Proposal for C2x
WG14 N2355

Title:	TS 18661-4 mathematical functions
Author, affiliation:	C FP group
Date:	2019-03-06
Proposal category:	New features
Target audience:	Science, engineering, finance, mathematics

Abstract: This proposal incorporates the ISO/IEC TS 18661-4 mathematical functions for <math.h> and <tgmath.h> into C2x. These functions complete the C support for the mathematical operations recommended in the IEC 60559:2008 floating-point standard, updated in IEEE 754-2019. This proposal does not include the reduction functions in TS 18661-4.

In IEC 60559 mathematical operations are recommended, rather than required, because the floating-point standard allows for small specialized implementations that don't necessarily implement a language standard. See the IEEE 754 committee's background document
http://754r.ucbtest.org/background/conformance-and-options.txt

IEC 60559 specifies the mathematical operations to be correctly rounded. TS 18661-4 does not require correct rounding, but does reserve names (with cr prefix) for correctly rounded versions.

As shown in the table below, C already supports 22 of the 39 IEC 60559 mathematical operations. The remaining ones are proposed here.

These complete the set of exponential and logarithmic functions already in C, for bases e, 2, and 10:

exp2m1	exp10	exp10m1
logp1 (= log1p)	log2p1	log10p1

These are variations on the C pow function, that allow for better performance in common applications (see the 754 committee's background document http://754r.ucbtest.org/background/power.txt):

rsqrt	an alternative to 1/sqrt, allowing better performance and a single rounding error -– a common primitive in graphics
compound	basic function for finance and growth/decay applications, more accurate than (1+x)^n
rootn	primitive n'th root
pown	power function for integer exponents
powr	models continuous power function

These are pi (half revolution) trig functions, which avoid roundoff error at multiples of pi and which allow faster argument reduction:

sinpi	cospi	tanpi	
asinpi	acospi	atanpi	atan2pi

IEC 60559 math operation	Current C function	Proposed C function
exp	**exp**	
expm1	**expm1**	
exp2	**exp2**	
exp2m1		**exp2m1**
exp10		**exp10**
exp10m1		**exp10m1**
log	**log**	
log2	**log2**	
log10	**log10**	
logp1	**log1p**	**logp1**
log2p1		**log2p1**
log10p1		**log10p1**
hypot	**hypot**	
rSqrt		**rsqrt**
compound		**compoundn**
rootn		**rootn**
pown		**pown**
pow	**pow**	
powr		**powr**
sin	**sin**	
cos	**cos**	
tan	**tan**	
sinPi		**sinpi**
cosPi		**cospi**
tanPi		**tanpi**
asinPi		**asinpi**
acosPi		**acospi**
atanPi		**atanpi**
atan2Pi		**atan2pi**
asin	**asin**	
acos	**acos**	
atan	**atan**	
atan2	**atan2**	
sinh	**sinh**	
cosh	**cosh**	
tanh	**tanh**	
asinh	**asinh**	
acosh	**acosh**	
atanh	**atanh**	

Prior art: Implementations include:

HP: exp10, rsqrt, compound

GCC, Microsoft: exp10

Intel: exp10, expm1, log1p, powr, pow, acospi, asinpi, acospi, cospi, sinpi, tanpi, atan2pi, compound

Microsoft, Khronos: trig functions based on units of pi

Microsoft: rsqrt

Sun Solaris: all the *pi and {log,exp}*{2,10}* functions