

Mathematical Functions

The following mathematical functions are supported in arithmetic processing statements (ADD, COMPUTE, DIVIDE, MULTIPLY, SUBTRACT) and in logical condition criteria:

Function	Format/Length	Explanation
ABS(<i>field</i>)	same as <i>field</i>	Absolute value of <i>field</i> .
ATN(<i>field</i>)	F8 (*)	Arc tangent of <i>field</i> .
COS(<i>field</i>)	F8 (*)	Cosine of <i>field</i> . If the value of the <i>field</i> is equal to or greater than 10^{17} , COS(<i>field</i>) will be "1".
EXP(<i>field</i>)	F8 (*)	Exponentiation of exponent <i>field</i> to base e, i.e. e^{field} , where e is Euler's number.
FRAC(<i>field</i>)	same as <i>field</i>	Fractional part of <i>field</i> .
INT(<i>field</i>)	same as <i>field</i>	Integer part of <i>field</i> .
LOG(<i>field</i>)	F8 (*)	Natural logarithm of <i>field</i> .
SGN(<i>field</i>)	same as <i>field</i>	Sign of <i>field</i> (-1, 0, +1).
SIN(<i>field</i>)	F8 (*)	Sine of <i>field</i> . If the value of the <i>field</i> is equal to or greater than 10^{17} , SIN(<i>field</i>) will be "0".
SQRT(<i>field</i>)	(*) (**)	Square root of <i>field</i> . A negative value in the argument field will be treated as positive. On mainframe computers, the maximum number of digits before the decimal point of the argument is 22.
TAN(<i>field</i>)	F8 (*)	Tangent of <i>field</i> . If the value of the <i>field</i> is equal to or greater than 10^{17} , TAN(<i>field</i>) will be "0".
VAL(<i>field</i>)	same as target field	Extract numeric value from an alphanumeric <i>field</i> . The content of the <i>field</i> must be the character representation of a numeric value. Leading or trailing blanks in the <i>field</i> will be ignored; decimal point and leading sign character will be processed. If the target field is not long enough, decimal digits will be truncated (see also Field Truncation and Field Rounding in the section Rules for Arithmetic Assignment of the Natural Statements documentation). Note: VAL cannot be used in conjunction with the COMPUTE statement.

* On UNIX and Windows platforms, these functions are evaluated as follows: The argument is converted to format/length F8 and then passed to the operating system for computation; the result returned by the operating system has format/length F8, which is then converted to the target format.

** On mainframe computers, the following applies:

If *field* has format/length F4, format/length of SQRT(*field*) will be F4;
if *field* has format/length F8 or I, format/length of SQRT(*field*) will be F8;
if *field* has format N or P, format/length of SQRT(*field*) will be Nn.7 or Pn.7 respectively (where n is automatically calculated to be large enough).

A *field* to be used with a mathematical function - except VAL - may be a constant or a scalar; its format must be numeric, packed numeric, integer, or floating point (N, P, I or F).

A *field* to be used with the VAL function may be a constant, a scalar, or an array; its format must be alphanumeric.

Mathematical Functions Example:

```
/* EXAMPLE 'MATHEX': MATHEMATICAL FUNCTIONS
*****
DEFINE DATA LOCAL
1 #A (N2.1) INIT <10>
1 #B (N2.1) INIT <-6.3>
1 #C (N2.1) INIT <0>
1 #LOGA (N2.6)
1 #SQRSTA (N2.6)
1 #TANA (N2.6)
1 #ABS (N2.1)
1 #FRAC (N2.1)
1 #INT (N2.1)
1 #SGN (N1)
END-DEFINE
*****
COMPUTE #LOGA = LOG(#A)
WRITE NOTITLE '=' #A 5X 'LOG' 40T #LOGA
*****
COMPUTE #SQRSTA = SQRT(#A)
WRITE      '=' #A 5X 'SQUARE ROOT' 40T #SQRSTA
*****
COMPUTE #TANA = TAN(#A)
WRITE      '=' #A 5X 'TANGENT' 40T #TANA
*****
COMPUTE #ABS = ABS(#B)
WRITE     // '=' #B 5X 'ABSOLUTE' 40T #ABS
*****
COMPUTE #FRAC = FRAC(#B)
WRITE      '=' #B 5X 'FRACTIONAL' 40T #FRAC
*****
COMPUTE #INT = INT(#B)
WRITE      '=' #B 5X 'INTEGER' 40T #INT
*****
COMPUTE #SGN = SGN(#A)
WRITE     // '=' #A 5X 'SIGN'      40T #SGN
*****
COMPUTE #SGN = SGN(#B)
WRITE      '=' #B 5X 'SIGN'      40T #SGN
*****
COMPUTE #SGN = SGN(#C)
WRITE      '=' #C 5X 'SIGN'      40T #SGN
*****
END
```

#A:	10.0	LOG	2.302585
#A:	10.0	SQUARE ROOT	3.162277
#A:	10.0	TANGENT	0.648360
#B:	-6.3	ABSOLUTE	6.3
#B:	-6.3	FRACTIONAL	-0.3
#B:	-6.3	INTEGER	-6.0
#A:	10.0	SIGN	1
#B:	-6.3	SIGN	-1
#C:	0.0	SIGN	0