - h_n^2)$ where $n \in [0, k] \cap \mathbb{Z}$; finally we calculate $\cos\theta = 1 - \frac{h_k^2}{2}$, giving the following algorithm:

Algorithm 4.2.1: Geometric calculation of $\cos$

```
geometric_cos(θ ∈ [0, π/2], k ∈ ℕ)
    h_0 := θ / 2^k
    n := 0
    while n < K:
        h_{n+1}^2 := h_n^2 · (4 − h_n^2)
        n ↦ n + 1
    return 1 − h_k^2 / 2
```

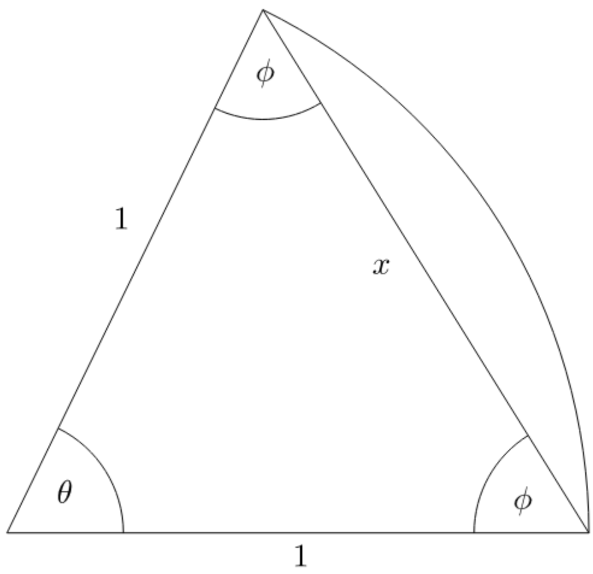
Now we can use the above pseudocode to calculate any trigonometric function value by using various trigonometric identities. First we suppose $\theta \in \mathbb{R}$, then we can repeatedly apply the identity $\cos\theta = \cos(\theta \pm 2\pi)$ to either add or subtrack 2π until we have a value $\theta' \in [0, 2pi)$. Once we have this value we can utilise the following assignment to calculate $\cos\theta$:

$$\cos\theta = \begin{cases} \cos\theta' & : \quad \theta' \in [0, \frac{\pi}{2}] \\ -\cos(\pi - \theta') & : \quad \theta' \in [\frac{\pi}{2}, \pi] \\ -\cos(\theta' - \pi) & : \quad \theta' \in [\pi, \frac{3\pi}{2}] \\ \cos(2\pi - \theta') & : \quad \theta' \in [\frac{3\pi}{2}, 2\pi) \end{cases}$$

Using Algorithm 4.2.1 we can also easily calculate both $\sin\theta$ and $\tan\theta$, by further use of trigonometric identities. In particular we note that $\sin\theta = \cos(\theta - \frac{\pi}{2}$ and $\tan\theta = \frac{\sin\theta}{\cos\theta}$. Hence we can now calculate the trigonometric function value of any angle.

We now wish to analyse the error of our approximation for $\cos$, as the other methods have errors that are derivative of the error for approximating $\cos$. Now Figure 4.2.4 shows an arc of a circle which creates chord x, with this we will be able to calculate the exact length of the chord and thus work on the error of our approximations.

Figure 4.2.4: Diagram to find actual arc approximation



To start we will note that $\phi = \frac{\pi-\theta}{2} = \frac{\pi}{2} - \frac{\theta}{2}$, and then by using the Sine Law we get

$$\frac{x}{\sin\theta} = \frac{1}{\sin\phi} \implies x = \frac{\sin\theta}{\sin\phi}$$

Now we can recall the double angle formula for $\sin$, which gives $\sin\theta = 2\sin\frac{\theta}{2}\cos\frac{\theta}{2}$, and also $\sin\phi = \cos\frac{\theta}{2}$. This allows us to see that

$$x = \frac{2\sin\frac{\theta}{2}\cos\frac{\theta}{2}}{\cos\frac{\theta}{2}} = 2\sin\frac{\theta}{2}$$

Therefore we see that h_n is approximating the chord length associated with angle $\theta 2^{n-k}$, and thus $\epsilon_n = |h_n - 2\sin(\theta 2^{n-k-1})|$. Now as $h_0 = \theta 2^{-k} \approx 2\sin(\theta 2^{-k-1})$ then if follows that $\exists\phi$ such that $h_0 = 2\sin(\phi 2{-}k-1)$, from this we can see that $\phi = 2^{k+1}\sin^{-1}(\theta 2^{-k-1})$. We will uses these facts to prove a couple of propositions.

Proposition 4.2.1. $h_n = 2\sin(\phi 2^{n-k-1}) \forall n \in [0,k]\cap\mathbb{Z}$ *where* $\phi := 2^{k+1}\sin^{-1}(\theta 2^{-k-1})$.

Proof. Proceed by induction on $n \in [0,k]\cap\mathbb{Z}$.

$\mathbf{H}(n)$**:** $h_n = 2\sin(\phi 2^{n-k-1})$

$\mathbf{H}(0)$**:**

$$\begin{aligned} 2\sin(\phi 2^{-k-1}) &= 2\sin(\sin^{-1}(\phi 2^{-k-1})) \\ &= \theta 2^{-k} \\ &= h_0 && \text{by definition of } h_0 \end{aligned}$$

$\mathbf{H}(n) \implies \mathbf{H}(n+1)$**:**

$$\begin{aligned} h_{n+1} &= h_n\sqrt{4-h_n^2} \\ &= 2\sin(\phi 2^{n-k-1})\sqrt{4-4\sin^2(\phi 2^{n-k-1})} && \text{by H}(n) \\ &= 4\sin(\phi 2^{n-k-1})\cos(\phi 2^{n-k-1}) \\ &= 2\sin(\phi 2^{n-k}) && \text{by the use of double angle formulas} \end{aligned}$$

□

Proposition 4.2.2. $h_n > 2\sin(\theta 2^{n-k-1}) \forall n \in [0,k]\cap\mathbb{Z}$

Proof. We start by considering the expansion of the exact value of h_n.

$$\begin{aligned} h_n &= 2\sin(\phi 2^{n-k-1}) \\ &= 2\sin(2^{n-k-1}(2^{k+1}\sin^{-1}(\theta 2^{-k-1}))) \\ &= 2\sin(2^n\sin^{-1}(\theta 2^{-k-1})) \\ &= 2\sin(\theta 2^{n-k-1} + \tfrac{1}{6}\theta^3 2^{n-3k-3} + \mathcal{O}(2^{-5k})) && \text{Detailed in section } \textbf{??} \end{aligned}$$

Now as we know that $n \le k$, then it follows that $\theta 2^{n-k-1} \le \frac{1}{2}\theta$.

Also as $\theta \leq \frac{\pi}{2}$ we know that $\theta 2^{n-k-1} \leq \frac{\pi}{4}$.

We can also show that $\frac{1}{6}\theta^3 2^{n-3k-3} + \mathcal{O}(2^{-5k}) \leq \frac{\pi}{4}$, though the proof is ommited here for brevity; therefore we see that $\phi 2^{n-k-1} \leq \frac{\pi}{2}$, and obviously that $\phi 2^{n-k-1} > \theta 2^{n-k-1}$.

Hence, as $\sin$ is an increasing function in the range $[0, \frac{\pi}{2}]$, we conclude that

$$h_n = 2\sin(\phi 2^{n-k-1}) > 2\sin(\theta 2^{n-k-1})$$

. $\square$

With these two propositions we can now consider the error of our approximation of $\cos$. First we will prove the following proposition regarding the error of the approximation of s:

Proposition 4.2.3. *If* $\epsilon_n := |h_n - 2\sin(\theta 2^{n-k-1})| \forall n \in [0, k] \cap \mathbb{Z}$, *then* $\epsilon_k < 2^k \epsilon_0)$.

Proof. $\epsilon_n = h_n - 2\sin(\theta 2^{n-k-1})$ as $h_n > 2\sin(\theta 2^{n-k-1})$ by Proposition 4.2.2.

Now we see that:

$$\begin{aligned}
\epsilon_{n+1} &= h_{n+1} - 2\sin(\theta 2^{n-k}) \\
&= h_n\sqrt{4 - h_n^2} - 4\sin(\theta 2^{n-k-1})\cos(\theta 2^{n-k-1})
\end{aligned}$$

If we consider the equation $\alpha\beta - \gamma\delta = (\alpha - \gamma) + \alpha(\beta - 1) - \gamma(\delta - 1)$ and apply it to our current formula we get:

$$\begin{aligned}
\epsilon_{n+1} &= (h_n - 2\sin(\theta 2^{n-k-1})) + h_n(\sqrt{4 - h_n^2} - 1) - 2\sin(\theta 2^{n-k-1})(2\cos(\theta 2^{n-k-1}) - 1) \\
&= \epsilon_n + h_n(\sqrt{4 - h_n^2} - 1) - 2\sin(\theta 2^{n-k-1})(2\cos(\theta 2^{n-k-1}) - 1) \\
&= 2\epsilon_n + h_n(\sqrt{4 - h_n^2} - 2) - 2\sin(\theta 2^{n-k-1})(2\cos(\theta 2^{n-k-1}) - 2) \\
&= 2\epsilon_n + h_n(\sqrt{4 - h_n^2} - 2) + 2\sin(\theta 2^{n-k-1})(2 - 2\cos(\theta 2^{n-k-1})) \\
&< 2\epsilon_n + h_n(\sqrt{4 - h_n^2} - 2\cos(\theta 2^{n-k-1})) \\
&< 2\epsilon_n + h_n(\sqrt{4 - 4\sin^2(\theta 2^{n-k-1})} - 2\cos(\theta 2^{n-k-1})) \\
&= 2\epsilon_n + h_n(2\cos(\theta 2^{n-k-1}) - 2\cos(\theta 2^{n-k-1})) \\
&= 2\epsilon_n
\end{aligned}$$

The inequalities in the above derivation arrise from the fact that $h_n > 2\sin(\theta 2^{n-k-1})$ by Proposition 4.2.2.

Hence as we now know that $\epsilon_{n+1} < 2\epsilon_n$, we then see that $\epsilon_n < 2^n \epsilon_0$. Therefore we prove our statement that

$$\epsilon_k < 2^k \epsilon_0$$

$\square$

Obviously $\epsilon_k = |h_k - s|$, and we can now use this to find the error of our final answer. First we will start by letting $\mathcal{C} := 1 - \frac{1}{2}h_k^2$ and note that analytically $cos\theta = 1 - \frac{1}{2}s^2$. Therefore we will now consider $\epsilon_{\mathcal{C}} = |\mathcal{C} - \cos(\theta)|$:

$$
\begin{aligned}
\epsilon_{\mathcal{C}} &= |1 - \frac{h_k^2}{2} - 1 + \frac{s^2}{2}| \\
&= \frac{1}{2}|h_k^2 - s^2| \\
&= \frac{1}{2}|h_k h_k - 2\sin(\frac{\theta}{2})2\sin(\frac{\theta}{2})| \\
&= \frac{1}{2}(h_k h_k - 2\sin(\frac{\theta}{2})2\sin(\frac{\theta}{2})) && \text{as } 2\sin(\frac{\theta}{2}) < h_k \\
&= \frac{1}{2}(2\epsilon_k + h_k(h_k - 2) - 2\sin(\frac{\theta}{2})(2\sin(\frac{\theta}{2}) - 2) \\
&< \frac{1}{2}(2\epsilon_k + h_k(h_k - 2\sin(\frac{\theta}{2}))) \\
&= \frac{1}{2}(2 + h_k)\epsilon_k \\
&= \frac{1}{2}(2 + 2\sin(\frac{\phi}{2}))\epsilon_k \\
&= (1 + \sin(\frac{\phi}{2}))\epsilon_k \\
&\leq 2\epsilon_k
\end{aligned}
$$

As $\epsilon_{\mathcal{C}} \leq 2\epsilon_k$, then by Proposition 4.2.3 we see that $\epsilon_{\mathcal{C}} < 2^{k+1}\epsilon_0$. Now to consider ϵ_0 we first observe that $\epsilon_0 = \theta 2^{-k} - 2\sin\theta 2^{-k-1}$, and therefore we can conclude that:

$$\epsilon_{\mathcal{C}} < 2\theta - 2^{k+2}\sin(\theta 2^{-k-1})$$

If we then wish to calculate $\cos\theta$ accurate to N decimal places then we are looking to find $k \in \mathbb{N}$ such that

$$2\theta - 2^{k+2}\sin(\theta 2^{-k-1}) < 10^{-N} \implies 2^{k+2}\sin(\theta 2^{-k-1}) > 2\theta - 10^{-N}$$

For an example of the above in action we will be taking $\theta = 0.5$. The table below shows the minimum $k \in \mathbb{N}$ to guarantee N digits of accuracy in the result:

N	k
5	6
10	14
50	80
100	163
1000	1658

As can be seen the value of k required to acheive N digits of accuracy increases roughly linearly when $\theta = 0.5$. Testing for other values of θ reveals them to have similar required values for k, at least within the same order of each other.

Another consideration for Algorithm 4.2.1 is that we could "run it in reverse" to attain an algorithm for the inverse cosine function. To start take line 7 which is $\mathcal{C} = 1 - \frac{1}{2}h_k^2$, which can be re-arranged to give $h_k^2 = 2 - 2\mathcal{C}$, where we know $\mathcal{C}$ as our initial value.

Line 5 is a little more difficult, but by re-arranging we see that $h_n^4 - 4h_n^2 + h_{n+1}^2 = 0$, which can be solved via the quadratic formula to give $h_n^2 = 2 \pm \sqrt{4 - h_{n+1}^2}$. Now we can make the observation that if $x \in \mathbb{R}_0^+$, then $\cos^{-1}(-x) = \pi - \cos^{-1}(x)$ and so we can restric our algorithm to only consider $x \in [0,1]$. With this we know that $\theta \in [0, \frac{\pi}{2}]$, and thus $h_k \leq \sqrt{2}$. Therefore as $h_{n+1} > h_n \forall n \in [0, k-1] \cap \mathbb{Z}$ we see that $h_n^2 \leq 2 \forall n \in [0,k] \cap \mathbb{Z}$. This allows us to ascertain that to reverse Line 5 we perform $h_n^2 = 2 - \sqrt{4 - h_{n+1}^2}$.

Finally line 2 is reversed by returning the value $2^k h_0$; therefore we get the following algorithm for $\cos^{-1}(x)$ where $x \in [0,1]$:

Algorithm 4.2.2: Geometric calculation of $\cos^{-1}$

```
1  geometric_aCos(x ∈ [0,1], k ∈ ℕ)
2      h_k := 2 − 2x
3      n := k − 1
4      while n ≥ 0:
5          h_n^2 := 2 − √(4 − h_{n+1}^2)
6          n ↦ n − 1
7      return 2^k h_0
```

Similar to the regular trigonometric functions we can use trigonometric identities to calculate the inverse trigonometric functions from $\cos^{-1}$. To start we recall that $\cos^{-1}(-x) = -\cos(x)$ where $x \in [0,1]$, then we can use the identities that $\sin^{-1}(x) = \frac{\pi}{2} - \cos^{-1}(x)$ and $\tan^{-1}(x) = \sin^{-1}(\frac{x}{\sqrt{x^2+1}})$.

If we suppose that all operations in the method are accurately computed then Algorithm 4.2.2 is a computation with high accuracy. This is because there is no initial guess, such as in Algorithm 4.2.1, and so the only introduction of error is assuming that $2^k h_0 \approx \theta$. However as we discuss in detail in Section **??**, calculating square roots is not a simple task and thus will introduce error to the method in general; therefore the accuracy of the method is roughly as accurate as our method of calculating square roots.

4.3 Taylor Series

If we consider our definition of a McClaurin Series from Section **??**, we can use this to approximate our Trigonometric Functions. Consider first $\cos\theta$, for which we know that $\frac{d}{d\theta}\cos\theta = -\sin\theta$; it then follows that $\frac{d^2}{d\theta^2}\cos\theta = -\cos\theta$, $\frac{d^3}{d\theta^3}\cos\theta = \sin\theta$ and $\frac{d^4}{d\theta^4}\cos\theta = \cos\theta$.

If we let $f(x) = \cos x$ and use the known values $\cos(0) = 1$ and $\sin(0) = 0$, then we see that:

$$f^{(n)}(0) = \begin{cases} 1 & : \ 4 \mid n \\ 0 & : \ 4 \mid n-1 \\ -1 & : \ 4 \mid n-2 \\ 0 & : \ 4 \mid n-3 \end{cases}$$

By simplifying this by ommitting the 0 coefficient terms we get the following series:

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots \tag{4.3.1}$$

By using similar working we can get that the series associated with $\sin x$):

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots \tag{4.3.2}$$

Before we go any further we need to consider when Equations 4.3.1 and 4.3.2 converge to their respective functions. To do this we will use the ratio test for series as defined in **??**, using Equation 4.3.1 we see that

$$\begin{aligned}
L_{\mathcal{C}} &= \lim_{n\to\infty} \left| \frac{a_{n+1}}{a_n} \right| \\
&= \lim_{n\to\infty} \left| \frac{\frac{(-1)^{n+1}}{(2n+2)!} x^{2n+2}}{\frac{(-1)^n}{(2n)!} x^{2n}} \right| \\
&= \frac{(2n)!}{(2n+2)!} |x|^2 \\
&= \frac{1}{(2n+2)(2n+1)} |x|^2
\end{aligned}$$

Now it is easy to see that, $L_{\mathcal{C}} = 0$ for all values of x as the fractional component decreases as n increases and $|x|^2$ is a constant. Therefore we can conclude that Equation 4.3.1 converges to $\cos(x)$ for all values of x. We can use a very similar deduction to show that Equation 4.3.2 converges to $\sin(x)$ for all values of x.

The above means that $\cos$ and $\sin$ can be approximated using Taylor Polynomials, in particular for a given $N \in \mathbb{N}$:

$$\cos x \approx \textstyle\sum_{n=0}^{N} \frac{(-1)^n}{(2n)!} x^{2n} \quad \text{and} \quad \sin x \approx \textstyle\sum_{n=0}^{N} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

This allows us to create the following two methods for computing $\cos x$ and $\sin x$:

Algorithm 4.3.1: Taylor computation of $\cos$ and $\sin$

```
taylor_cos(x ∈ ℝ, N ∈ ℕ)
    C := 0
    n := 0
    while n < N:
        C ↦ C + (−1)^n · 1/(2n)! x^(2n)
        n ↦ n + 1
    return C

taylor_sin(x ∈ ℝ, N ∈ ℕ)
    S := 0
    n := 0
    while n < N:
        S ↦ S + (−1)^n · 1/(2n+1)! x^(2n+1)
        n ↦ n + 1
    return S
```

As these two methods are obviously very similar and the fact that $\sin(x) = \cos(x - \frac{\pi}{2})$, we will continue by examining only the taylor method for approximating $\cos$. We will assume that any calculations for $\sin$ are transformed into a problem of finding a $\cos$ value.

It should be noted that this $\cos$ algorithnm is particularly inefficient to calculate on a computer implementation; this is primarily due to the way in which the update of $\mathcal{C}$ is calculated each loop.

In each loop we are calculating x^{2n}, which has a naieve complexity of $\mathcal{O}(2n)$. However what we are actually calculating $x^{2(n-1)} \cdot x^2$ and thus if we store the values of $x^{2(n-1)}$ and x^2, the complexity of this step drops to $\mathcal{O}(1)$. Similarly we are also calculating $\frac{1}{(2n)!}$ in each loop which, by the same logic, is $\frac{1}{2(n-1)!} \cdot \frac{1}{(2n)(2n-1)}$, and we can use the same storage and update method as for x^{2n}.

As another step towards optimizing the algorithm we can start with an initial value of $\mathcal{C} = 1$, and then perform two updates of $\mathcal{C}$ each loop until we reach or surpass N. This saves calculating $(-1)^n$ each loop, by explicitly performing two different calculations. Implementing all of the above gives us the following two updated methods:

Algorithm 4.3.2: Taylor computation of $\cos$ optimised

taylor_cos $(x \in \mathbb{R}, N \in \mathbb{N})$
    $\mathcal{C} := 1$
    $x_2 := x^2$
    $a := 1$
    $b := 1$
    $n := 1$
    while $n < N$:
        $a \mapsto a \cdot \frac{1}{(2n-1)(2n)}$
        $b \mapsto b \cdot x_2$
        $\mathcal{C} \mapsto \mathcal{C} - a \cdot b$
        $a \mapsto a \cdot \frac{1}{(2n+1)(2n+2)}$
        $b \mapsto b \cdot x_2$
        $\mathcal{C} \mapsto \mathcal{C} + a \cdot b$
        $n \mapsto n + 2$
    return $\mathcal{C}$

As the next term of the polynomial is known definitively then we can see that it is very easy to calculate the error of our approximation. We see that

$$\begin{aligned}
\epsilon_N &= |\cos(x) - \text{taylor_cos(x, N)}| \\
&= \mathcal{O}(|x|^{N'+1}) && \text{where } N' \text{ is the smallest} \\
& && \text{odd integer such that } N' \geq N \\
&\leq \frac{1}{(2(N'+1))!}|x|^{N'+1} \\
&\leq \frac{1}{(2(N+1))!}|x|^{N+1}
\end{aligned}$$

If we place bounds on the value of $\cos$ calculated as in Section 4.2, then we know that $|x| \leq \frac{\pi}{2}$, and thus we get the following bound for the error of our approximation:

$$\epsilon_N \leq \frac{\pi^{N'+1}}{2^{N'+1}(2(N'+1))!}$$

Thus if we find $N \in \mathbb{N}$ such that $\frac{\pi^N+1}{2^{N+1}(2(N+1)!)} < \tau \in \mathbb{R}+$ then we know that $\epsilon_N < \tau$. If we consider $\tau = 10^k$, then we can find $N \in \mathbb{N}$ such that our approximation is accurate to k decimal places. Below is a table which details some values of k and the corresponding minimum N to guarantee k decimal places of accuracy:

k	N
5	4
10	7
50	21
100	36
1000	233

Now for $\tan x$ we can either calculate both $\sin x$ and $\cos x$ using taylor_cos(x,N) and divide the resulting value, or we can calculate $\tan x$ directly using a Taylor expansion.

In calculating the McClaurin series for $\tan x$ we start by letting $\tan x = \sum_{n=0}^{\infty} a_n x^n$, and then noting that as $\tan x$ is an odd series then it's McClaurin series only contains non-zero coefficients for odd powers of x; therefore we get that $\tan x = \sum_{n=0}^{\infty} a_{2n+1} x^{2n+1} = a_1 x + a_3 x^3 + a_5 x^5 + \cdots$.

Next we consider that $\frac{d}{dx} \tan x = 1 + \tan^2 x$, and knowing the McClaurin series form of $\tan x$ we get the following:

$$\begin{aligned}\sum_{n=0}^{\infty}(2n+1)a_{2n+1}x^{2n} &= 1 + (\sum_{n=0}^{\infty} a_{2n+1}x^{2n+1})^2 \\ &= 1 + a_1^2 x^2 + (2a_1 a_3)x^4 + (2a_1 a_5 + a_3^2)x^6 + \cdots\end{aligned}$$

Considering the co-efficients of powers on the right hand side of the above equation we see that $2a_1a_3 = a_1a_3 + a_{3a1} = a_1a_{4-1} + a_3a_{4-3}$ and $2a_1a_5 + a_3^2 = a_1a_5 + a_3a_3 + a_5a_1 = a_1a_{6-1} + a_3a_{6-3} + a_5a_{6-5}$. This indicates that our general form for the co-efficient of $2n$ on the right hand side is $\sum_{k=1}^{n} a_{2k-1}a_{2n-2k+1}$, and thus returning to our equation we get

$$a_1 + \sum_{n=1}^{\infty}(2n+1)a_{2n+1}x^{2n} = 1 + \sum_{n=1}^{\infty}(\sum_{k=1}^{n} a_{2k-1}a_{2n-2k+1})x^{2n}$$

Using this we conclude that $a_1 = 1$ and $a_{2n+1} = \frac{1}{2n+1}\sum_{k=1}^{n} a_{2k-1}a_{2n-2k+1} \forall n \in \mathbb{N}$. We can note immediately that the calculation of any previous co-efficients will provide no help in calculating later co-efficients and so the entire sum must be calculated each loop, while also storing each co-efficient already calcualted.

This means that the complexity to calculate co-efficient a_{2n+1} is $\mathcal{O}(n)$ and will be the n^{th} such calculation, making the complexity of calculating n co-efficients to be $\mathcal{O}(n^2)$. Comparing this to the taylor_cos method we see that to calculate up to n co-efficients of both $\cos$ and $\sin$ has complexity $\mathcal{O}(n)$. Therfore it is more efficient to calculate $\tan$ by calculating both $\cos$ and $\sin$ using Algorithm **??**, and performing division than directly using Taylor Polynomial approximation.

We would also like to be able to calculate the inverse trigonometric functions using this method, which means we need to find our McClaurin series of the inverse trigonometric functions. The simplest of these is $\tan^{-1}$, where we start by recalling that $\frac{d}{dx}\tan^{-1}x = \frac{1}{1+x^2}$ and then by intergrating both sides we get:

$$\begin{aligned}
\tan^{-1} x &= \int \frac{1}{1+x^2} dx \\
&= \int (1-(-x^2))^{-}1 dx \\
&= \int \sum_{n=0}^{\infty} (-x^2)^n dx \qquad \text{by Equation } \textbf{??} \\
&= \int \sum_{n=0}^{\infty} (-1)^n x^{2n} dx \\
&= c + \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1}
\end{aligned}$$

As $\tan^{-1}(0) = 0$ then we see that $c = 0$ and thus gives us the following formula for $\tan^{-1}$:

$$\tan^{-1} x = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1}$$

Now due to the restrictions from Equation **??** the above is only valid for $x \in [-1, 1]$, but we know that the domain of $\tan^{-1}$ is $x \in \mathbb{R}$. To fix this we will first recognise that $\tan^{-1}(-x) = -\tan^{-1}(x)$, so we can restric our problem to $x \in \mathbb{R}_0^+$. Now if we take the double angle formula for $\tan$:

$$\tan(\alpha+\beta) = \frac{\tan(\alpha)+\tan(\beta)}{1-\tan(\alpha)\tan(\beta)}$$

By substituting $\alpha = \tan^{-1}(x)$ and $\beta = \tan^{-1}(x)$ into the above then we get

$$\tan^{-1}(x) + \tan^{-1}(y) = \tan^{-1}\left(\frac{x+y}{1-xy}\right)$$

Using this, suppose we are looking for $\tan^{-1}(z)$ where $z \in (1, \infty)$ and let $y = 1$, then $\tan^{-1}(y) = \frac{\pi}{4}$. We can then re-arrange the equation $z = \frac{x+1}{1-x}$ to get $x = \frac{z-1}{z+1}$; finally as $z > 1$, then $0 < x < 1$. This allows us to calculate:

$$\tan^{-1}(z) = \frac{\pi}{4} + \tan^{-1}\left(\frac{z-1}{z+1}\right)$$

In the above the calculated value is in the range $[0, 1]$ and so it is valid to use a Taylor polynomial using our McClaurin series above. This gives the following method

Algorithm 4.3.3: Taylor Method for $\tan^{-1}$

taylor_aTan$(x \in [0,1], N \in \mathbb{N})$
    $\mathcal{T} := 0$
    $x_2 := x^2$

```
    y := x
    n := 0
    while n < N:
        T ↦ T + 1/(2n+1) y
        y ↦ y · x_2
        T ↦ T − 1/(2n+2) y
        y ↦ y · x_2
        n ↦ n + 2
    return T
```

Similar to Algorithm **??** the error of Algorithm 4.3.3 is easy to calculate. We see that

$$\begin{aligned}
\epsilon_N &= |\tan^{-1}(x) - \text{taylor_aTan(x, N)}| \\
&\leq \frac{1}{2N+3}|x|^{2N+3} \\
&\leq \frac{1}{2N+3} && \text{as } x \leq 1
\end{aligned}$$

The next function we will consider is $\sin^{-1}$, which starts it's derivation in much the same way as $\tan^{-1}$. First we start by recalling that $\frac{d}{dx}\sin^{-1}(x) = (1-x^2)^{-\frac{1}{2}}$, then by taking integrals of both sides we get the following derivation:

$$\begin{aligned}
\sin^{-1}(x) &= \int (1-x^2)^{-\frac{1}{2}} dx \\
&= \int \sum_{n=0}^{\infty} \binom{-\frac{1}{2}}{n} (-x^2)^n \\
&= c + \sum_{n=0}^{\infty} (-1)^n \left(\prod_{k=1}^{n} \frac{-\frac{1}{2} - k + 1}{k} \right) \frac{x^{2n+1}}{2n+1} \\
&= c + \sum_{n=0}^{\infty} \frac{(-1)^n}{n!(2n+1)} \left(\prod_{k=1}^{n} \tfrac{1}{2} - k \right) x^{2n+1} \\
&= c + \sum_{n=0}^{\infty} \frac{(-1)^{2n}}{n!(2n+1)} \left(\prod_{k=1}^{n} \frac{2k-1}{2} \right) x^{2n+1} \\
&= c + \sum_{n=0}^{\infty} \frac{1}{n!(2n+1)2^n} \left(\prod_{k=1}^{n} 2k-1 \right) x^{2n+1} \\
&= c + \sum_{n=0}^{\infty} \frac{1}{n!(2n+1)2^n} (1 \times 3 \times 5 \times \cdots \times (2n-1)) x^{2n+1} \\
&= c + \sum_{n=0}^{\infty} \frac{1}{n!(2n+1)2^n} \times \frac{1 \times 2 \times 3 \times \cdots \times (2n)}{2 \times 4 \times \cdots \times (2n)} x^{2n+1} \\
&= c + \sum_{n=0}^{\infty} \frac{(2n)!}{(n!)^2(2n+1)4^n} x^{2n+1}
\end{aligned}$$

As $\sin^{-1}(0) = 0$ then we see that $c = 0$. Because the above is valid for $x \in (-1, 1)$, and we know the values of $\sin^{-1}(-1)$ and $\sin^{-1}(1)$, then we can have the following method for evaluating $\sin^{-1}$:

Algorithm 4.3.4: Taylor Method for $\sin^{-1}$

```
taylor_aSin(x ∈ [−1, 1], N ∈ ℕ)
    if x = 1:
        return π/2
    if x = −1:
        return −π/2
    S := x
    x_2 := x^2
    y := x
    a := 1
    b := 1
    c := 1
    n := 1
    while n < N:
        a ↦ 2n · (2n − 1) · a
        b ↦ n^2 · b
        c ↦ 4 · c
        y ↦ x_2 · y
        S ↦ S + a/(b·c·(2n+1)) · y
        n ↦ n + 1
    return S
```

The error for this method is similar to the $\tan^{-1}$ method, in that $\epsilon_N \leq \frac{(2(N+1))!}{((N+1)!)(2N+1)4^{N+1}}$. Finally we note that $\cos^{-1}(x) = \frac{\pi}{2} - \sin^{-1}(x)$, and thus can be calculated from a value calculated with Algorithm 4.3.4.

4.4 CORDIC

CORDIC is an algorithm that stands for COrdinate Rotation DIgital Computer and can be used to calculate many functions, including Trigonometric Values. The CORDIC algorithm works by utilising Matrix Rotations of unit vectors. This algorithm is less accurate than some other methods but has the advantage of being able to be implemented for fixed point real numbers in efficient ways using only addition and bitshifting.

CORDIC works by taking an initial value of $\mathbf{x}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ which can be rotated through an anti-clockwise angle of γ by the matrix

$$\begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix} = \frac{1}{\sqrt{1+\tan\gamma^2}} \begin{pmatrix} 1 & -\tan\gamma \\ \tan\gamma & 1 \end{pmatrix}$$

By taking taking smaller and smaller values of γ we can create an iterative process to find $\mathbf{x}_n$ which converges, for a given $\beta \in (-\frac{\pi}{2}, \frac{\pi}{2})$, to

$$\begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix}$$

To do this we repreately add and subtract our values for γ from β to bring it as close to 0 as possible. For our purposes we wish to have a sequence $(\gamma_k : k \in [0, n] \cap \mathbb{Z})$ which will allow us to construct all angles in the range $(-\frac{\pi}{2}, \frac{\pi}{2})$ to within a known level of accuracy. There are many possible choices here, but we wish to consider $(\gamma_k : k \in [0, n] \cap \mathbb{Z})$ such that $\tan \gamma_k = 2^{-k} \forall k \in [0, n] \cap \mathbb{Z}$.

We can note that the powers of 2 have a useful property, in that if $m > n \in \mathbb{N}$ we see that $\sum_{k=n}^{m-1} 2^k = 2^m - 2^n$. We wish to show that our choice for γ_k have a similar property which will be usefull in showing that they are a good choice for our CORIC algorithm.

Proposition 4.4.1. *If $m \in \mathbb{Z}_0^+$ and $n \in \mathbb{Z}^+$ such that $m > n$ and $\gamma_k = \tan^{-1}(2^-k) \forall k \in \mathbb{Z}_0^+$, then $\gamma_m < \gamma_n + \sum_{k=m+1}^{n} \gamma_k$.*

Proof. We know that $2^{-m} = 2^{-n} + \sum_{k=m+1}^{n} 2^-k$, and thus by applying $\tan^{-1}$ to both sides we get:

$$\tan^{-1} 2^{-m} = \gamma_m = \tan^{-1}(2^{-m-1} + 2^{-m-2} + \cdots + 2^{-n} + 2^{-n})$$

Let $a := 2^{-m-1} + 2^{-m-2} + \cdots + 2^{-n} + 2^{-n}$ and $b := 2^{-m-2} + \cdots + 2^{-n} + 2^{-n}$. Obviously $a < b$ and further we know that $\tan^{-1}$ is continuous on $[a, b]$ and differentiable on (a, b). Therefore we can apply the Mean Value Theorem from calculas to find that

$$\exists c \in (a, b) : \frac{1}{c^2 + 1} = \frac{\tan^{-1}(b) - \tan^{-1}(a)}{b - a}$$

By re-arranging we see that

$$\begin{aligned} \tan^{-1}(b) &= \frac{2^{-m-1}}{c^2 + 1} + \tan^{-1}(a) \\ &< \frac{2^{-m-1}}{2^{-2m-2} + 1} + \tan^{-1}(a) \end{aligned}$$

It can be shown, by considering the series expansion of $\tan^{-1}(2^{-m-1})$, that $\frac{2^{-m-1}}{2^{-2m-2}+1} < \tan^{-1}(2^{-m-1}) \forall m \in \mathbb{Z}_0^+$; therefore we get that:

$$\tan^{-1}(b) < \tan^{-1}(2^{-m-1}) + \tan^{-1}(a)$$

Following this an using the assumed value of γ_{m+1}, we see that:

$$\gamma_m < \gamma_{m+1} + \tan^{-1}(2^{-m-2} + \cdots + 2^{-n} + 2^{-n})$$

By repeating the above process we eventually see that:

$$\gamma_m < \sum_{k=m+1}^{n-1} \gamma_k + \tan^{-1}(2^{-n} + 2^{-n})$$

In a similar manner we can repeat the above process with $a := \tan^{-1}(2^{-n})$ and $b := \tan^{-1}(2^{-n} + 2^{-n})$. This will show that:

$$\gamma_m < \gamma_n + \sum_{k=m+1}^{n} \gamma_n$$

□

Using the previous proposition we can then show that our γ_k have the property that every angle in $(-\frac{\pi}{2}, \frac{\pi}{2})$ can be approximated by either adding or subtracting successive γ_k to within a tolerance of γ_n.

Proposition 4.4.2. *If* $\gamma_k = \tan^{-1}(2^-k) \forall k \in \mathbb{Z}$, *then for any* $n \in \mathbb{N}$

$$\exists\, (c_k \in \{-1, 1\} : k \in [0, n] \cap \mathbb{Z}) \ : \ |\beta - \sum_{k=0}^{n} c_k \gamma_k| \leq \gamma_n \quad \forall\, \beta \in (-\frac{\pi}{2}, \frac{\pi}{2})$$

Proof. We let $\beta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ and then will proceed by induction on $n \in \mathbb{N}$.

$\mathbf{H}(n)$**:** $\exists (c_k \in -1, 1 : k \in [0, n] \cap \mathbb{Z}) : |\beta - \sum_{k=0}^{n} c_k \gamma_k| \leq \gamma_n$

$\mathbf{H}(0)$**:** We have 4 cases to consider:

Case $\beta \in [0, \frac{\pi}{4})$**:** In this case $-\frac{\pi}{4} \leq \beta - \gamma_0 < 0$
Therefore $|\beta - \gamma_0| \leq \gamma_0$.

Case $\beta \in [\frac{\pi}{4}, \frac{\pi}{2})$**:** In this case $0 \leq \beta - \gamma_0 < \frac{\pi}{4}$
Therefore $|\beta - \gamma_0| \leq \gamma_0$.

Case $\beta \in (-\frac{\pi}{4}, 0)$**:** In this case $0 < \beta + \gamma_0 < \frac{\pi}{4}$
Therefore $|\beta - \gamma_0| < \gamma_0$.

Case $\beta \in (-\frac{\pi}{2}, -\frac{\pi}{4}$:] In this case $-\frac{\pi}{4} < \beta - \gamma_0 \leq 0$
Therefore $|\beta - \gamma_0| < \gamma_0$.

Therefore we see that $\mathrm{H}(0)$ holds true.

$\mathbf{H}(n) \implies \mathbf{H}(n+1)$**:**
By $\mathrm{H}(n)$ $\exists (c_k \in -1, 1 : k \in [0, n] \cap \mathbb{Z}) : |\beta - \sum_{k=0}^{n} c_k \gamma_k| \leq \gamma_n$; so let $\beta_n := \beta - \sum_{k=0}^{n} c_k \gamma_k$.
By Proposition 4.4.1 we know that $\gamma_n < 2\gamma_{n+1}$, and so we can proceed by case analysis:

Case $\beta_n \in [0, \gamma_{n+1})$**:**
$-\gamma_{n+1} \leq \beta_n - \gamma_{n+1} < 0 \implies |\beta - \sum_{k=0}^{n+1} c_k \gamma_k| \leq \gamma_{n+1}$ where $c_{n+1} = -1$.

Case $\beta_n \in [\gamma_{n+1}, \gamma_n)$**:**
$0 \leq \beta_n - \gamma_{n+1} < \gamma_{n+1} \implies |\beta - \sum_{k=0}^{n+1} c_k \gamma_k| \leq \gamma_{n+1}$ where $c_{n+1} = -1$.

Case $\beta_n \in [-\gamma_{n+1}, 0)$**:**
$0 \leq \beta_n + \gamma_{n+1} < \gamma_{n+1} \implies |\beta - \sum_{k=0}^{n+1} c_k \gamma_k| \leq \gamma_{n+1}$ where $c_{n+1} = 1$.

Case $\beta_n \in (-\gamma_n, -\gamma_{n+1})$**:**
$-\gamma_{n+1} < \beta_n + \gamma_{n+1} < 0 \implies |\beta - \sum_{k=0}^{n+1} c_k \gamma_k| \leq \gamma_{n+1}$ where $c_{n+1} = 1$.

Therefore as we have found a suitable c_n in all cases then we have shown that $\mathrm{H}(n) \implies \mathrm{H}(n+1)$. $\square$

With this proposition we see that our choice for γ_k is a good choice to use for the CORDIC algorithm as it covers the entire range of $(-\frac{\pi}{2}, \frac{\pi}{2})$.

Now, as stated before, the basis of our algorithm is to calculate $\begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix}$ by using rotations of a unit vector. By putting our values for γ_k into our rotation matrix we get the following:

$$\begin{pmatrix} \cos\gamma_k & -\sin\gamma_k \\ \sin\gamma_k & \cos\gamma_k \end{pmatrix} = \frac{1}{\sqrt{1+2^{-2k}}} \begin{pmatrix} 1 & -2^{-k} \\ 2^{-k} & 1 \end{pmatrix}$$

7 Preliminary References

http://math.exeter.edu/rparris/peanut/cordic.pdf
Inside your Calculator by Gerald R Rising
Wolfram Alpha