

Algorithmic C (AC) Math Release Notes

Software Version v3.1.2

November 2018

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Table of Contents

Release 3.1.2.....2
 Improved Support for AC Float.....2
 Cleanup Sine/Cosine CORDIC.....2
 Corrected Issues.....2

Release 3.1.0.....3
 Hyperbolic Tangent File Renamed.....3
 Corrected Issues.....3

Release 2.0.10.....4
 Basic Math Functions.....4
 AC Matrix Class.....5
 Linear Algebra Functions.....5

Supported Compilers.....7

Release 3.1.2

The following topics describes the changes that were made to the *ac_math* library since the last release. This release provides new functionality and bug fixes. This version of *ac_math* was included in Catapult release 10.3a.

Improved Support for AC Float

The following functions were updated to provide improved support for AC Float datatype:

- `ac_inverse_sqrt_pwl()`
- `ac_reciprocal_pwl()`
- `ac_sqrt_pwl()`

Cleanup Sine/Cosine Cordic

The `ac_sincos_cordic.h` file was updated to rename a typedef so as to avoid a redeclaration conflict.

Corrected Issues

The following user-reported problems were fixed in this release:

- **(no bug #):** File: `ac_matrix.h` – `Transpose()` member function was incorrect.
- **(no bug #):** File `ac_sqrt_pwl.h` – Fixed bug in output near normalized 1 by adding extra bit to account for upward shifting of segments against the direction of concavity.

Release 3.1.0

The following topics describes the changes that were made to the *ac_math* library since the last release. This release provides new functionality and bug fixes. This version of *ac_math* was included in Catapult release 10.3.

Hyperbolic Tangent File Renamed

The file `ac_hyperbolic_tan_pwl.h` was renamed to `ac_tanh_pwl.h` for consistency with the other header files.

Corrected Issues

The following user-reported problems were fixed in this release:

- **(no bug #):** File: `ac_aqrt.h` – Fixed expression that could cause overflow.

Release 2.0.10

This is the first official open-source release of the *ac_math* library. The following table lists the functions available in this release. For details about how to use the *ac_math* library consult the AC Math Reference Manual. This version of *ac_math* was included in Catapult release 10.2d.

Basic Math Functions

Function Type	Function Call	Approximation Method	Supported Data Types		
			ac_fixed	ac_float	ac_complex
Absolute Value	<i>ac_abs()</i>	N/A	Yes	Yes	No
Division	<i>ac_div()</i>	N/A	Yes	Yes	Yes
Normalization	<i>ac_normalize()</i>	N/A	Yes	No	Yes
Reciprocal	<i>ac_reciprocal_pwl()</i>	PWL	Yes	Yes	Yes
Logarithm Base e	<i>ac_log_pwl()</i>	PWL	Yes	No	No
	<i>ac_log_cordic()</i>	CORDIC	Yes	No	No
Logarithm Base 2	<i>ac_log2_pwl()</i>	PWL	Yes	No	No
	<i>ac_log2_cordic()</i>	CORDIC	Yes	No	No
Exponent Base e	<i>ac_exp_pwl()</i>	PWL	Yes	No	No
	<i>ac_exp_cordic()</i>	CORDIC	Yes	No	No
Exponent Base 2	<i>ac_pow2_pwl()</i>	PWL	Yes	No	No
	<i>ac_exp2_cordic()</i>	CORDIC	Yes	No	No
Generic Exponent	<i>ac_pow_pwl()</i>	PWL	Yes	No	No
	<i>ac_pow_cordic()</i>	CORDIC	Yes	No	No
Square Root	<i>ac_sqrt_pwl()</i>	PWL	Yes	Yes	Yes
	<i>ac_sqrt()</i>	N/A	Yes	No	No
Inverse Square Root	<i>ac_inverse_sqrt_pwl()</i>	PWL	Yes	Yes	Yes
Sine/Cosine	<i>ac_sincos()</i>	LUT	Yes	No	No
	<i>ac_cos_cordic()</i>	CORDIC	Yes	No	No
	<i>ac_sin_cordic()</i>	CORDIC	Yes	No	No
	<i>ac_sincos_cordic()</i>	CORDIC	Yes	No	No
Tangent	<i>ac_tan_pwl()</i>	PWL	Yes	No	No
Inverse Trig	<i>ac_atan_pwl()</i>	PWL	Yes	No	No
	<i>ac_arccos_cordic()</i>	CORDIC	Yes	No	No

Function Type	Function Call	Approximation Method	Supported Data Types		
			ac_fixed	ac_float	ac_complex
	<i>ac_arcsin_cordic()</i>	CORDIC	Yes	No	No
	<i>ac_arctan_cordic()</i>	CORDIC	Yes	No	No
Shift Left/Right	<i>ac_shift_left</i>	N/A	Yes	No	Yes
	<i>ac_shift_right</i>	N/A	Yes	No	Yes
Hyperbolic Tangent	<i>ac_tanh_pwl</i>	PWL	Yes	No	No
Sigmoid	<i>ac_sigmoid_pwl</i>	PWL	Yes	No	No

AC Matrix Class

The class `ac_matrix` implements a 2-D container class with a template parameter to specify the data type of the internal storage.

The class has member functions to implement some common operations including

- Assignment: `operator=()`
- Read-Only and Read-Write Element Access: `*this(<row>,<col>)`
- Comparison: `operator!=()`, `operator==()`
- Piecewise Addition: `operator+()`, `operator+=()`
- Piecewise Subtraction: `operator-()`, `operator-=()`
- Piecewise Multiplication: `pwisemult()`
- Matrix Multiplication (nested loops): `operator*()`
- Matrix Transpose: `transpose()`
- Sum All Elements: `sum()`
- Scale All Elements: `scale(value)`
- Formatted Stream Output: `ostream &operator<<()`

When using the computational functions with AC Datatypes, the form that returns a value is designed in such a way as to determine the full precision required in the output type in order to preserve accuracy during the operation. So using `operator+` between two 10 bit `ac_fixed` matrices will return an 11 bit `ac_fixed` matrix. If you wish to prevent the bit growth and accept the truncation, you can use the compound operators `+=`, `-=`, etc. so that the target object receives the truncated values.

In addition to the built-in member functions, the `ac_math` library also includes stand-alone functions for more complicated linear algebra operations as described in the next section.

Linear Algebra Functions

The `ac_math` library includes several linear algebra functions that operate on either `ac_matrix` or plain C-style arrays. These functions, when used with AC Datatypes, are designed to give the user greater control

over the bit precision of internal variables and the return value.

- Matrix Multiplication
- Matrix Determinant
- Cholesky Decomposition
- Cholesky Inverse
- QR Decomposition

Supported Compilers

The PWL functions use default template arguments. This requires using a C++ compiler that support the C++11 or newer standard.