Accuracy of Mathematical Functions in Single Precision

Paul Zimmermann

September 17, 2020

The IEEE 754 single-precision (`binary32`) format has $2^{32} - 2^{24} = 4278190080$ values, not counting +Inf, -Inf, and NaN values. For a function with a single input—i.e., excluding the `pow` function for example—it is possible to check all values by exhaustive search, and to compare them to the GNU MPFR library, which guarantees correct rounding. For rounding to nearest, we have checked the accuracy of all single-precision functions for six mathematical libraries: GNU libc 2.32, the Intel Math Library shipped with the Intel compiler (icc) 19.0.4.243, AMD LibM 3.5, RedHat Newlib 3.3.0, OpenLibm 0.7.0, and Musl 1.2.1.

For each function, assuming y is the value returned by the library, and z is the exact result (as with infinite precision), we denote by e the absolute difference between y and z in terms of units-in-last-place of z, and by E the absolute difference between y and $Z = \mathrm{RN}(z)$, in terms of units-in-last-place of Z. Thus e is a real, while E is an integer (except in some corner cases). Table 1 summarizes the maximal value of e for each function and each library. In detailed tables (Tables 2, 3, 4), we indicate the number of inputs with $E \geq 1$ (thus incorrectly rounded), with $E \geq 2$, and the maximum value of e. Maximal values of e are given with 3 decimal digits, rounded up; thus for example 2.01 means that the relative error is bounded by $2.01\mathrm{ulp}(z)$ for all `binary32` inputs.

Our definition of ulp (unit-in-last-place) is the following: for $2^{e-1} \leq |x| < e$, and precision p, we define $\mathrm{ulp}(x) = 2^{e-p}$. i.e., the distance between two consecutive p-bit floating-point numbers in the binade $[2^{e-1}, 2^e]$. Some other definition exist, see [1].

The results for GNU libc were obtained on a Xeon E7-4850, with GCC 8.3.0 under Debian 10. Those for the Intel Math Library were obtained on an AMD EPYC 7702, with icc version 19.0.4.243 (gcc version 9.2.0 compatibility). The results with AMD LibM, RedHat Newlib, OpenLibm and Musl were obtained on an Intel Core i5-4590. Newlib was configured with default flags (in particular, without use of hardware FMA).

We see that for all libraries, the sqrt function is correctly rounded for all binary32 inputs, as required by IEEE 754. The Intel Math Library gives in general more accurate results, except for some functions, where other libraries have a smaller maximal error.

The j0, j1, y0, and y1 functions give large errors for all libraries except the Intel Math Library.

For AMD LibM, the maximal error for exp is 1.00 since for $x = -1.032789383e+02$, it yields 0 instead of the smallest subnormal 2^{-149}, where $\exp(x)$ is slightly smaller than the smallest subnormal. The same issue arises with exp2 and $x = -1.490000153e + 02$, and with exp10 and $x = -4.485347366e + 01$. (Decimal floating-point values should be rounded to nearest to get the corresponding `binary32` value.)

Notes about RedHat Newlib: for $x = -0$, and $-2^{-128} \leq x < 0$, tgamma returns ∞ instead of $-\infty$, this case was not taken into account in the maximal error. Still for Newlib, we used the

library	GNU libc	Intel Math Library	AMD LibM	RedHat Newlib	OpenLibm	Musl
version	2.32	icc 19.0.4.243	3.5	3.3.0	0.7.0	1.2.1
acos	0.899	**0.528**	0.669	0.899	0.918	0.918
acosh	2.01	**0.501**	0.504	2.01	2.01	2.01
asin	0.898	**0.528**	0.861	0.926	0.743	0.743
asinh	1.78	0.527	**0.518**	1.78	1.78	1.78
atan	0.853	0.541	**0.501**	0.853	0.853	0.853
atanh	1.73	0.507	**0.506**	1.73	1.73	1.73
cbrt	0.969	0.520	0.548	3.56	**0.500**	**0.500**
cos	0.561	0.548	0.530	2.91	**0.501**	**0.501**
cosh	1.89	0.506	**0.500**	2.51	1.36	1.03
erf	0.968	**0.507**	0.968	0.968	0.943	0.968
erfc	3.13	**0.502**	3.13	63.9	3.17	3.13
exp	**0.502**	0.506	1.00	0.911	0.911	**0.502**
exp10	**0.502**	0.507	1.00	1.06	NA	3.88
exp2	0.502	0.519	1.00	1.02	**0.501**	0.502
expm1	0.813	0.544	**0.537**	0.813	0.813	0.813
j0	6.18e6	**0.678**	4.77e9	6.18e6	3.66e6	3.66e6
j1	2.25e6	**1.69**	7.15e8	1.68e7	2.25e6	2.25e6
lgamma	6.78	**0.510**	6.78	7.50e6	7.50e6	7.50e6
log	0.818	**0.519**	0.940	0.888	0.888	0.818
log10	2.07	**0.516**	0.626	2.10	0.832	0.832
log1p	1.30	0.525	**0.504**	1.30	0.839	0.835
log2	0.752	**0.508**	0.586	1.65	0.865	0.752
sin	0.561	0.546	0.530	1.37	**0.501**	**0.501**
sinh	1.89	0.538	**0.501**	2.51	1.83	1.83
sqrt	**0.500**	**0.500**	**0.500**	**0.500**	**0.500**	**0.500**
tan	1.48	0.520	**0.509**	3.48	0.800	0.800
tanh	2.19	**0.514**	1.27	2.19	2.19	2.19
tgamma	7.91	0.510	7.91	239.	**0.501**	**0.501**
y0	4.86e6	**3.40**	1.52e10	4.84e6	4.84e6	4.84e6
y1	6.18e6	**2.07**	4.65e8	6.18e6	4.17e6	3.66e6

Table 1: Maximal value of e.

	GNU libc 2.32			icc 19.0.4.243		
function	$E \geq 1$	$E \geq 2$	max e	$E \geq 1$	$E \geq 2$	max e
acos	5422146	0	0.899	65950	0	0.528
acosh	243413455	2698	2.01	283	0	0.501
asin	4581700	0	0.898	469988	0	0.528
asinh	619608176	2748	1.78	963558	0	0.527
atan	21089464	0	0.853	505178	0	0.541
atanh	52062348	5790	1.73	240154	0	0.507
cbrt	453492162	0	0.969	15279372	0	0.520
cos	28209642	0	0.561	15732888	0	0.548
cosh	17868534	3558	1.89	139708	0	0.506
erf	126805016	0	0.968	908674	0	0.507
erfc	20494449	302363	3.13	1761	0	0.502
exp	170648	0	0.502	250299	0	0.506
exp10	169838	0	0.502	386017	0	0.507
exp2	168362	0	0.502	717434	0	0.519
expm1	12920601	0	0.813	539655	0	0.544
j0	1334176546	269351612	6.18e6	11960	0	0.678
j1	1340594104	274741908	2.25e6	11628	2	1.69
lgamma	500354453	8246657	6.78	100287	0	0.510
log	416908	0	0.818	1045	0	0.519
log10	29787060	62225	2.07	151074	0	0.516
log1p	11534111	0	1.30	255701	0	0.525
log2	313550	0	0.752	276	0	0.508
sin	29362812	0	0.561	12374252	0	0.546
sinh	71328448	34776	1.89	247226	0	0.538
sqrt	**0**	**0**	**0.500**	**0**	**0**	**0.500**
tan	83411250	0	1.48	694770	0	0.520
tanh	118674314	729782	2.19	164068	0	0.514
tgamma	209259574	20924067	7.91	3971282	0	0.510
y0	1304302535	187031173	4.86e6	6785	1	3.40
y1	1199498354	146032505	6.18e6	10384	1	2.07

Table 2: GNU libc and Intel Math Library.

lgammaf_r function, since we were unable to compile the lgammaf function (with lgamma Newlib says undefined reference to '_impure_ptr').

The notation NA means "Not Available" (exp10 is not available in OpenLibm).

Acknowledgements. The author thanks Claude-Pierre Jeannerod and Vincent Lefèvre who helped to improve that note, and Alexei Sibidanov who managed to compile Newlib 3.3.0.

References

[1] Muller, J.-M. On the definition of ulp(x). Research Report RR-5504, LIP RR-2005-09, INRIA, LIP, Feb. 2005.

	AMD LibM 3.5			RedHat Newlib 3.3.0		
function	$E \geq 1$	$E \geq 2$	max e	$E \geq 1$	$E \geq 2$	max e
acos	707885	0	0.669	5422146	0	0.899
acosh	21852	0	0.504	244658623	2698	2.01
asin	2454466	0	0.861	2358230	0	0.926
asinh	185582	0	0.518	542122908	2748	1.78
atan	3534	0	0.501	6406812	0	0.853
atanh	50260	0	0.506	52062348	5790	1.73
cbrt	10626352	0	0.548	1799139486	116334632	3.56
cos	2876076	0	0.530	209833072	6	2.91
cosh	**0**	**0**	**0.500**	23905668	7706	2.51
erf	126805016	0	0.968	126741900	0	0.968
erfc	20494449	302363	3.13	21247299	1131209	63.9
exp	114132	0	1.00	17982847	0	0.911
exp10	102461	0	1.00	18423203	0	1.06
exp2	86902	0	1.00	18401203	0	1.02
expm1	102330	0	0.537	12920601	0	0.813
j0	1353797232	310998452	4.77e9	1338235574	279528826	6.18e6
j1	1369557306	337817680	7.15e8	1818091384	1376362116	1.68e7
lgamma	500354453	8246657	6.78	510903809	13277834	7.50e6
log	72371093	0	0.940	13363494	0	0.888
log10	2418509	0	0.626	30061115	91958	2.10
log1p	73898	0	0.504	11534111	0	1.30
log2	179825	0	0.586	602745869	258	1.65
sin	2866930	0	0.530	206155238	0	1.37
sinh	2	0	0.501	74587762	38924	2.51
sqrt	**0**	**0**	**0.500**	**0**	**0**	**0.500**
tan	529444	0	0.509	83455936	32	3.48
tanh	4314486	0	1.27	118674314	729782	2.19
tgamma	209259574	20924067	7.91	2028164923	1833526367	239.
y0	1314115311	207848357	1.52e10	1306144386	191859954	4.84e6
y1	1213975420	177569092	4.65e8	1201178797	153321647	6.18e6

Table 3: AMD LibM and RedHat Newlib.

	OpenLibm 0.7.0			Musl 1.2.1		
function	$E \geq 1$	$E \geq 2$	max e	$E \geq 1$	$E \geq 2$	max e
acos	5717768	0	0.918	1700216587		0.918
acosh	244658828	2698	2.01	319260148	23345165	2.01
asin	4220748	0	0.743	4220748	0	0.743
asinh	542176908	2748	1.78	642880516	2730	1.78
atan	6483278	0	0.853	1717759310	0	0.853
atanh	52089660	5790	1.73	52062556	5740	1.73
cbrt	**0**	**0**	**0.500**	**0**	**0**	**0.500**
cos	647594	0	0.501	647594	0	0.501
cosh	23865830	0	1.36	16675588	0	1.03
erf	126619324	0	0.943	127569522	0	0.968
erfc	24416748	343931	3.17	19695704	302363	3.13
exp	19194854	0	0.911	170646	0	0.502
exp10	NA	NA	NA	41421106	3446689	3.88
exp2	102250	0	0.501	168362	0	0.502
expm1	12920593	0	0.813	12920592	0	0.813
j0	1332944944	268168176	3.66e6	1422932510	271739106	3.66e6
j1	1339381958	273573380	2.25e6	1320601392	117301706	2.25e6
lgamma	508702215	10980627	7.50e6	504159259	10758067	7.50e6
log	13361747	0	0.888	416908	0	0.818
log10	12305116	0	0.832	12305116	0	0.832
log1p	11588705	0	0.839	11678873	0	0.835
log2	11476491	0	0.865	313550	0	0.752
sin	625106	0	0.501	625106	0	0.501
sinh	72347778	31216	1.83	72812234	31516	1.83
sqrt	**0**	**0**	**0.500**	**0**	**0**	**0.500**
tan	303818252	0	0.800	303818252	0	0.800
tanh	70733480	729768	2.19	112377586	290564	2.19
tgamma	2	0	0.501	3	0	0.501
y0	1303826513	186525437	4.84e6	1309884649	190741595	4.84e6
y1	1198288693	144090005	4.17e6	1171308902	67527225	3.66e6

Table 4: OpenLibm and Musl.