An Implementation of Sin and Cos Using Gal's Accurate Tables

Pascal Leroy (phl)

2025-02-02

This document describes the implementation of functions `Sin` and `Cos` in Principia. The goals of that implementation are to be portable (including to machines that do not have a fused multiply-add instruction), achieve good performance, and ensure correct rounding.

Overview

The implementation follows the ideas described by [GB91] and uses accurate tables produced by the method presented in [SZ05]. It guarantees correct rounding with a high probability. In circumstances where it cannot guarantee correct rounding, it falls back to the (slower but correct) implementation provided by the CORE-MATH project [SZG22] [ZSG+24]. More precisely, the algorithm proceeds through the following steps:

- perform argument reduction using Cody and Waite's algorithm in double precision (see [Mul+10, p. 379]);
- if argument reduction loses too many bits (i.e., the argument is close to a multiple of $\frac{\pi}{2}$), fall back to `cr_sin` or `cr_cos`;
- otherwise, uses accurate tables and a polynomial approximation to compute `Sin` or `Cos` with extra accuracy;
- if the result has a "dangerous rounding configuration" (as defined by [GB91]), fall back to `cr_sin` or `cr_cos`;
- otherwise return the rounded result of the preceding computation.

Notation and Accuracy Model

In this document we assume a base-2 floating-point number system with M significand bits[1] similar to the IEEE formats. We define a real function $\mathfrak{m}$ and an integer function $\mathfrak{e}$ denoting the *significand* and *exponent* of a real number, respectively:

$$x = \pm\,\mathfrak{m}(x) \times 2^{\mathfrak{e}(x)} \qquad \text{with} \qquad 2^{M-1} \leq \mathfrak{m}(x) \leq 2^M - 1$$

Note that this representation is unique. Furthermore, if x is a floating-point number, $\mathfrak{m}(x)$ is an integer.

The *unit of the last place*[2] of x is defined as:

$$\mathfrak{u}(x) \coloneqq 2^{\mathfrak{e}(x)}$$

In particular, $\mathfrak{u}(1) = 2^{1-M}$ and:

$$\frac{|x|}{2^M} < \frac{|x|}{2^M - 1} \leq \mathfrak{u}(x) \leq \frac{|x|}{2^{M-1}} \tag{1}$$

We ignore the exponent bias, overflow and underflow as they play no role in this discussion.

[1] In `binary64`, $M = 53$.

[2] Intuitively, this is the ULP "above" for powers of 2.

Finally, for error analysis we use the accuracy model of [Hig02], equation (2.4): everywhere they appear, the quantities δ_i represent a roundoff factor such that $|\delta_i| < u = \mathfrak{u}\left(\frac{1}{2}\right) = 2^{-M}$ (see pages 37 and 38). We also use θ_n and γ_n with the same meaning as in [Hig02], lemma 3.1.

Approximation of $\frac{\pi}{2}$

To perform argument reduction, we need to build approximations of $\frac{\pi}{2}$ with extra accuracy and analyse the circumstances under which they may be used and the errors that they entail on the reduced argument.

Let $z \geq 0$. We start by defining the truncation function $\mathrm{Tr}(\kappa, z)$ which clears the last κ bits of the significand of z:

$$\mathrm{Tr}(\kappa, z) \coloneqq \lfloor 2^{-\kappa}\, \mathfrak{m}(z) \rfloor\, 2^{\kappa}\, \mathfrak{u}(z)$$

We have:

$$z - \mathrm{Tr}(\kappa, z) = (2^{-\kappa}\, \mathfrak{m}(z) - \lfloor 2^{-\kappa}\, \mathfrak{m}(z) \rfloor)\, 2^{\kappa}\, \mathfrak{u}(z)$$

The definition of the floor function implies that the quantity in parentheses is in $[0, 1[$ and therefore:

$$0 \leq z - \mathrm{Tr}(\kappa, z) < 2^{\kappa}\, \mathfrak{u}(z)$$

Furthermore if the bits that are being truncated start with exactly k zeros we have the stricter inequality:

$$2^{\kappa'-1}\, \mathfrak{u}(z) \leq z - \mathrm{Tr}(\kappa, z) < 2^{\kappa'}\, \mathfrak{u}(z) \quad \text{with} \quad \kappa' = \kappa - k \tag{2}$$

This leads to the following upper bound for the unit of the last place of the truncation error:

$$\mathfrak{u}(z - \mathrm{Tr}(\kappa, z)) < 2^{\kappa'-M+1}\, \mathfrak{u}(z)$$

which can be made more precise by noting that the function $\mathfrak{u}$ is always a power of 2:

$$\mathfrak{u}(z - \mathrm{Tr}(\kappa, z)) = 2^{\kappa'-M}\, \mathfrak{u}(z) \tag{3}$$

Two-Term Approximation

In this scheme we approximate $\frac{\pi}{2}$ as the sum of two floating-point numbers:

$$\frac{\pi}{2} \simeq C_1 + \delta C_1$$

which are defined as:

$$\begin{cases} C_1 & \coloneqq \mathrm{Tr}\left(\kappa_1, \dfrac{\pi}{2}\right) \\ \delta C_1 & \coloneqq \left[\!\left[\dfrac{\pi}{2} - C_1 \right]\!\right] \end{cases}$$

Equation (2) applied to the definition of C_1 yields:

$$2^{\kappa_1'-1}\, \mathfrak{u}\left(\frac{\pi}{2}\right) \leq \frac{\pi}{2} - C_1 < 2^{\kappa_1'}\, \mathfrak{u}\left(\frac{\pi}{2}\right)$$

where $\kappa_1' \leq \kappa_1$ accounts for any leading zeroes in the bits of $\frac{\pi}{2}$ that are being truncated. Accordingly equation (3) yields, for the unit of the last place:

$$\mathfrak{u}\left(\frac{\pi}{2} - C_1\right) = 2^{\kappa_1'-M}\, \mathfrak{u}\left(\frac{\pi}{2}\right)$$

Noting that the absolute error on the rounding that appears in the definition of δC_1 is bounded by $\frac{1}{2}\, \mathfrak{u}\left(\frac{\pi}{2} - C_1\right)$, we obtain the absolute error on the two-term approximation:

$$\left| \frac{\pi}{2} - C_1 - \delta C_1 \right| \leq \frac{1}{2}\, \mathfrak{u}\left(\frac{\pi}{2} - C_1\right) = 2^{\kappa_1'-M-1}\, \mathfrak{u}\left(\frac{\pi}{2}\right) \tag{4}$$

[3]

From this we derive the following upper bound for δC_1:

$$\begin{aligned} |\delta C_1| &< \frac{\pi}{2} - C_1 + \frac{1}{2}\,\mathrm{u}\Big(\frac{\pi}{2} - C_1\Big) \\ &< 2^{\kappa_1'}\,\mathrm{u}\Big(\frac{\pi}{2}\Big) + 2^{\kappa_1'-M-1}\,\mathrm{u}\Big(\frac{\pi}{2}\Big) = 2^{\kappa_1'}(1 + 2^{-M-1})\,\mathrm{u}\Big(\frac{\pi}{2}\Big) \end{aligned} \tag{5}$$

This scheme gives a representation with a significand that has effectively $2M - \kappa_1'$ bits and is such that multiplying C_1 by an integer less than or equal to 2^{κ_1} is exact.

Three-Term Approximation

In this scheme we approximate $\frac{\pi}{2}$ as the sum of three floating-point numbers:

$$\frac{\pi}{2} \simeq C_2 + C_2' + \delta C_2$$

which are defined as:

$$\begin{cases} C_2 & \coloneqq \mathrm{Tr}\Big(\kappa_2, \dfrac{\pi}{2}\Big) \\ C_2' & \coloneqq \mathrm{Tr}\Big(\kappa_2, \dfrac{\pi}{2} - C_2\Big) \\ \delta C_2 & \coloneqq \Big\llbracket \dfrac{\pi}{2} - C_2 - C_2' \Big\rrbracket \end{cases}$$

Equation (2) applied to the definition of C_2 yields:

$$2^{\kappa_2'-1}\,\mathrm{u}\Big(\frac{\pi}{2}\Big) \leq \frac{\pi}{2} - C_2 < 2^{\kappa_2'}\,\mathrm{u}\Big(\frac{\pi}{2}\Big) \tag{6}$$

where $\kappa_2' \leq \kappa_2$ accounts for any leading zeroes in the bits of $\frac{\pi}{2}$ that are being truncated. Accordingly equation (3) yields, for the unit of the last place:

$$\mathrm{u}\Big(\frac{\pi}{2} - C_2\Big) = 2^{\kappa_2'-M}\,\mathrm{u}\Big(\frac{\pi}{2}\Big)$$

Similarly, equation (2) applied to the definition of C_2' yields:

$$\begin{aligned} 2^{\kappa_2''-1}\,\mathrm{u}\Big(\frac{\pi}{2} - C_2\Big) &\leq \frac{\pi}{2} - C_2 - C_2' < 2^{\kappa_2''}\,\mathrm{u}\Big(\frac{\pi}{2} - C_2\Big) \\ 2^{\kappa_2'+\kappa_2''-M-1}\,\mathrm{u}\Big(\frac{\pi}{2}\Big) &\leq \qquad\qquad\qquad\; < 2^{\kappa_2'+\kappa_2''-M}\,\mathrm{u}\Big(\frac{\pi}{2}\Big) \end{aligned}$$

where $\kappa_2'' \leq \kappa_2$ accounts for any leading zeroes in the bits of $\frac{\pi}{2} - C_2$ that are being truncated. Note that normalization of the significand of $\frac{\pi}{2} - C_2$ effectively drops the zeroes at positions κ_2 to κ_2' and therefore the computation of C_2' applies to a significand aligned on position κ_2'.

It is straightforward to transform these inequalities using (6) to obtain bounds on C_2':

$$2^{\kappa_2'}\Big(\frac{1}{2} - 2^{\kappa_2''-M}\Big)\,\mathrm{u}\Big(\frac{\pi}{2}\Big) < C_2' < 2^{\kappa_2'}(1 - 2^{\kappa_2''-M-1})\,\mathrm{u}\Big(\frac{\pi}{2}\Big)$$

Equation (3) applied to the definition of C_2' yields, for the unit of the last place:

$$\begin{aligned} \mathrm{u}\Big(\frac{\pi}{2} - C_2 - C_2'\Big) &= 2^{\kappa_2''-M}\,\mathrm{u}\Big(\frac{\pi}{2} - C_2\Big) \\ &= 2^{\kappa_2'+\kappa_2''-2M}\,\mathrm{u}\Big(\frac{\pi}{2}\Big) \end{aligned}$$

Noting that the absolute error on the rounding that appears in the definition of δC_2 is bounded by $\frac{1}{2}\,\mathrm{u}\Big(\frac{\pi}{2} - C_2 - C_2'\Big)$, we obtain the absolute error on the three-term approximation:

$$\Big|\frac{\pi}{2} - C_2 - C_2' - \delta C_2\Big| \leq \frac{1}{2}\,\mathrm{u}\Big(\frac{\pi}{2} - C_2 - C_2'\Big) = 2^{\kappa_2'+\kappa_2''-2M-1}\,\mathrm{u}\Big(\frac{\pi}{2}\Big) \tag{7}$$

[3] With the chosen value $\kappa_1 = 8$, the two sides of this inequality turn out to be $2^{-103.217}$ and 2^{-101}. The difference comes from the fact that $\left|\frac{\pi}{2} - C_1 - \delta C_1\right|$ has three zeroes after the last bit of its mantissa, but the theoretical computation above assumes the worse, i.e., a run of ones.

and the following upper bound for δC_2:

$$|\delta C_2| < 2^{\kappa_2' + \kappa_2'' - M}(1 + 2^{-M-1})\,\mathrm{u}\left(\frac{\pi}{2}\right) \tag{8}$$

This scheme gives a representation with a significand that has effectively $3M - \kappa_2' - \kappa_2''$ bits and is such that multiplying C_2 and C_2' by an integer less than or equal to 2^{κ_2} is exact.

Argument Reduction

Given an argument x, the purpose of argument reduction is to compute a pair of floating-point numbers $(\tilde{x}, \delta\tilde{x})$ such that:

$$\begin{cases} \tilde{x} + \delta\tilde{x} \cong x \pmod{\frac{\pi}{2}} \\ \tilde{x} \text{ is approximately in } \left[-\frac{\pi}{4}, \frac{\pi}{4}\right] \\ |\delta\tilde{x}| \le \frac{1}{2}\,\mathrm{u}(\tilde{x}) \end{cases}$$

Argument Reduction for Small Angles

If $|x| < \left[\!\left[\frac{\pi}{4}\right]\!\right]$ then $\tilde{x} = x$ and $\delta\tilde{x} = 0$.

Argument Reduction Using the Two-Term Approximation

If $|x| \le 2^{\kappa_1}\left[\!\left[\frac{\pi}{2}\right]\!\right]$ we compute:

$$\begin{cases} n & = \left[\!\left[\left[\!\left[x \left[\!\left[\frac{2}{\pi}\right]\!\right] \right]\!\right] \right]\!\right] \\ y & = x - n\ C_1 \\ \delta y & = [\![n\ \delta C_1]\!] \\ (\tilde{x}, \delta\tilde{x}) & = \mathrm{TwoDifference}(y, \delta y) \end{cases}$$

The first thing to note is that $|n| \le 2^{\kappa_1}$. We have:

$$|x| \le 2^{\kappa_1}\left[\!\left[\frac{\pi}{2}\right]\!\right] = 2^{\kappa_1}\frac{\pi}{2}(1 + \delta_1)$$

and:

$$\left[\!\left[x \left[\!\left[\frac{2}{\pi}\right]\!\right] \right]\!\right] = x\frac{2}{\pi}(1 + \delta_2)(1 + \delta_3) \tag{9}$$

from which we deduce the upper bound:

$$\begin{aligned} |n| &\le \left\lceil 2^{\kappa_1}\frac{\pi}{2}(1 + \delta_1)\frac{2}{\pi}(1 + \delta_2)(1 + \delta_3) \right\rfloor \\ &\le \left\lceil 2^{\kappa_1}(1 + \gamma_3) \right\rfloor \end{aligned}$$

If $2^{\kappa_1}\gamma_3$ is small enough (less than $1/2$), the rounding cannot cause n to exceed 2^{κ_1}. In practice we choose a relatively small value for κ_1, so this condition is met.

Now if x is close to an odd multiple of $\frac{\pi}{4}$ it is possible for misrounding to happen. There are two kinds of misrounding, with different bounds.

A misrounding of the first kind occurs if, assuming $n > 0$:

$$x < \left(n - \frac{1}{2}\right)\frac{\pi}{2} \quad \text{and} \quad \left[\!\left[x \left[\!\left[\frac{2}{\pi}\right]\!\right] \right]\!\right] > n - \frac{1}{2}$$

Using equation (9) we find that this misrounding is only possible if:

$$x > \frac{\pi}{2}\left(n - \frac{1}{2}\right)\frac{1}{(1 + \delta_2)(1 + \delta_3)} \ge \frac{\pi}{2}\left(n - \frac{1}{2}\right)\frac{1}{1 + \gamma_2}$$

In which case the computation of n results in:

$$n\frac{\pi}{2} - x < \frac{\pi}{4}\left(1 + \frac{\gamma_2}{1+\gamma_2}(2n-1)\right)$$

In this case, misrounding causes the absolute value of the reduced angle to increase and it may thus exceed $\frac{\pi}{4}$ by as much as:

$$\frac{\pi}{4}\frac{\gamma_2}{1+\gamma_2}(2^{\kappa_1+1}-1) \tag{10}$$

A misrounding of the second kind occurs if, assuming $n \geq 0$:

$$x > \left(n + \frac{1}{2}\right)\frac{\pi}{2} \quad \text{and} \quad \left[\!\left[x\left[\!\left[\frac{2}{\pi}\right]\!\right]\right]\!\right] < n + \frac{1}{2}$$

A derivation similar to the one above gives the following condition for this misrounding to be possible. Using equation (9):

$$x < \frac{\pi}{2}\left(n+\frac{1}{2}\right)\frac{1}{(1+\delta_2)(1+\delta_3)} \leq \frac{\pi}{2}\left(n+\frac{1}{2}\right)(1+\gamma_2)$$

we derive the bound:

$$x - n\frac{\pi}{2} < \frac{\pi}{4}(1 + \gamma_2(2n+1))$$

In this case, misrounding causes the absolute value of the reduced angle to decrease by as much as:

$$\frac{\pi}{4}\gamma_2(2^{\kappa_1+1}+1)$$

This is however not a concern for the accurate tables as it cannot cause the reduced angle to become negative.

Using the bound on $|n|$ and the fact that C_1 has κ_1 trailing zeroes, we see that the product $n\ C_1$ is exact. The subtraction $x - n\ C_1$ is exact by Sterbenz's Lemma. Finally, the last step performs an exact addition[4] using algorithm 4 of [HLB08].

To compute the overall error on argument reduction[5], first remember that, from equation (4), we have:

$$C_1 + \delta C_1 = \frac{\pi}{2} + \zeta \quad \text{with} \quad |\zeta| \leq 2^{\kappa_1'-M-1}\,\mathfrak{u}\!\left(\frac{\pi}{2}\right)$$

The error computation proceeds as follows:

$$\begin{aligned} y - \delta y &= x - n\ C_1 - n\ \delta C_1(1+\delta_4) \\ &= x - n(C_1 + \delta C_1) - n\ \delta C_1\ \delta_4 \\ &= x - n\frac{\pi}{2} - n(\zeta + \delta C_1\ \delta_4) \end{aligned}$$

[4] The more efficient QuickTwoDifference is not usable here. First, note that $|y|$ is equal to $\mathfrak{u}(x)$ if we take x to be the successor or the predecessor of nC_1 for any n. Ignoring rounding errors we have:

$$|\delta y| \geq n\ 2^{\kappa_1'-1}\,\mathfrak{u}\!\left(\frac{\pi}{2}\right) \geq 2^{\kappa_1'+M-2}\,\mathfrak{u}\!\left(\frac{\pi}{2}\right)\mathfrak{u}(n)$$

where we used the bound given by equation (1). Now the computation of n can result in a value that is either in the same binade or in the binade below that of x. Therefore $\mathfrak{u}(n) \geq \frac{1}{2}\,\mathfrak{u}(x)$ and the above inequality becomes:

$$|\delta y| \geq 2^{\kappa_1'+M-3}\,\mathfrak{u}\!\left(\frac{\pi}{2}\right)\mathfrak{u}(x)$$

plugging $\mathfrak{u}\!\left(\frac{\pi}{2}\right) = 2^{1-M}$ we find:

$$|\delta y| \geq 2^{\kappa_1'-2}\,\mathfrak{u}(x)$$

Therefore, as long as $\kappa_1' > 2$, there exist arguments x for which $|\delta y| > |y|$.

[5] Note that this error analysis is correct even in the face of misrounding.

from which we deduce an upper bound on the absolute error of the reduction:

$$\begin{aligned}\left|y - \delta y - \left(x - n\frac{\pi}{2}\right)\right| &\leq 2^{\kappa_1} 2^{\kappa_1'} (2^{-M-1} + 2^{-M} + 2^{-2M-1})\, \mathrm{u}\!\left(\frac{\pi}{2}\right) \\ &= 2^{\kappa_1+\kappa_1'-M}\left(\frac{3}{2} + 2^{-M-1}\right)\mathrm{u}\!\left(\frac{\pi}{2}\right) \\ &< 2^{\kappa_1+\kappa_1'-M+1}\, \mathrm{u}\!\left(\frac{\pi}{2}\right)\end{aligned}$$

where we have used the upper bound for δC_1 given by equation (5).

The exact TwoDifference yields a pair such that $|\delta\tilde{x}| \leq \frac{\mathrm{u}(\tilde{x})}{2} \leq 2^{-M}|\tilde{x}|$. Furthermore, misrounding of the first kind and the above error on the reduction may combine to cause $|\tilde{x}|$ to move above $\frac{\pi}{4}$ by as much as:

$$\frac{\pi}{4}\frac{\gamma_2}{1+\gamma_2}(2^{\kappa_1+1} - 1) + 2^{\kappa_1+\kappa_1'-M+1}\, \mathrm{u}\!\left(\frac{\pi}{2}\right)$$

The accurate tables must be constructed so that the last interval covers angles misrounded in that manner[6].

In the computation of the trigonometric functions, we need $\tilde{x} + \delta\tilde{x}$ to provide enough accuracy that the final result is correctly rounded most of the time. The above error bound shows that, if $\tilde{x}$ is very small (i.e., if x is very close to a multiple of $\frac{\pi}{2}$), the two-term approximation may not provide enough correct bits. Formally, say that we want to have $M + \kappa_3$ correct bits in the mantissa of $\tilde{x} + \delta\tilde{x}$. The error must be less than $2^{-\kappa_3}$ half-units of the last place of the result:

$$2^{\kappa_1+\kappa_1'-M+1}\, \mathrm{u}\!\left(\frac{\pi}{2}\right) \leq 2^{-\kappa_3-1}\, \mathrm{u}(\tilde{x}) \leq 2^{-\kappa_3-M}|\tilde{x}| \tag{11}$$

which leads to the following condition on the reduced angle:

$$|\tilde{x}| \geq 2^{\kappa_1+\kappa_1'+\kappa_3+1}\, \mathrm{u}\!\left(\frac{\pi}{2}\right) = 2^{\kappa_1+\kappa_1'+\kappa_3-M+2}$$

The rest of the implementation assumes that $\kappa_3 = 18$ to achieve correct rounding with high probability. If we choose $\kappa_1 = 8$ we find that $\kappa_1' = 5$ (because there are three consecutive zeroes at this location in the significand of $\frac{\pi}{2}$) and the desired accuracy is obtained as long as $|\tilde{x}| \geq 2^{-20} \simeq 9.5 \times 10^{-7}$.

Argument Reduction Using the Three-Term Approximation

If $|x| \leq 2^{\kappa_2}\left\llbracket\frac{\pi}{2}\right\rrbracket$ we compute:

$$\begin{cases} n & = \left\llbracket \left\llbracket x \left\llbracket \frac{2}{\pi} \right\rrbracket \right\rrbracket \right\rrbracket \\ y & = x - n\ C_2 \\ y' & = n\ C_2' \\ \delta y & = \llbracket n\ \delta C_2 \rrbracket \\ (z, \delta z) & = \mathrm{QuickTwoSum}(y', \delta y) \\ (\tilde{x}, \delta\tilde{x}) & = \mathrm{LongSub}(y, (z, \delta z)) \end{cases}$$

The products $n\ C_2$ and $n\ C_2'$ are exact thanks to the κ_2 trailing zeroes of C_2 and C_2'. The subtraction $x - n\ C_2$ is exact by Sterbenz's Lemma. QuickTwoSum performs an exact addition using algorithm 3 of [HLB08]; it is usable in this case because clearly $|\delta y| < |y'|$. LongSub is the obvious adaptation of the algorithm LongAdd presented in section 5 of [Lin81], which implements precise (but not exact) double-precision arithmetic.

[6] In practice this is not a stringent constraint because the distance between accurate table entries is much larger than this quantity.

It is straightforward to show, like we did in the preceding section, that:

$$|n| \leq \lceil 2^{\kappa_2}(1 + \gamma_3) \rfloor$$

and therefore that $|n| \leq 2^{\kappa_2}$ as long as $2^{\kappa_2}\gamma_3 < 1/2$. Similarly, the misrounding bound (10) is applicable with κ_2 replacing κ_1.

To compute the overall error on argument reduction, first remember that, from equation (7), we have:

$$C_2 + C_2' + \delta C_2 = \frac{\pi}{2} + \zeta_1 \quad \text{with} \quad |\zeta_1| \leq 2^{\kappa_2' + \kappa_2'' - 2M - 1}\,\mathrm{u}\left(\frac{\pi}{2}\right)$$

Let ζ_2 be the relative error introduced by LongAdd. Table 1 of [Lin81] indicates that $|\zeta_2| < 2^{2-2M}$. The error computation proceeds as follows:

$$\begin{aligned} y - y' - \delta y &= (x - n\;C_2 - n\;C_2' - n\;\delta C_2(1 + \delta_4))(1 + \zeta_2) \\ &= \left(x - n\frac{\pi}{2} - n(\zeta_1 + \delta C_2\;\delta_4)\right)(1 + \zeta_2) \\ &= x - n\frac{\pi}{2} - n(\zeta_1 + \delta C_2\;\delta_4)(1 + \zeta_2) + \left(x - n\frac{\pi}{2}\right)\zeta_2 \end{aligned}$$

from which we deduce an upper bound on the absolute error of the reduction, noting that $\left|x - n\frac{\pi}{2}\right| \leq \frac{\pi}{4}(1 + \gamma_2(2^{\kappa_2+1} + 1))$ as per (10):

$$\begin{aligned} &\left|y - y' - \delta y - \left(x - n\frac{\pi}{2}\right)\right| \\ &\quad \leq 2^{\kappa_2 + \kappa_2' + \kappa_2''}(2^{-2M-1} + 2^{-2M} + 2^{-3M-1})(1 + 2^{2-2M})\,\mathrm{u}\left(\frac{\pi}{2}\right) + 2^{2-2M}\frac{\pi}{4}(1 + \gamma_2(2^{\kappa_2+1} + 1)) \\ &\quad < 2^{\kappa_2 + \kappa_2' + \kappa_2'' - 2M}\left(\frac{3}{2} + 2^{-M-1}\right)(1 + 2^{2-2M})\,\mathrm{u}\left(\frac{\pi}{2}\right) + 2^{-2M}\;\pi\left(1 + 3 \times 2^{\kappa_2}\,\mathrm{u}\left(\frac{\pi}{2}\right)\right) \\ &\quad < 2^{\kappa_2 - 2M}(2^{\kappa_2' + \kappa_2'' + 1} + 3)\,\mathrm{u}\left(\frac{\pi}{2}\right) + 2^{-2M}\;\pi \end{aligned}$$

where the second inequality uses $\gamma_2(2^{\kappa_2+1} + 1) < 3u\;2^{\kappa_2+1}$.

A sufficient condition for the reduction to guarantee κ_3 extra bits of accuracy is for this error to be less than $2^{-\kappa_3-1}\,\mathrm{u}(\tilde{x})$ which itself is less than $2^{-\kappa_3-M}|\tilde{x}|$. Therefore we want:

$$\begin{aligned} |\tilde{x}| &\geq 2^{\kappa_3 - M}\left(2^{\kappa_2}(2^{\kappa_2' + \kappa_2'' + 1} + 3)\,\mathrm{u}\left(\frac{\pi}{2}\right) + \pi\right) \\ &= 2^{\kappa_3 - M}(2^{\kappa_2 - M + 1}(2^{\kappa_2' + \kappa_2'' + 1} + 3) + \pi) \end{aligned}$$

and it is therefore sufficient to have:

$$|\tilde{x}| \geq 2^{\kappa_3 - M}(2^{\kappa_2 + \kappa_2' + \kappa_2'' - M + 2} + 4)$$

If we choose $\kappa_3 = 18$ as above, and $\kappa_2 = 18$ we find that $\kappa_2' = 14$ and $\kappa_2'' = 15$. Therefore, the desired accuracy is obtained as long as $|\tilde{x}| \geq 65 \times 2^{-39} \simeq 1.2 \times 10^{-10}$.

Fallback

If any of the conditions above is not met, we fall back on the CORE-MATH implementation.

Accurate Tables and Their Generation

Polynomial Approximations

TODO(phl): Document t
and all the tricks that we
their generation.

The *Mathematica* function `GeneralMiniMaxApproximation` produces a minimax polynomial p such that $p(q(x))$ approximates a function $f(x)$ by minimizing the quantity $\frac{f(x) - p(q(x))}{g(x)}$. By choosing $g(x)$ appropriately, we can obtain an approximation that minimizes either the absolute or relative error on the result.

Sin Near Zero

For the sin function near zero the accurate tables method is not usable because the correction term is not small compared to the tabulated value of the function (which would be zero)[7]. Instead we use a polynomial approximation that minimizes the relative error on the result. Since $\sin x$ is an even function and since its dominant term is x, we are looking for an approximation having the form:

$$\sin x \simeq x + x^3 p_{s0}(x^2)$$

over the interval $[0, \varDelta]$, where $\varDelta$ is chosen so that $\varDelta^2 \ll 1$.

We are therefore calling `GeneralMiniMaxApproximation` with:

$$\begin{cases} q(x) & := x^2 \\ f(x) & := \dfrac{\sin x - x}{x^3} \\ g(x) & := \dfrac{\sin x}{x^3} \end{cases}$$

which results in a polynomial p_{s0} which minimizes the relative error on $\sin x$; the degree of p_{s0} is choosen so that the error is less than $\mathrm{u}(\varDelta^2)$.

In practice we choose $\varDelta = 2^{-10}$, and compute a degree-1 polynomial which induces a relative error smaller than $2^{-75.538}$ (before rounding the coefficients to machine numbers).

Sin and Cos Around Table Entries

Let (x_k, s_k, c_k) be an accurate table entry. x_k is close to $2k\varDelta$ and the accurate table interval[8] containing x_k is $\mathcal{I}_k := [(2k-1)\varDelta, (2k+1)\varDelta]$ (for $k = 0$ the interval is $\mathcal{I}_0 := [0, \varDelta]$ and $x_0 = 0$). The implementation of sin and cos starts by choosing, from the argument x (which in the case of sin, is not close to zero), a k such that $x \in \mathcal{I}_k$ and computes $h = x - x_k$. We are therefore looking for approximations of the form:

$$\sin h \simeq h + h^3 p_s(h^2)$$
$$\cos h \simeq 1 + h^2 p_c(h^2)$$

which must cover the interval $[0, h_{max}]$ with:

$$h_{max} := \max_k \{x_k - (2k-1)\varDelta, (2k+1)\varDelta - x_k\}$$

The error function used in these approximations is guided by the error analysis. We will see below (see Sin and Cos) that the analysis involves an expression like:

$$\begin{aligned} h(2\ \delta\tilde{x} + h)p_c(h^2) &= h(2\ \delta\tilde{x} + h)\left(\frac{\cos h - 1}{h^2} + \mathcal{E}(h)\right) \\ &= \cos h - 1 + 2\ \delta\tilde{x}\frac{\cos h - 1}{h} + 2h\ \delta\tilde{x}\ \mathcal{E}(h) \end{aligned}$$

where $\mathcal{E}(h)$ is an error term resulting from the minimax approximation. For this expression to behave properly when $h \to 0$ it is necessary that $\mathcal{E}(h) = \mathcal{O}(1)$. This is not a trivial property: it is not verified by approximations with a relative error or an absolute error on cos. While this leaves us with many choices for the error function, the simplest one is $\mathcal{E}(h) \propto (\cos h - 1)/h^2$. Therefore, for sin we call `GeneralMiniMaxApproximation` with:

$$\begin{cases} q(h) & := h^2 \\ f(h) & := \dfrac{\sin h - h}{h^3} \\ g(h) & := \dfrac{\sin h - h}{h^3} \end{cases}$$

[7] It would be possible to have one set of tables per binade with progressively denser intervals, but that would have a terrible performance as the tables would end up being very large.

[8] Obviously the intervals $\mathcal{I}_k$ cannot all contain their bounds. Because of the way they are computed, the odd multiples of $\varDelta$ which separate the intervals are rounded to the nearest even k. Therefore, there is an alternation of open-open intervals (for k odd) and closed-closed intervals (for k even). We do not have a convenient notation for this, and anyway this detail is mostly irrelevant.

and for cos:

$$\begin{cases} q(h) & \coloneqq h^2 \\ f(h) & \coloneqq \dfrac{\cos h - 1}{h^2} \\ g(h) & \coloneqq \dfrac{\cos h - 1}{h^2} \end{cases}$$

In practice we chose above $\varDelta = 2^{-10}$ and the accurate tables construction yielded $h_{max} < \varDelta + 2^{-17.834}$. The minimax computation results in a 1-degree polynomial p_s with an error smaller than $2^{-52.689}$ and a 1-degree polynomial p_c with an error smaller than $2^{-51.466}$ (before rounding the coefficients to machine numbers).

TODO(phl): Cross-reference the accurate tables section.

Core Implementation

This section documents the core of the algorithms used to compute sin and cos. They take as input the result of argument reduction, $(\tilde{x}, \delta\tilde{x})$ and produce a pair $(y, \delta y)$ which is passed to the rounding test described in [Mul+10, p. 397] to decide whether $[\![y + \delta y]\!]$ is the correctly-rounded result (this is the case with high probability) or whether we need to fall back to the CORE-MATH implementation.

We use $[\![expr]\!]_{\dots}$ to denote evaluation where appropriate rounding happens on each literal or operation of the expression $expr$. This notation is used for error intervals computed using the function `IEEEEvaluateWithRelativeError` in file `mathematica/ieee754_floating_point_evaluation.wl`. In particular, this function takes into account that the evaluation of the polynomials p_{s0}, p_s, and p_c has two sources of errors: the rounding of the coefficients to machine numbers, and the error due to the floating-point operations; this may lead to asymmetrical error intervals.

TODO(phl): Try to tune the high-degree coefficient after rounding the low-degree one.

We assume that $\tilde{x}$ is positive ($\delta\tilde{x}$ may be positive or negative) and that the rounding direction is $roundTiesToEven$. We *do not* assume that the machine has an FMA instruction in our error analysis: even though we actually use this instruction when available (for performance) our rounding bounds are valid even in the absence of an FMA[9].

Reduced Angle

Argument reduction took the input angle x and produced a pair $(\tilde{x}, \delta\tilde{x})$ approximating the angle reduced modulo $\frac{\pi}{4}$. That approximation is correct to $M + \kappa_3$ bits. We therefore have, from equation (11):

$$x \equiv \tilde{x} + \delta\tilde{x} + \zeta_0\,\tilde{x} \pmod{\frac{\pi}{4}} \qquad |\zeta_0| \le 2^{-\kappa_3 - M}$$

For simplicity of the error analysis we can assume $0 \le x \le \frac{\pi}{4}$: even though we do not do argument reduction when x is less than $\frac{\pi}{4}$, the error analysis would work exactly the same if x was simply provided as a high-accuracy value (e.g., a double-double) and reduced using the techniques above.

Rounding Test

The rounding test described in [Mul+10, pp. 397–400] is done by comparing y and $[\![y + [\![\delta y\ e]\!]]\!]$ for equality, where $e > 1$ is computed based on the relative error bound on $y + \delta y$. In our case, where an FMA is available, we want to compute the second part as $[\![y + \delta y\ e]\!]$. We must analyse what this implies for the computation of e.

The proof of the rounding test is rather convoluted, but for our purpose the important part is the implication[10], when y is not a power of 2 or $\delta y \ge 0$:

$$y = [\![y + [\![\delta y\ e]\!]]\!] \implies [\![\delta y\ e]\!] \le \frac{\mathrm{u}(y)}{2} \implies \delta y\ e\left(1 - \mathrm{u}\left(\frac{1}{2}\right)\right) \le \frac{\mathrm{u}(y)}{2}$$

[9] This means that the accuracy computed below is possibly pessimistic by a few hundredth of bits, which has no effect on performance.

[10] We adapt the proof to use our notation.

When an FMA is used the smaller term has to be less that half a ULP of the larger term, otherwise the result would round away from y. This gives the tighter bound:

$$y = [\![y + \delta y\ e]\!] \implies \delta y\ e \leq \frac{\mathfrak{u}(y)}{2}$$

Similarly when y is a power of 2 and $\delta y < 0$ the proof depends on the implications:

$$y = \left[\!\!\left[y + [\![\delta y\ e]\!] \right]\!\!\right] \implies |[\![\delta y\ e]\!]| \leq \frac{\mathfrak{u}(y)}{4} \implies -\delta y\ e \left(1 - \mathfrak{u}\left(\frac{1}{2}\right)\right) \leq \frac{\mathfrak{u}(y)}{4}$$

When an FMA is used the smaller term, which is negative, has to be less than half a ULP *below* the larger term otherwise the result would round below y. Because the ULP below is half the ULP above, this gives the tighter bound:

$$y = [\![y + \delta y\ e]\!] \implies -\delta y\ e \leq \frac{\mathfrak{u}(y)}{4}$$

Putting these tighter bounds together we obtain a slightly smaller value for e:

$$\begin{aligned} k &:= \lfloor -\log_2 \bar{\epsilon}_1 - M \rfloor \\ e &:= 1 + \frac{1 + 2^{M+1}\ \bar{\epsilon}_1}{1 - \bar{\epsilon}_1 - 2^{-k+1}} \end{aligned}$$

In each of the error analyses below we document the factor e to use for the rounding test.

Sin Near Zero

If $|\tilde{x}| \leq \Delta = 2^{-10}$ the steps of the computation are as follows:

$$\begin{cases} t_1 & := [\![p_{s0}(\tilde{x}^2)]\!]_{\ldots} \\ t_2 & := \left[\!\!\left[[\![\tilde{x}^2]\!]\ \tilde{x} \right]\!\!\right] \\ t_3 & := \left[\!\!\left[[\![t_1 t_2]\!] + \delta\tilde{x} \right]\!\!\right] \\ y & := \tilde{x} \\ \delta y & := t_3 \end{cases}$$

The relative error on the minimax polynomial may be rewritten as:

$$p_{s0}(x^2) = \frac{(1 + \zeta_1) \sin x - x}{x^3} \qquad |\zeta_1| < 2^{-75.538}$$

for $|x| \leq \Delta$.

The errors committed at each step are as follows:

$$\begin{cases} t_1 = p_{s0}(\tilde{x}^2)(1 + \zeta_2) & \zeta_2 \in\ ]-2^{-52.415}, 2^{-53.999}[\\ \quad = \dfrac{(1 + \zeta_1) \sin \tilde{x} - x}{\tilde{x}^3}(1 + \zeta_2) & \\ t_2 = \tilde{x}^3(1 + \delta_1)(1 + \delta_2) & \\ t_3 = (t_1 t_2 (1 + \delta_3) + \delta\tilde{x})(1 + \delta_4) & \end{cases}$$

The relative error on the entire computation is:

$$r(\tilde{x}, \delta\tilde{x}) := \frac{y + \delta y}{\sin(\tilde{x} + \delta\tilde{x} + \zeta_0\ \tilde{x})} - 1$$

The function r can be computed using *Mathematica* interval arithmetic over the triangular domain $|\tilde{x}| \leq \Delta$, $|\delta\tilde{x}| \leq 2^{-M}|\tilde{x}|$. Plotting shows that this function reaches its extrema at the corners of the domain, which is logical because we expect the minimax polynomial to reach its largest errors on the bounds of the optimization interval. In practice we find that $|r(\tilde{x}, \delta\tilde{x})| < 2^{-70.517}$ (as expected, this is a bit worse than the error originating from the angle reduction). The rounding test must be done with $e =$ `0x1.00000B2D8'1D8p0`.

TODO(phl): $k = 0.19$ would gain 0.7 bits. Worthwhile?

Note that the above steps do not compute the terms $\tilde{x}^n\ \delta\tilde{x}$ for $n > 0$. The largest such term is for $n = 2$ and the relative error that it induces is:

$$-\frac{1}{2}\frac{\tilde{x}^2\ \delta\tilde{x}}{\sin(\tilde{x} + \delta\tilde{x})}$$

This function reaches its extrema on the corners of the triangular domain and is found to be smaller than $2^{-73.999}$, so its contribution to the overall relative error would be very small.

Sin and Cos Around Table Entries

If $\tilde{x} \in \mathcal{I}_k$ we first compute $h = \tilde{x} - x_k$. It is essential that this subtraction be exact, and therefore that the conditions of Sterbenz's lemma be met: we must have $\tilde{x}/2 \leq x_k \leq 2\tilde{x}$. Based on the range of $\tilde{x}$, a sufficient condition for this is:

TODO(p
exact.

$$\frac{2k+1}{2}\Delta \leq x_k \leq 2(2k-1)\Delta$$

Because x_k is close to $2k\Delta$ the left part of this condition is trivially met for all $k > 0$. However the second part is only trivially met if $k > 1$. If $k = 1$ it becomes $x_1 \leq 2\ \Delta$. In other words, when building the accurate tables we must only look for x_1 *below* $2\ \Delta$.

TODO(phl): Cross-reference in the accurate tables section.

For sin we do not use the interval $\mathcal{I}_0$ (see Sin Near Zero, above) but for cos we do. That interval is special because $x_0 = 0$: the computation of h is therefore trivially correct.

Interestingly, $x_1 < 2\Delta$ is also the necessary condition to compute $h\ c_k + s_k$ exactly using an FMA as explained in section 2.1 of [SZ05]. The subtraction in the second step of their algorithm must be exact, therefore the following Sterbenz inequalities must hold[11]:

$$\frac{s_k}{2} \leq h\ c_k + s_k \leq 2\ s_k$$

This may be rewritten as:

$$-\frac{1}{2}\tan x_k \leq h \leq \tan x_k$$

Let $x_k = 2k\Delta + \epsilon_k$ for some small ϵ_k. By the definition of h we have:

$$-\Delta - \epsilon_k \leq h \leq \Delta - \epsilon_k$$

Now we know that $x_k < \tan x_k$ and since $\Delta - \epsilon_k < 2k\Delta + \epsilon_k$ for all $k > 0$ the right side of the Sterbenz condition is always true. The left side is more interesting though as:

$$-\frac{1}{2}\tan x_k < -\frac{x_k}{2} = -k\Delta - \frac{\epsilon_k}{2}$$

For $k > 1$, this quantity is clearly smaller than $-\Delta - \epsilon_k$, the lower bound of h. However, when $k = 1$, $-\Delta - \epsilon_1/2 \leq -\Delta - \epsilon_1$ requires that $\epsilon_1 \leq 0$, in other words, that x_1 be below 2Δ. Note that we do not need to go through a similar proof for the exactness of $-h\ s_k + c_k$ because $h\ s_k$ is much smaller than c_k for all k.

Finally, for the purpose of error analysis, the relative errors on the minimax polynomials may be rewritten as:

$$p_s(h^2) = \frac{\sin h - h}{h^3}(1 + \zeta_1) \qquad |\zeta_1| < 2^{-52.689}$$

$$p_c(h^2) = \frac{\cos h - 1}{h^2}(1 + \zeta_2) \qquad |\zeta_2| < 2^{-51.466}$$

[11] Note that $\mathcal{I}_0$ is uninteresting because $c_0 = 1$ and $s_0 = 0$ so the computation is trivially exact.

Sin

The first step of the computation is to evaluate $h\ c_k + s_k$ exactly using an FMA:

$$(z, \delta z) = t_0 = h\ c_k + s_k$$

where z and δz have nonoverlapping significands. For the purpose of describing the computation and analysing errors we will write $\delta z = t_0\ \delta_0$ and $z = t_0(1 - \delta_0)$, where the two terms are exact.

The remaining steps of the computation are then as follows:

$$\begin{cases} t_1 & \coloneqq \llbracket p_s(h^2) \rrbracket_{\ldots} \\ t_2 & \coloneqq \llbracket p_c(h^2) \rrbracket_{\ldots} \\ t_3 & \coloneqq \llbracket h \llbracket (\delta\tilde{x} + \delta\tilde{x}) + h \rrbracket \rrbracket \\ t_4 & \coloneqq \llbracket \llbracket h^2 \rrbracket\ h \rrbracket \\ t_5 & \coloneqq \left\llbracket \llbracket \llbracket s_k \rrbracket\ t_3 \rrbracket\ t_2 \right\rrbracket \\ t_6 & \coloneqq \llbracket \llbracket t_4\ t_1 \rrbracket + \delta\tilde{x} \rrbracket \\ t_7 & \coloneqq \left\llbracket \llbracket \llbracket c_k \rrbracket\ t_6 \rrbracket + t_5 \right\rrbracket \\ t_8 & \coloneqq \llbracket \llbracket t_0\ \delta_0 \rrbracket + t7 \rrbracket \\ y & \coloneqq t_0(1 - \delta_0) \\ \delta y & \coloneqq t_8 \end{cases}$$

where we have made the rounding of the accurate table elements s_k and c_k explicit, and used the fact that the computation of $\delta\tilde{x} + \delta\tilde{x}$ is exact.

The errors commited at each step are as follows:

$$\begin{cases} t_1 = p_s(h^2)\big(1 + \zeta_3\big) & \zeta_3 \in \left]-2^{-54.946}, 2^{-52.200}\right[\\ \quad = \dfrac{\sin h - h}{h^3}\big(1 + \zeta_1\big)\big(1 + \zeta_3\big) & \\ t_2 = p_c(h^2)\big(1 + \zeta_4\big) & \zeta_4 \in \left]-2^{-52.855}, 2^{-53.160}\right[\\ \quad = \dfrac{\cos h - 1}{h^2}\big(1 + \zeta_2\big)\big(1 + \zeta_4\big) & \\ t_3 = h(2\ \delta\tilde{x} + h)(1 + \delta_1)(1 + \delta_2) & \\ t_4 = h^3(1 + \delta_3)(1 + \delta_4) & \\ t_5 = \llbracket s_k \rrbracket\ t_3\ t_2\ (1 + \delta_5)(1 + \delta_6) & \\ t_6 = (t_4\ t_1\ (1 + \delta_7) + \delta\tilde{x})(1 + \delta_8) & \\ t_7 = (\llbracket c_k \rrbracket\ t_6\ (1 + \delta_9) + t_5)(1 + \delta_{10}) & \\ t_8 = (t_0\ \delta_0\ (1 + \delta_{11}) + t_7)(1 + \delta_{12}) & \end{cases}$$

The relative error on the entire computation is:

$$r(\tilde{x}, \delta\tilde{x}) \coloneqq \frac{y + \delta y}{\sin\big(\tilde{x} + \delta\tilde{x} + \zeta_0\ \tilde{x}\big)} - 1$$

Great care is required when evaluating this expression using interval arithmetic because the terms in δ_0 in t_8 and y could be considered independent and lead to a useless bound. To avoid this issue δ_0 must be factored out to appear only once. Specifically the δ_0 term after factoring is:

$$(h\llbracket c_k \rrbracket + \llbracket s_k \rrbracket)((1 + \delta_{11})(1 + \delta_{12}) - 1)$$

which is pleasantly small, and even though δ_{12} also appears in the term that does not involve δ_0, the effect of replicating it is acceptable.

The function r must be evaluated separately for each interval around x_k, with $\tilde{x} \in [(2k - 1)\varDelta, (2k + 1)\varDelta]$ and $|\delta\tilde{x}| < |\tilde{x}|/2^M$. This defines a trapezoidal domain and r reaches its extrema at the corners of that domain. In practice we find using

Mathematica interval arithmetic that $|r(\tilde{x}, \delta\tilde{x})| < 2^{-68.726}$. The relative error for $k = 1$ is particularly high and is an outlier. Other than than, the worst relative errors correspond to table entries that have many ones after their accurate zeroes (or many zeroes after their accurate ones). The rounding test must be done with $e =$ `0x1.0002'6ABE'2720'1p0`.

TODO(phl): Try to understand why.

TODO(phl): Should we try accurate tables with 19 bits?

Note that the above steps do not compute the terms $h^n\ \delta\tilde{x}$ for $n > 1$. The largest such term is for $n = 2$ and the relative error that it induces is:

$$-\frac{1}{2}\frac{h^2\ \delta\tilde{x}\ \cos x_k}{\sin(\tilde{x} + \delta\tilde{x})}$$

This function reaches its extrema on the corners of the trapezoidal domain and is found to be smaller than $2^{-73.999}$ for all k, so its contribution to the overall relative error would be very small. (The term for $n = 1$, on the other hand, induces an error of the order of $2^{-63.348}$ so it must be computed.)

Cos

The first step of the computation is to evaluate $-h\ s_k + c_k$ exactly using an FMA:

$$(z, \delta z) = t_0 = -h\ s_k + c_k$$

where z and δz have nonoverlapping significands. For the purpose of describing the computation and analysing errors we will write $\delta z = t_0\ \delta_0$ and $z = t_0(1 - \delta_0)$, where the two terms are exact.

The remaining steps of the computation are then as follows:

$$\begin{cases}
t_1 & \coloneqq [\![p_s(h^2)]\!]_{\ldots} \\
t_2 & \coloneqq [\![p_c(h^2)]\!]_{\ldots} \\
t_3 & \coloneqq [\![h[\![(\delta\tilde{x} + \delta\tilde{x}) + h]\!]]\!] \\
t_4 & \coloneqq [\![[\![h^2]\!]\ h]\!] \\
t_5 & \coloneqq \left[\!\!\left[[\![[\![c_k]\!]\ t_3]\!]\ t_2\right]\!\!\right] \\
t_6 & \coloneqq [\![[\![t_4\ t_1]\!] + \delta\tilde{x}]\!] \\
t_7 & \coloneqq \left[\!\!\left[-[\![[\![s_k]\!]\ t_6]\!] + t_5\right]\!\!\right] \\
t_8 & \coloneqq [\![[\![t_0\ \delta_0]\!] + t_7]\!] \\
y & \coloneqq t_0(1 - \delta_0) \\
\delta y & \coloneqq t_8
\end{cases}$$

where we have made the rounding of the accurate table elements s_k and c_k explicit, and used the fact that the computation of $\delta\tilde{x} + \delta\tilde{x}$ is exact.

The errors commited at each step are as follows:

$$\begin{cases}
t_1 = p_s(h^2)(1 + \zeta_3) & \zeta_3 \in \left]-2^{-54.946}, 2^{-52.200}\right[\\
\quad = \dfrac{\sin h - h}{h^3}(1 + \zeta_1)(1 + \zeta_3) & \\
t_2 = p_c(h^2)(1 + \zeta_4) & \zeta_4 \in \left]-2^{-52.855}, 2^{-53.160}\right[\\
\quad = \dfrac{\cos h - 1}{h^2}(1 + \zeta_2)(1 + \zeta_4) & \\
t_3 = h(2\ \delta\tilde{x} + h)(1 + \delta_1)(1 + \delta_2) & \\
t_4 = h^3(1 + \delta_3)(1 + \delta_4) & \\
t_5 = [\![c_k]\!]\ t_3\ t_2\ (1 + \delta_5)(1 + \delta_6) & \\
t_6 = (t_4\ t_1\ (1 + \delta_7) + \delta\tilde{x})(1 + \delta_8) & \\
t_7 = ([\![s_k]\!]\ t_6\ (1 + \delta_9) + t_5)(1 + \delta_{10}) &
\end{cases}$$

The relative error on the entire computation is:

$$r(\tilde{x},\delta\tilde{x}) \coloneqq \frac{y+\delta y}{\sin\left(\tilde{x}+\delta\tilde{x}+\zeta_0\ \tilde{x}\right)}-1$$

As explained above (see Sin) the expression for r must be rewritten so that δ_0 only appears once. In this case the δ_0 term is:

$$(h\llbracket s_k\rrbracket - \llbracket c_k\rrbracket)(1-(1+\delta_{11})(1+\delta_{12}))$$

and a *Mathematica* computation yields $|r(\tilde{x},\delta\tilde{x})| < 2^{-69.217}$. Again, the largest errors come from table entries that have many ones after their accurate zeroes (or many zeroes after their accurate ones). The rounding test must be done with $e =$ `0x1.0001'B839'335E'1p0`.

Note that the above steps do not compute the terms $h^n\ \delta\tilde{x}$ for $n > 1$. The largest such term is for $n = 2$ and the relative error that it induces is:

$$\frac{1}{2}\frac{h^2\ \delta\tilde{x}\ \ \sin x_k}{\cos(\tilde{x}+\delta\tilde{x})}$$

This function reaches its extrema on the corners of the trapezoidal domain and is found to be smaller than $2^{-74.348}$ for all k, so its contribution to the overall relative error would be very small. (The term for $n = 1$, on the other hand, induces an error of the order of $2^{-63.346}$ so it must be computed.)

References

[GB91] S. Gal and B. Bachelis. "An Accurate Elementary Mathematical Library for the IEEE Floating Point Standard". In: *ACM Transactions on Mathematical Software* 17.1 (Mar. 1991), pp. 26–45.

[Hig02] N. J. Higham. *Accuracy and Stability of Numerical Algorithms.* Society for Industrial and Applied Mathematics, 2002.

[HLB08] Y. Hida, X. S. Li, and D. H. Bailey. "Library for Double-Double and Quad-Double Arithmetic". Preprint at `https://www.davidhbailey.com/dhbpapers/qd.pdf`. May 8, 2008.

[Lin81] S. Linnainmaa. "Software for Doubled-Precision Floating-Point Computations". In: *ACM Transactions on Mathematical Software* 7.3 (Sept. 1981), pp. 272–283.
DOI: `10.1145/355958.355960`.

[Mul+10] J.-M. Muller, N. Brisebarre, F. De Dinechin, C.-P. Jeannerod, V. Lefèvre, G. Melquiond, N. Revol, D. Stehlé, and S. Torres. *Handbook of Floating-Point Arithmetic.* Birkhäuser, 2010.

[SZ05] D. Stehlé and P. Zimmermann. "Gal's accurate tables method revisited". In: *17th IEEE Symposium on Computer Arithmetic (ARITH'05)* (Cape Cod, MA, USA, June 27–29, 2005). Ed. by P. Montuschi and E. Schwarz. IEEE Computer Society, June 2005, pp. 257–264.
DOI: `10.1109/ARITH.2005.24`.

[SZG22] A. Sibidanov, P. Zimmermann, and S. Glondu. "The CORE-MATH Project". In: *2022 IEEE 29th Symposium on Computer Arithmetic (ARITH).* IEEE, Sept. 2022, pp. 26–34.
DOI: `10.1109/ARITH54963.2022.00014`.
eprint: `https://inria.hal.science/hal-03721525v3/file/core-math-final.pdf`.

[ZSG+24] P. Zimmermann, A. Sibidanov, S. Glondu, et al. *The CORE-MATH Project.* Software. Apr. 2024.