rexxmathlib

Thomas Richter

COLLABORATORS

	TITLE : rexxmathlib		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Thomas Richter	July 8, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

rexxmathlib

1.1 RexxMathLib Guide

RexxMathLib.Guide - First Aid about RexxMathLib © 1995 THOR-Software Guide Version 1.03 / Library Version 38.02

Table of Contents

© THOR-Software Thomas Richter Rühmkorffstraße 10A

12209 Berlin

Germany

EMail: thor@math.tu-berlin.de

The rexxmathlib.library is FREEWARE and copyrighted © 1995/1998 by Thomas Richter. No commercial use without perimission of the author. Read the licence !

There is a bug in the mathieeedoubbas.library that comes with Workbench 3.1. Read here to find out how to fix it.

1.2 The THOR-Software Licence

The THOR-Software Licence (v2, 24th June 1998)

This License applies to the computer programs known as "rexxmathlib.library" and the "RexxMathLib.guide". The "Program", below, refers to such program. The "Archive" refers to the package of distribution, as prepared by the author of the Program, Thomas Richter. Each licensee is addressed as "you".

The Program and the data in the archive are freely distributable under the restrictions stated below, but are also Copyright (c) Thomas Richter.

Distribution of the Program, the Archive and the data in the Archive by a commercial organization without written permission from the author to any third party is prohibited if any payment is made in connection with such distribution, whether directly (as in payment for a copy of the Program) or indirectly (as in payment for some service related to the Program, or payment for some product or service that includes a copy of the Program "without charge"; these are only examples, and not an exhaustive enumeration of prohibited activities).

However, the following methods of distribution involving payment shall not in and of themselves be a violation of this restriction:

(i) Posting the Program on a public access information storage and retrieval service for which a fee is received for retrieving information (such as an on-line service), provided that the fee is not content-dependent (i.e., the fee would be the same for retrieving the same volume of information consisting of random data).

(ii) Distributing the Program on a CD-ROM, provided that

1.3 Overview

This library provides transcendental functions for the ARexx - programming language. It is a complete rewrite of Willy Langevelds rexxmathlib, no original code has been used. The new release has been completely rewritten in assembly language and is therefore not only faster (approx. 10 times), but provides also a higher precision of 15.9 digits, thanks to smarter ASCII- to float conversion routines. As an extra, some more functions have been added, see the Function Index , and the check for proper function arguments is now more strictly.

To use this library in AREXX, add the following line to your ARexx-script:

call addlib('rexxmathlib.library',0,-30,0)

The rexxmathlib.library will use the system math libraries, namely the mathieeedoubbas.library and the mathieeedoubtrans.library and will therefore work fine, regardless of a math-coprocessor.

There is a bug in the V38 mathieeedoubbas.library float-compare routine resulting in a wrong ordering of negative numbers of small absolute value.

However, I included the necessary stuff to fix this bug. A patch for this problem is provided.

1.4 How to fix the mathieeedoubbase-compare bug.

I advise you to fix the bug in the mathieeedoubbas library version 38.2, that comes with Workbench 3.1. For copyright reasons, I can not provide the patched library, but a patch file and a patch program. To apply the patch:

1) Copy the file LIBS:mathieeedoubbas.library to RAM: 2) Copy the file mathieeedoubbas.pch, which comes with this archive, to RAM: 3) Copy the "spatch" program, which is also included in this archive, to RAM: 4) Change the directory to ram: with "cd RAM:" 5) Apply the patch with "spatch mathieeedoubbas.library" 6) Copy the file RAM:mathieeedoubbas.new to LIBS:mathieeedoubbas.library. It contains the fixed library.

If any problems arise, make sure you use the original and unmodified version of the library!

1.5 Function Index

RexxMathLib.library - Function Index

ABS ACOS ACOSH ASIN ASINH ATAN ATAN2 ATANH CEIL COS COSEC COSH COT COTAN CSC DEG E EPSM EPSP EXP FABS FACT FLOOR FRACT INT LN LOG LOG10 NINT PI POL POW POWER RAD ROOT SEC SIN SINH SQR SQRT TAN TANH XTOY

1.6 ABS,FABS

NAME ABS(x),FABS(x)

calculate absolute value of the argument

ARGUMENT REQUIREMENTS none

BUGS -ABS is never called by AREXX cause it is provided as a AREXX build-in function. However, you SHOULD use FABS if you need the absolute value because it provides a higher precision than the build-in ABS.

SEE ALSO

1.7 ACOS

NAME ACOS(x)

calculate the inverse cosine of the argument (in radiants)

ARGUMENT REQUIREMENTS -1.0 <= x <= 1.0

BUGS

SEE ALSO COS SIN TAN ASIN ATAN RAD

1.8 ACOSH

NAME ACOSH(x)

calculate the inverse hyperbolic cosine of the argument

ARGUMENT REQUIREMENTS x >= 1.0

BUGS This function is implemented using the identity ACOSH(x)=LN(x+SQRT(x^2-1)) and might cause an overflow if the argument of the logarithm overflows or x^2 is out of range. An additional problem of this implementation is possibly it's non-optimal precision.

SEE ALSO COSH SINH TANH ASINH ATANH RAD

1.9 ASIN

NAME ASIN(x)

calculate the inverse sine of the argument (in radiants)

ARGUMENT REQUIREMENTS -1.0 <= x <= 1.0

BUGS

SEE ALSO COS SIN TAN ACOS ATAN RAD

1.10 ASINH

NAME ASINH(x)

calculate the inverse hyperbolic sine of the argument

ARGUMENT REQUIREMENTS none

BUGS This function is implemented using the identity ASINH(x)=LN(x+SQRT(x^2+1)) and might cause an overflow if the argument of the logarithm overflows or x^2 is out of range. The precision of this implementation might be less than optimal.

SEE ALSO COSH SINH TANH ACOSH ATANH

1.11 ATAN

NAME ATAN(x)

calculate the inverse tangent of the argument (in radiants)

ARGUMENT REQUIREMENTS none

BUGS Due to round-off errors, the inverse tangent of PI/2 is NOT infinity.

SEE ALSO COS SIN TAN ACOS ASIN RAD

1.12 ATAN2,POL

NAME ATAN2(y,x),POL(x,y)

calculate the angle between the point (x|y) and the origin (in radiants). NOTE THE DIFFERENT ARGUMENT ORDERING OF ATAN2 AND POL ! This function is also known as the argument-function of the complex number z=x+iy. For *many* values of x and y this argument is identical to ATAN(x/y), but the library TRIES to provide a higher precision if possible.

ARGUMENT REQUIREMENTS x<>0 | y<>0 x and y must not be zero at the same time, however x=0 OR y=0 is allowed.

BUGS

SEE ALSO ATAN RAD

1.13 ATANH

NAME ATANH(x)

calculate the inverse hyperbolic tangent of the argument

ARGUMENT REQUIREMENTS -1.0 < x < 1.0

BUGS Starting with v38.02, this function uses either the FPU if it is available, or uses the identity ASINH(x)=LN((1+x)/(1-x))/2 and might cause an overflow if the argument of the logarithm overflows. The precision might therefore be less than optimal.

SEE ALSO COSH SINH TANH ACOSH ASINH

1.14 CEIL

NAME CEIL(x)

calculate the lowest integer higher than x

ARGUMENT REQUIREMENTS none

BUGS Not a bug, but you should note that this function results for negative values of x in a number of lower absolute value. Hence

CEIL(2.5)=3

but

CEIL(-2.5)=-2

However, this is the CORRECT mathematical implementation of CEIL !

SEE ALSO FLOOR INT FRACT NINT

1.15 COS

NAME COS(x)

calculate the cosine of the argument (in radiants)

ARGUMENT REQUIREMENTS none

BUGS Due to round-off errors, the cosine of x is more or less random if the absolute value of x is "too large".

SEE ALSO SIN TAN ACOS ASIN ATAN DEG

1.16 COSEC,CSC

NAME COSEC(x),CSC(x)

calculate the cosecans of the argument (in radiants)

ARGUMENT REQUIREMENTS x<>0

BUGS Due to round-off errors, the cosecans of integer multiples of PI is not infinity; additionally, the result is more or less random if the absolute value of x is "too large".

SEE ALSO SEC COT COTAN

1.17 COSH

NAME COSH(x)

calculate the hyperbolic cosine of the argument

ARGUMENT REQUIREMENTS -700 < x < 700 (approx.)

BUGS

SEE ALSO SINH TANH ACOSH ASINH ATANH

1.18 COT

NAME COT(x),COTAN(x)

calculate the hyperbolic cotangent of the argument (in radiants)

ARGUMENT REQUIREMENTS x<>0

BUGS Due to round-off errors, the cotangent of PI/2 is not precisely zero.

SEE ALSO TAN SEC COSEC CSC DEG

1.19 DEG

NAME DEG(x)

convert degrees in radiants

ARGUMENT REQUIREMENTS none

BUGS Due to round-off errors, the precision might be less than optimal, but is still better than what could be expected if converted manually by ARexx.

SEE ALSO SIN COS TAN RAD

1.20 E

NAME E(x)

returns the value of E, the base of the natural logarithm. The argument is not used.

ARGUMENT REQUIREMENTS none

BUGS The result has a precision of 17 digits, although the rexxmathlib provides only a precision of 15.9 digits (and AREXX of 14 digits).

SEE ALSO PI

1.21 EPSM

NAME EPSM(x)

returns the highest floating point number lower than and distinguishable from x

ARGUMENT REQUIREMENTS none

BUGS The result is only useful as input for mathrexxlib because AREXX itself provides only a precision of 14 digits. Hence, EPSM(x)=x as far as AREXX is concerned.

SEE ALSO EPSP

1.22 EPSP

NAME EPSP(x)

returns the lowest floating point number higher than and distinguishable from x

ARGUMENT REQUIREMENTS none

BUGS The result is only useful as input for mathrexxlib because AREXX itself provides only a precision of 14 digits. Hence, EPSP(x)=x as far as AREXX is concerned.

SEE ALSO EPSM

1.23 EXP

NAME EXP(x)

calculate the exponential of the argument

ARGUMENT REQUIREMENTS x < 700 (approx.)

BUGS

SEE ALSO E LOG LN

1.24 FACT

NAME FACT(x)

calculate the factorial of x

ARGUMENT REQUIREMENTS x>=0 & x<=87 & x integer

BUGS For x lower or equal than 12, the result is calculated in integers, for higher x floating point numbers are used, hence the result might be non-integer for large x. This call should really evaluate the Gamma-function for non-integer x, but this is a non-trivial task!

SEE ALSO

1.25 FLOOR,INT

NAME FLOOR(x),INT(x)

calculate the highest integer lower than x

ARGUMENT REQUIREMENTS none

BUGS Not a bug, but you should note that this function results for negative values of x in a number of higher absolute value. Hence

INT(2.5)=2

but

INT(-2.5)=-3

However, this is the CORRECT mathematical implementation of INT !

SEE ALSO CEIL FRACT NINT

1.26 FRACT

NAME FRACT(x)

calculate the fractional part of x

ARGUMENT REQUIREMENTS none

BUGS Not a bug, but you should note that this function results for negative values of x in a positive number, too, because it is implemented as x-FLOOR(x).

Hence

FRACT(2.4)=0.4

but

FRACT(-2.4)=0.6

However, this is the CORRECT mathematical implementation of FRACT !

SEE ALSO CEIL FRACT NINT

1.27 LN,LOG

NAME LN(x),LOG(x)

calculate the natural logarithm of x

ARGUMENT REQUIREMENTS x>0

BUGS

SEE ALSO E EXP LOG10

1.28 LOG10

NAME LOG10(x)

calculate the decadic logarithm of x

ARGUMENT REQUIREMENTS x>0

BUGS

SEE ALSO LOG LN

1.29 NINT

NAME NINT(x)

calculate the nearest integer to x

ARGUMENT REQUIREMENTS none

BUGS Not a bug, but you should note that this function results for negative values of x with a fractional part of 0.5 in a different integer than for positive x. Hence

NINT(2.5)=3

but

NINT(-2.5)=-2

However, this is the CORRECT mathematical implementation of NINT, but differs from the behavior of the old version of the rexxmathlib.

SEE ALSO FLOOR INT CEIL FRACT

1.30 PI

NAME PI(x)

returns the value of PI. The argument is not used.

ARGUMENT REQUIREMENTS none

BUGS The result has a precision of 17 digits, although rexxmathlib provides only a precision of 15.9 digits. (and AREXX of 14 digits)

SEE ALSO E

1.31 POW,POWER,XTOY

NAME POW(x,y),POWER(x,y),XTOY(x,y)

return x to the power of y

ARGUMENT REQUIREMENTS messy... For non-integer y, x must be positive or zero. For integer y, x can be both positive or negative, however x and y must not be both zero. A second requirement is that both x and y must not be "to large".

BUGS 0 to the power of 0 is not allowed, although the old version of rexxmathlib can handle this. However, 0ˆ0 is mathematically not well defined and can be both, zero or one.

SEE ALSO ROOT

1.32 RAD

NAME RAD(x)

convert radiants in degrees

ARGUMENT REQUIREMENTS none

BUGS Due to round-off errors, the precision might be less than optimal, but is still better than what could be expected if converted manually by ARexx.

SEE ALSO ASIN ACOS ATAN DEG

1.33 ROOT

NAME ROOT(x,y)

return the y-th root of x

ARGUMENT REQUIREMENTS messy... For non-integer y, x must be positive or zero. For integer and odd y, x can be both positive or negative, y must be non-zero. A second requirement is that x must not be "to large" and y not "to small".

BUGS For y = 1, x is returned immediatly and for y = 2, the square-root function is used. All other arguments are passed to POW except for the extra sign handling of odd roots. This is a real mess...

SEE ALSO POW POWER XTOY

1.34 SEC

NAME SEC(x)

calculate the secans of the argument (in radiants)

ARGUMENT REQUIREMENTS none

BUGS Due to round-off errors, the secans of integer odd multiples of PI/2 is not infinity; additionally, the result is more or less random if the absolute value of x is "too large".

SEE ALSO COSEC CSC COT COTAN DEG

1.35 SIN

NAME SIN(x)

calculate the sine of the argument (in radiants)

ARGUMENT REQUIREMENTS none

BUGS Due to round-off errors, the sine of x of large absolute value is more or less random.

SEE ALSO COS TAN ACOS ASIN ATAN DEG

1.36 SINH

NAME SINH(x)

calculate the hyperbolic sine of the argument

ARGUMENT REQUIREMENTS -700 < x < 700 (approx.)

BUGS

SEE ALSO COSH TANH ACOSH ASINH ATANH

1.37 SQR,SQRT

NAME SQR(x),SQRT(x)

calculate the square root of x

ARGUMENT REQUIREMENTS x >= 0

BUGS

SEE ALSO ROOT POW POWER XTOY

1.38 TAN

NAME TAN(x)

calculate the tangent of the argument (in radiants)

ARGUMENT REQUIREMENTS none

BUGS Due to round-off errors, the tangent of PI/2 is not infinity; additionally, the result is more or less random if the absolute value of x is "too large".

SEE ALSO COS SIN ACOS ASIN ATAN DEG

1.39 TANH

NAME TANH(x)

calculate the hyperbolic tangent of the argument (in radiants)

ARGUMENT REQUIREMENTS none

BUGS

SEE ALSO COSH SINH ACOSH ASINH ATANH

1.40 Index