HP Forum Archive 12

[Return to Index | Top of Index]

HP71 ROM module - functions ?A

*Message #1 Posted by **GE (France)** on 9 May 2003, 3:33 p.m.*

Hello, Thanks to Doug, a nice fellow met on this site I recently got a 71 Math module, but without doc. I have an old version of the HP Museum CDs, which don't carry the manual fot this module. Does anyone have pointers for me ? I do a CAT and it shows just a 32k LEX file, is this OK? TIA.

Re: HP71 ROM module - functions ?A

*Message #2 Posted by **Alain (Canada)** on 10 May 2003, 8:31 a.m.,*
in response to message #1 by GE (France)

Using cat, it returns the same thing for me. If you want to have the function list try using the lex "listlex".
The syntax is : listlex$("the name of lex"). It lists the keywords for the lex in parameter.

Also, you can have a look at Joseph K. Horn LEXicon by sorted by Lex Filename. See below for the MathROM.

url : http://pweb.jps.net/~joehorn/HP71/lexfiles.htm

Alain

MATHROM --/00 =pVER$ "MATH:1A"
MATHROM --/1E =pIMCHR
MATHROM --/1F =pIMXCH
MATHROM --/20 =pIMbck
MATHROM --/21 =pIMcpi
MATHROM --/22 =pIMcpw
MATHROM --/38 =pCMPLX
MATHROM --/39 =pREN
MATHROM --/3D =pFNIN
MATHROM --/3E =pFNOUT

MATHROM --/F1 =pMEM
MATHROM --/F2 =pERROR
MATHROM --/F7 =pZERPG
MATHROM --/FB =pCONFG
MATHROM 02/01 ACOSH(#)
MATHROM 02/02 ASINH(#)
MATHROM 02/03 ATANH(#)
MATHROM 02/04 BSTR$(#,#)
MATHROM 02/05 BVAL($,#)
MATHROM 02/06 CNORM(#())
MATHROM 02/07 COMPLEX #var(?#,?#),?#var(?#,?#),...
MATHROM 02/08 COSH(#)
MATHROM 02/09 DETL fn
MATHROM 02/0A DET(?#())
MATHROM 02/0B DOT(#(),#())
MATHROM 02/0C FGUESS fn
MATHROM 02/0D FNORM(#())
MATHROM 02/0E FNROOT(#,#,#) ffn
MATHROM 02/0F FVALUE fn
MATHROM 02/10 GAMMA(#)
MATHROM 02/11 IBOUND fn
MATHROM 02/12 IMPT(#)
MATHROM 02/13 INTEGRAL(#,#,#,#) ffn
MATHROM 02/14 IROUND(#)
MATHROM 02/15 IVALUE fn
MATHROM 02/16 LBND(#(),#)
MATHROM 02/17 LBOUND(#(),#)
MATHROM 02/18 LOG2(#)
MATHROM 02/19 MAT {matrix command}
MATHROM 02/1A NAN$(#)
MATHROM 02/1B NEIGHBOR(#,#)
MATHROM 02/1C REPT(#)
MATHROM 02/1D RNORM(#())
MATHROM 02/1E SCALE10(#,#)
MATHROM 02/1F SINH(#)

MATHROM 02/20 TANH(#)
MATHROM 02/21 UBND(#(),#)
MATHROM 02/22 UBOUND(#(),#)
MATHROM 02/23 CON postfix
MATHROM 02/24 IDN postfix
MATHROM 02/25 ZERO postfix
MATHROM 02/26 ZER postfix
MATHROM 02/27 INV postfix
MATHROM 02/28 TRN postfix
MATHROM 02/29 SYS postfix
MATHROM 02/2B FOUR postfix
MATHROM 02/2C PROOT postfix
MATHROM 02/2D FVAR fn
MATHROM 02/2E IVAR fn
MATHROM 02/2F CONJ(#)
MATHROM 02/30 TYPE(#/$)
MATHROM 02/31 ARG(#)
MATHROM 02/32 PROJ(#)
MATHROM 02/33 RECT(#)
MATHROM 02/34 POLAR(#)

[LONG] Re: HP71 ROM module - functions ?

*Message #3 Posted by **Valentin Albillo** on 12 May 2003, 5:33 a.m.,*
in response to message #1 by GE (France)

While you manage to get the manual or Quick User's Guide, I think this will help.

Best regards.

```
============================================================
#         HP-71 MATH ROM FUNCTIONS & SOME EXAMPLES           #
============================================================


------------------------------------------------------------
|                     COMPLEX OPERATIONS                   |
------------------------------------------------------------
```

```
COMPLEX A,B(2),C(3,3)
COMPLEX SHORT C(5,5)


    Complex variable creation with REAL precission (12+3), or SHORT (5+2)


IMAGE 2D,C(2D.2D,2D.2D,"i")


    Output formats for complex variables


All the following complex functions do work in CALC mode and leave
the result in RES:


Z=(2,3)         Assignment
Z=(2,3)+(4,5)   Sum
Z=(2,3)-(4,5)   Substraction
Z=(2,3)*(4,5)   Multiplication
Z=(2,3)/(4,5)   Division
Z=(2,3)^(4,5)   Raising to a power
Z=SQR((2,3))    Square root
Z=SIN((2,3))    Sine
Z=COS((2,3))    Cosine
Z=TAN((2,3))    Tangent
Z=SINH((2,3))   Hyperbolic sine
Z=COSH((2,3))   Hyperbolic cosine
Z=TANH((2,3))   Hyperbolic tangent
Z=EXP((2,3))    exponential
Z=LOG((2,3))    logarithm
R=REPT(Z)       real part of a complex
I=IMPT(Z)       imaginary part of a complex
Z=CONJ(Z)       complex conjugate
Z=ABS(Z)        Modulus
Z=ARG(Z)        Argument
Z=SGN(Z)        Unitary vector (SIGN if real)
Z=PROJ(Z)       Projectivity
Z=POLAR(Z)      Conversion rect -> polar
Z=RECT(Z)       Conversion polar -> rect


IF (2,3)=(3,4) THEN       Equals test
IF (2,3)#(3,4) THEN       Unequals test
```

```
-------------------------------------------------
|    ASSORTED REAL FUNCTIONS [X can't be COMPLEX]  |
-------------------------------------------------


A=SINH(X)       Hyperbolic sine (COMPLEX X allowed, too)
A=COSH(X)       Hyperbolic cosine (COMPLEX X allowed, too)
A=TANH(X)       Hyperbolic tangent (COMPLEX X allowed, too)
A=ASINH(X)      Hyperbolic arcsine
A=ACOSH(X)      Hyperbolic arccosine
A=ATANH(X)      Hyperbolic arctangent
A=GAMMA(X)      gamma function
A=LOG2(X)       base-2 logarithm
A$=BSTR$(N,B)   Converts N (base-10) to A$ in base B (2,8,or 16)
A=BVAL(N$,B)    Converts N$ (base B (2,8,or 16)) to A in base-10
A$=NAN$(N)      Gives the error contained in the NaN stored in N
N=NEIGHBOR(X,Y) Gives the successor [next machine representable number] of X
                in the direction of Y
N=SCALE10(X,Y)  Gives X times 10 raised to Y (i.e: scales X)
N=IROUND(X)     Rounds X based on the active OPTION ROUND
N=TYPE(X)       Gives the type of the variable X (X can be any machine variable)


-------------------------------------------------
|     MATRIX OPERATIONS                          |
-------------------------------------------------


NOTE: All matrix operations work with either real- or complex-valued matrices,
      unless otherwise specified.


DIM A(2),B(3,4)
REAL A(2),B(3,4)
COMPLEX A(2),B(3,4)


    Dimensions or redimensions matrices to be REAL precision (12+3)


SHORT A(2),B(3,4)
COMPLEX SHORT A(2),B(3,4)


    Dimensions or redimensions matrices to be SHORT precision (5+2)
```

```
INTEGER A(2),B(3,4)


    Dimensions or redimensions matrices to be INTEGER precision
    (-99999 to +99999). There are *no* COMPLEX INTEGER matrices.


DESTROY A(2),B(3,4)


    Destroys matrices or vectors


Note: for the following operations, the result matrix can be an
      argument too, i.e: MAT A=INV(A) or MAT A=A*A are allowed too,
      and mixing real and complex matrices in the same operation
      is allowed in most cases.


MAT A=ZER                 Fills up matrix A with zeros
MAT A=IDN                 Converts A to an identity matrix
MAT A=CON                 Fills up matrix A with ones
MAT A=(X)                 Fills up matrix A with the value of X
MAT A=B                   Matrix assignment
MAT A=-B                  Matrix change sign
MAT A=TRN(B)              Matrix transpose
MAT A=B+C                 Matrix addition
MAT A=B-C                 Matrix substraction
MAT A=B*C                 Matrix multiplication
MAT A=(X)*C               Multiplies all elements of a matrix by a number
MAT A=TRN(B)*C            Matrix transpose multiplication
MAT A=INV(B)              Matrix inverse
MAT X=SYS(A,B)            Solves all systems with coefficient matrix A and independent
                          terms B and places the solution matrix in X
MAT INPUT A,B             Matrix input
MAT DISP A,B              Matrix output to the display device
MAT DISP USING 1000;A,B   Same, but using a format image
MAT PRINT A,B             Matrix output to the printer device
MAT PRINT USING 1000;A,B  Same, but using a format image
X=DET(A)                  Returns matrix determinant [Note: only for real matrices]
X=DET                     Returns determinant of last matrix used in DET,INV,or SYS
X=DOT(A,B)                Dot product
X=RNORM(A)                Row norm of a matrix
X=CNORM(A)                Column norm of a matrix
X=FNORM(A)                Frobenius norm of a matrix
X=UBOUND(A,1)             Upper bound of the first dimension of a matrix
X=LBOUND(A,2)             Lower bound of the second dimension of a matrix
```

Example: Solve this system of equations

```
 2*a + b   + 3*c =  6
 5*a - b   + 4*c =  8
-3*a + 2*b -   c = -2
```

```
DESTROY ALL @ OPTION BASE 1 @ DIM A(3,3),B(3),X(3)
MAT INPUT A,B
                2,1,3,5,-1,4,-3,2,-1,6,8,-2 [ENTER]

MAT X=SYS(A,B) @ FIX 4 @ DELAY 0.5,0.5 @ MAT DISP X
```

```
----------------------------------------
|   ROOTS OF ARBITRARY FUNCTIONS         |
----------------------------------------
```

```
X=FNROOT(A,B,FNF(FVAR))
```

Finds a real root between A and B of the equation FNF(X) = 0
where FNF(X) is a user-defined function. If it can't find a root,
it returns a minimum of the function in that interval.

You can nest 5 calls to FNROOT, so solving a system of up to 5
non-linear equations or finding maxima and minima of a function
of up to 5 variables is possible.

FVAR represents the variable in the function, and holds its value.
It must be used in the definition, in lieu of the unknown.

FVALUE it's the value of the function at the computed root, so it
should be near zero por true roots, else the computed value
is a maximum or a minimum of the function, not a root

FGUESS it's the previous approximation to the root

The equation to be solved can be specified:

- in the call itself

```
X=FNROOT(1,2,FVAR^3-FVAR-1)
```

- in a user-defined function, single or multiline

```
10 DEF FNF(X)=X^3-X-1

X=FNROOT(1,2,FNF(FVAR))
```

INTEGRALS OF ARBITRARY FUNCTIONS

```
X=INTEGRAL(A,B,P,FNF(IVAR))
```

Computes the integral between limits A,B of the user-defined function FNF(X), using precission P.

You can nest 5 calls to INTEGRAL, so multiple integrals of up to 5 variables can be computed.

IBOUND gives the maximum error. If negative, the process did not converge to the specified precission.

IVAR represents the integration variable and stores its value. It must be used in the function definition in lieu of the integration variable.

IVALUE es the value of the last integral computed

The function to be integrated can be specified as follows:

- in the call itself:

```
               I=INTEGRAL(0,1,1E-5,SIN(IVAR)*COS(IVAR))

         - in a user-defined function, single or multiline

               10 DEF FNF(X)=SIN(X)*COS(X)

               I=INTEGRAL(0,1,1E-5,FNF(IVAR))



Note: INTEGRAL can use FNROOT in the function definition, and viceversa,
      so you can:

         - find roots of equations defined by integrals

         - integrate implicit functions


-----------------------------------------------
|   ROOTS OF POLYNOMIAL EQUATIONS             |
-----------------------------------------------


MAT R=PROOT(P)


   Given the vector P, which holds the coefficients of a polynomial
   equation of any degree >0, whose roots we want to find, it
   computes all roots, real and/or complex, and returns them stored
   in the complex vector R.


   Example: Find all roots of the following 5th-degree equation:


            x^5-3*x^4+8.1*x^2-1.37=0   (5 roots, 6 coefficients)


   DESTROY ALL @ OPTION BASE 1 @ DIM P(6) @ COMPLEX R(5)
   MAT INPUT P
                P(1)=?   1,-3,0,8.1,0,-1.37 [ENTER]
```

```
   MAT R=PROOT(P) @ FIX 4 @ DELAY 0.5,0.5 @ MAT DISP R


----------------------------------------
|  FOURIER TRANSFORMS                  |
----------------------------------------


MAT Z=FOUR(B)


   Computes either the direct or inverse Fourier transform for a series
   of data stored in matrix B, and returns the result in matrix Z.


   The number of elements in matrix B must be an integer power of 2,
   i.e: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, ...
```

END OF MESSAGE --

[Return to Index | Top of Index]

Go back to the main exhibit hall