VARIOUS PROPOSED LISTS OF ELEMENTARY FUNCTIONS

JOHN PRYCE

1. Purpose of this document

As a basis for discussion of P1788 Motion 10, **I offer** a table that lists the elementary functions required or proposed in some actual or draft standards. The four basic arithmetic operations are not listed. At present it shows:

(1) The functions in Jürgen's P1788 Motion 10. For this list only, I have distinguished between Y="required" and y="recomended" functions.
(2) The functions in IEEE 754-2008. These are mainly from §9.2, with some from §5.4.
(3) The functions proposed in the Brönnimann–Melquiond–Pion C++ interval standard proposal n2137 of November 2006.

Anyone who wishes to add another column is welcome to do so.

I suggest, hereby, a (friendly) amendment to Jürgen's motion, whereby each function on this list, and maybe some others, is/are voted for individually. So the "voting slip" contains a check-box opposite each function. Each voter puts Yes or No into each check-box, or abstains. (Should abstention count as No, or be ignored? I leave that to George Corliss.) Those that don't reach a certain level of support are dropped. Those that remain are put on a final list. At this point we re-check carefully for consistency, and choose the name each function will have in the P1788 text. This final list is voted for as a single motion.

Whether the accuracy mode (e.g. Vienna's tight, accurate or valid) can be chosen as part of the above, I do not feel competent to judge. I suspect we need a separate motion to decide if we like the Vienna scheme or something else. And that then some experts on elementary functions should propose a motion saying what accuracy is required of each function in the P1788 list.

2. Comments

Functions appropriate for a point-arithmetic standard may not all be so for an interval standard, and we should view the 754 list in that light.

Many functions are in all three lists and I doubt if we shall have much controversy about them—but what do I know? For those that don't appear in all, here's our chance to weigh priorities.

I don't have strong feelings, except to **remember K.I.S.S.** However:

- I feel sceptical about Jürgen's `cos2, exp2, sin2`; anyway, why is `sin2` defined as $\dfrac{\sin(x)-x}{x^2}$ and not $\dfrac{\sin(x)-x}{x^3}$?
- If we have a function for $\dfrac{e^x-1}{x}$, surely there is no need of a separate one for e^x-1 ?
- How important are versions of the Gamma function and the Error function, for P1788? Will someone champion them?

Date: October 27, 2009.

- Personally I like 754's `cosPi` and related functions. E.g. one can use interval `atan2` to find the angle subtended at the origin by a box. If the true result is, say, [something, $\pi/2$] then ordinary `atan2` must increase the $\pi/2$ slightly to get a valid enclosure; whereas `atan2Pi` can report the upper bound exactly. For an algorithm that must check, say, that a box does *not* extend into the 2nd quadrant, this is a valuable simplification. (Some of my research has involved computing winding-numbers, for which this is relevant.)

Remember this motion is not concerned with "accuracy mode". Nor with the actual names we give to functions in the standard document.

I will be most grateful if people check my table for correctness, against the source documents. [List (1): Jürgen? List (2): Dan? List (3): Guillaume?]

3. The table

Function	(1)	(2)	(3)	Notes
`abs(interval X)`	y	y	y	
`acos(interval X)`	Y	y	y	
`acosh(interval X)`	Y	y	y	
`asin(interval X)`	Y	y	y	
`asinh(interval X)`	Y	y	y	
`atan(interval X)`	Y	y	y	
`atanPi(interval X)`		y		$\arctan(x)/\pi$
`atanh(interval X)`	Y	y	y	
`atan2(interval Y,interval X)`	Y	y	y	
`atan2Pi(interval Y,interval X)`		y		$\text{atan2}(y,x)/\pi$
`cbrt(interval X)`			y	cube root
`ceil(interval X)`	y			least integer $\geq x$
`compound(interval X, int y)`	y	y	y	$(1+x)^y$
`cos(interval X)`	Y	y	y	
`cos2(interval X)`	y			$(\cos(x)-1)/x^2$
`cosPi(interval X)`		y		$\cos(\pi x)$
`cosh(interval X)`	Y	y	y	
`cosh2(interval X)`	y			$(\cosh(x)-1)/x^2$
`erf(interval X)`			y	error function
`erfc(interval X)`			y	complementary error function
`exp(interval X)`	Y	y	y	
`exp1(interval X)`	y			$(e^x-1)/x$
`exp2(interval X)`		y	y	2^x
`exp2(interval X)`	y			$(e^x-1-x)/x^2$; **note name-clash**
`exp10(interval X)`		y		10^x
`expm1(interval X)`	y	y	y	e^x-1
`exp2m1(interval X)`		y		2^x-1
`exp10m1(interval X)`		y		10^x-1
`fdim(interval X,interval Y)`			y	$\max(x-y,0)$
`floor(interval X)`	y			greatest integer $\leq x$
`fma(interval X, interval Y, interval Z)`		y	y	fused multiply-add $xy+z$

Table 1: Elementary functions required or proposed in some actual or draft standards. In column (1) a distinction is made between Y=required and y=recommended.
(1) P1788 motion 10 (Vienna proposal is very similar).
(2) IEEE Std 754-2008 (mainly from clause 9).
(3) Draft C++ interval standard n2137.
Notes column describes underlying point function, when this is not obvious.

Function	(1)	(2)	(3)	Notes
`gamma(interval X)`			y	called tgamma in (3): this is also its C name
`hypot(interval X,interval Y)`	y	y	y	$\sqrt{x^2+y^2}$
`ldexp(interval X,int y)`		?	y	$x \times 2^y$
`lgamma(interval X)`			y	$\log\lvert\Gamma(x)\rvert$
`log(interval X)`	Y	y	y	natural log
`log2(interval X)`	Y	y	y	$\log_2(x)$
`log10(interval X)`	Y	y	y	$\log_{10}(x)$
`logp1(interval X)`	y	y	y	$\log(1+x)$; called log1p in (3)
`log2p1(interval X)`		y		$\log_2(1+x)$
`log10p1(interval X)`		y		$\log_{10}(1+x)$
`max(interval X,interval Y)`		y	y	
`min(interval X,interval Y)`		y	y	
`pown(interval X,int y)`	Y	y	y	x^y for integer y
`pow(interval X,interval Y)`	Y	y	y	x^y for real y
`powr(interval X,int y,int z)`	y			$x^{y/z}$ for integer y, z
`root(interval X,int y)`	y	y	y	$\sqrt[y]{x}$; called rootn in (2), nth_root in (3)
`rSqrt(interval X)`	y	y		$1/\sqrt{x}$
`sign3(interval X)`	y			sign, with values $-1, 0, 1$
`sin(interval X)`	Y	y	y	
`sin2(interval X)`	y			$(\sin(x)-x)/x^2$
`sinPi(interval X)`		y		$\sin(\pi x)$
`sinh(interval X)`	Y	y	y	
`sinh2(interval X)`	y			$(\sinh(x)-x)/x^2$
`sqrt(interval X)`	Y	y	y	
`sqr(interval X)`	Y		y	
`tan(interval X)`	Y	y	y	
`tanh(interval X)`	Y	y	y	

Table 1: Elementary functions required or proposed in some actual or draft standards. In column (1) a distinction is made between Y=required and y=recommended.
(1) P1788 motion 10 (Vienna proposal is very similar).
(2) IEEE Std 754-2008 (mainly from clause 9).
(3) Draft C++ interval standard n2137.
Notes column describes underlying point function, when this is not obvious.