HP 82480 Math Pac
Quick Reference Guide

For Use With the HP-71

Contents

Base Conversions

BVAL

Binary, Octal, or Hexadecimal to Decimal Conversion

BVAL(*S$*,*N*)

where *S$* is a binary, octal, or hexadecimal string expression and *N* is a numeric expression whose rounded integer value is 2, 8, or 16 respectively.

Converts a string expression *S$* representing a number expressed in base *N* into the equivalent decimal number. The value of the decimal equivalent can't exceed 999,999,999,999 (decimal).

Not usable in CALC mode.

BSTR$

Decimal to Binary, Octal, or Hexadecimal Conversion

BSTR$(*X*,*N***)**

where *X* is a numeric expression, 0 <= *X* < 999,999,999,999.5, and *N* is a numeric expression whose rounded integer value is 2, 8, or 16.

Not usable in CALC mode.

Complex Variables

Declaring Complex Variables

COMPLEX

Complex Variable Creation with 12-Digit Precision

COMPLEX *variable list*

where the syntax is the same as that used for **REAL**, **SHORT**, and **INTEGER** keywords. That is, *dim spec* means *numeric variable* [**(** *dim 1* [**,** *dim 2*] **)**], and *dim 1* and *dim 2* are real numeric expressions.

Not usable in CALC mode.

COMPLEX SHORT

Complex Variable Creation with 5-Digit Precision

COMPLEX SHORT *variable list*

where the syntax is the same as that used for **REAL**, **SHORT**, and **INTEGER** keywords.

Not usable in CALC mode.

Complex Operations

(,)

Real to Complex Conversion

(*X*,*Y*)

where *X* and *Y* are real- or complex-valued numeric expressions.

This is the way the HP-71 recognizes a complex number: as an ordered pair of real numbers. Since (X,Y) is defined as (real part of X, real part of Y), if either X or Y is complex, (X,Y) is not necessarily equivalent to $X + iY$.

Can be used in CALC mode.

REPT

Real Part of Complex Number

REPT(*Z*)

where *Z* is a real- or complex-valued numeric expression.

Returns the real part (first component) of Z. If Z is real, **REPT(*Z*)** $= Z$.

Can be used in CALC mode.

IMPT

Imaginary Part of Complex Number

IMPT(*Z*)

where *Z* is a real- or complex-valued numeric expression.

Returns the imaginary part (second component) of *Z*. If *Z* is real, **IMPT(*Z*)** = 0.

Can be used in CALC mode.

C(,)

Complex Field in an IMAGE String

[*n*]**C(***format string***)**

where *n* is an optional multiplier.

Causes a complex expression in a **DISP** or **PRINT** output list to be formatted according the the *format string*. The real part is formatted first and the imaginary part second. On output, the number is enclosed in parentheses, with the real and imaginary parts separated by a comma. The comma is sent out when the second numeric field is encountered.

Real Scalar Functions

Hyperbolic Functions

SINH(*real-valued numeric expression***)**	**Hyperbolic Sine**
COSH(*real-valued numeric expression***)**	**Hyperbolic Cosine**
TANH(*real-valued numeric expression***)**	**Hyperbolic Tangent**
ASINH(*real-valued numeric expression***)**	**Inverse Hyperbolic Sine**
ACOSH(*real-valued numeric expression***)**	**Inverse Hyperbolic Cosine**
ATANH(*real-valued numeric expression***)**	**Inverse Hyperbolic Tangent**

Other Functions Performing Calculations

GAMMA

Gamma Function

GAMMA(*X*)

where *X* is a real-valued numeric expression not equal to zero or a negative integer.

If X equals a positive integer, returns **FACT(*X*−1)**. Otherwise, returns $\Gamma(X)$.

Can be used in CALC mode.

LOG2

Base 2 Logarithm

LOG2(*X*)

where *X* is a real-valued numeric expression, $X > 0$.

Can be used in CALC mode.

SCALE10

Power of Ten Scaling

SCALE10(*X*,*P*)

where *X* is a real-valued numeric expression and *P* is a real numeric expression that must evaluate to an integer value.

Multiplies X by 10 raised to the power P by adding P to the exponent of X.

Can be used in CALC mode.

Integer Round

IROUND — Round to Integer

IROUND(*X*)

where *X* is a real-valued numeric expression.

Rounds X to an integer using the current **OPTION ROUND** setting.

Can be used in CALC mode.

Functions Providing Information

NAN$ — Not-a-Number Diagnostic Information

NAN$(*X*)

where *X* is a real-valued numeric expression.

Returns a string representing the error number contained in its **NaN** argument; that is, the number of the error that caused the **NaN** to be created. If X is not a **NaN**, then **NAN$(*X*)** returns a null string.

Not usable in CALC mode.

NEIGHBOR — Nearest Machine Number

NEIGHBOR(*X*,*Y*)

where *X* and *Y* are real-valued numeric expressions.

Returns the nearest machine-representable number to X in the direction toward Y.

Can be used in CALC mode.

TYPE

Expression Type and Dimension

TYPE(*X*)

where *X* is a real-, complex-, string-, or array-valued expression.

Returns an integer from 0 through 8 depending on the type and dimension of *X* as shown below.

Except for string and array arguments, can be used in CALC mode.

X	TYPE(*X*)
Simple real	0
Simple complex	1
Simple string	2
INTEGER array	3
SHORT array	4
REAL array	5
COMPLEX SHORT array	6
COMPLEX array	7
String array	8

Complex Functions and Operations

All the functions and operations described in this topic (except **ABS**, **ARG**, **CONJ**, and the relational operators) return a complex-type result.

With the exception of the **RECT** function, all complex numbers Z and W are assumed to be in rectangular, not polar, form.

Operators

+ Addition

$Z+W$

where Z and/or W are complex-valued numeric expressions.

Can be used in CALC mode.

− Unary Minus

$-Z$

where Z is a complex-valued numeric expression.

Can be used in CALC mode.

− Subtraction

$\mathrm{Z}-\mathrm{W}$

where Z and/or W are complex-valued numeric expressions.

Can be used in CALC mode.

* Multiplication

$Z*W$

where Z and/or W are complex-valued numeric expressions.

Can be used in CALC mode.

/ Division

Z/W

where Z and/or W are complex-valued numeric expressions, $W \neq (0,0)$.

Can be used in CALC mode.

^ Exponentiation

Z^*W*

where *Z* and/or *W* are complex-valued numeric expressions.

Can be used in CALC mode.

Logarithmic Functions

LOG Natural Logarithm

LOG(*Z*) or **LN(*Z*)**

where *Z* is a complex-valued numeric expression, $Z \neq (0,0)$.

Can be used in CALC mode.

EXP Exponential

EXP(*Z*)

where *Z* is a complex-valued numeric expression.

Can be used in CALC mode.

Trigonometric and Hyperbolic Functions

All trigonometric calculations take their arguments to be in radian measure regardless of the angular setting. All these functions can be used in CALC mode.

SIN(*complex-valued numeric expression***)**	**Sine**
COS(*complex-valued numeric expression***)**	**Cosine**
TAN(*complex-valued numeric expression***)**	**Tangent**
SINH(*complex-valued numeric expression***)**	**Hyperbolic Sine**
COSH(*complex-valued numeric expression***)**	**Hyperbolic Cosine**
TANH(*complex-valued numeric expression***)**	**Hyperbolic Tangent**

Polar/Rectangular Conversions

POLAR — Rectangular to Polar Conversion

POLAR(*Z***)**

where *Z* is a real- or complex-valued numeric expression.

Can be used in CALC mode.

RECT — Polar to Rectangular Conversion

RECT(*Z***)**

where *Z* is a real- or complex-valued numeric expression.

RECT is the only keyword in this topic, "Complex Functions and Operations," that assumes its argument *Z* to be in polar form.

Can be used in CALC mode.

General Functions

SQRT — Square Root

SQRT(*Z*) or **SQR(*Z*)**

where *Z* is a complex-valued numeric expression.

Returns the complex principal value of the square root of *Z*.

Can be used in CALC mode.

SGN — Unit Vector

SGN(*Z*)

where *Z* is a complex-valued numeric expression.

Returns the unit vector in the direction of *Z*.

Can be used in CALC mode.

ABS — Absolute Value

ABS(*Z*)

where *Z* is a complex-valued numeric expression.

Always returns real type.

Can be used in CALC mode.

ARG — Argument

ARG(*Z*)

where *Z* is a real- or complex-valued numeric expression.

Always returns real type.

Can be used in CALC mode.

CONJ

Complex Conjugate

> **CONJ(*Z*)**
>
> where *Z* is a real- or complex-valued numeric expression.

Always returns the same type (real or complex) as Z.

Can be used in CALC mode.

PROJ

Projective Infinity

> **PROJ(*Z*)**
>
> where *Z* is a real- or complex-valued numeric expression.

If $Z = x + iy$, then

$$\mathbf{PROJ}(Z) = Z \text{ if } \mathbf{ABS}(Z) \neq \mathbf{Inf}$$

or

$$\mathbf{PROJ}(Z) = \mathbf{Inf} + i0 \text{ if } \mathbf{ABS}(Z) = \mathbf{Inf}.$$

Can be used in CALC mode.

Relational Operators

=, <, >, #, ?

Equal or Unordered

> *Z comparison operator W*
>
> where *Z* and/or *W* are complex-valued numeric expressions.

When at least one of two expressions is complex valued, only two comparison results are possible: either the expressions are equal or they are unordered (or unequal, which is equivalent to unordered in this case).

Can be used in CALC mode.

Array Input and Output

Assignments

= Simple Assignment

MAT A=B

where **A** and **B** are both vectors or both matrices.

Array **B** may be either real or complex type.

If **B** is complex, then **A** must be complex.

If **B** is real, then **A** may be real or complex; if complex, all imaginary parts of all elements in **A** are set to zero.

Implicitly redimensions **A** to be the same size as **B**, and assigns the value of every element in **B** to the corresponding element in **A**.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

=() Numeric Expression Assignment

MAT A=(*X*)

where *X* is either a real- or complex-valued numeric expression.

If *X* is complex, then array **A** must be complex type.

If *X* is real, then **A** may be real or complex; if complex, all imaginary parts of all elements in **A** are set to zero.

Assigns *X* to all elements of **A**. Array **A** is not redimensioned.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

CON

Constant Array

MAT A=CON [(*X* [,*Y*])]

where **A** is either a real- or complex-type array, and the optional redimensioning subscripts *X* and *Y* are real-valued numeric expressions. *X* and *Y* are rounded to the nearest integer just as are subscripts in **DIM** statements.

Assigns the real value one to all elements of **A**. If redimensioning subscript(s) are provided, **A** is explicitly redimensioned according to the number and value of those subcripts.

Not usable in CALC mode.

IDN

Identity Matrix

MAT A=IDN [(*X*,*Y*)]

where **A** is either a real- or complex-type array and where the optional redimensioning subscripts *X* and *Y* are real-valued numeric expressions with the same rounded integer value. *X* and *Y* are rounded to the nearest integer just as are subscripts in **DIM** statements. If *X* and *Y* are not provided, **A** must be a square matrix (it must have two equal subscripts).

If no redimensioning subscripts *X* and *Y* are provided, then **A** will become an identity matrix. If redimensioning subscripts *X* and *Y* are provided, then **A** is explicitly redimensioned to a square matrix with the upper bound of each subscript equal to the rounded integer value of *X* and *Y* and then assigned the values of an identity matrix.

Not usable in CALC mode.

ZER

Zero Array

MAT A=ZER [(*X* [,*Y*])] or **MAT A=ZERO** [(*X* [,*Y*])]

where **A** is either a real- or complex-type array, and the optional redimensioning subscripts *X* and *Y* are real-valued numeric expressions. *X* and *Y* are rounded to the nearest integer just as are subscripts in **DIM** statements.

Assigns zero to all elements of **A**. If redimensioning subscript(s) are provided, **A** is explicitly redimensioned according to the number and value of those subcripts.

Not usable in CALC mode.

Array Input

INPUT

Assign Values from Keyboard Input

MAT INPUT A [,**B**]...

where **A** (and **B**) are real- or complex-type array(s).

Assigns real or complex numbers to the specified array(s). Complex values cannot be assigned to real array elements.

MAT INPUT prompts with the name of an array element and then accepts a numeric expression from the keyboard, evaluates that expression, and assigns the result as the value of that element. For each array, **MAT INPUT** gives prompts for the elements in row order (from left to right in each row, from the first row to the last). If there is more than one array, they are handled in the order specified.

When the name of an array element is displayed, enter its value by typing in the numeric expression and then pressing [END LINE]. You can enter values for several consecutive elements by separating the values with commas. When an array is filled, the remaining values are automatically entered into the next array. After you press $\boxed{\text{END LINE}}$, the computer will display the name of the next element (if any) to be assigned a value.

In other respects, **MAT INPUT** acts as does **INPUT**.

Not usable in CALC mode.

Array Output

To halt the operation of any of the keywords described below you need press $\boxed{\text{ATTN}}$ only once.

DISP

Display in Standard Format

MAT DISP A $\left[\begin{matrix} , \\ ; \end{matrix}\ \mathbf{B}\right] \ldots \left[\begin{matrix} , \\ ; \end{matrix}\right]$

where **A** (and **B**) are real- or complex-type array(s).

Displays the values of the elements of the specified arrays. The values are displayed in row order. Each row begins on a new line; a blank line is displayed between the last row of an array and the first row of the next array.

The choice of terminator—comma or semicolon—determines the spacing between the elements of an array as they do when **DISP** is used to display the results of numeric expressions.

If the last array specified doesn't have a terminator, the array will be displayed with wide spacing between elements.

Not usable in CALC mode.

PRINT

Print in Standard Format

MAT PRINT A $\left[\begin{matrix} , \\ ; \end{matrix}\ \mathbf{B}\right] \ldots \left[\begin{matrix} , \\ ; \end{matrix}\right]$

where **A** (and **B**) are real- or complex-type array(s).

Prints the values of the specified arrays. Operation is identical to **MAT DISP**, except that the output is sent to the **PRINTER IS** device, which requires HP-IL. If no **PRINTER IS** device is present, output is sent to the display, or to the HP-IL **DISPLAY IS** device. Also, you can override the CR/LF normally generated by **MAT PRINT USING** with the **ENDLINE** statement.

Not usable in CALC mode.

DISP USING **Display Using Custom Format**

MAT DISP USING $\left\{\begin{matrix}\textit{format string}\\\textit{line number}\end{matrix}\right\}$; **A** $\left[\begin{matrix},\\;\end{matrix}\ \textbf{B}\right]\ldots\left[\begin{matrix},\\;\end{matrix}\right]$

where **A** (and **B**) are real- or complex-type array(s).

Displays the values of the elements of the specified arrays in a format determined by the format string or by the specified **IMAGE** statement identified by the line number.

If any array is complex type, the corresponding field specifier in the format string or **IMAGE** statement must be a complex field specifier.

The values are displayed in row order. Each row begins on a new line; a blank line is displayed between the last row of an array and the first row of the next array.

The terminators between the arrays—commas or semicolons—serve only to separate the arrays and have no effect on the display format.

The Math Pac must be plugged in to **RENUMBER** a program containing a **MAT DISP USING** *[line number]* statement; otherwise, the line number will not be correctly updated.

Not usable in CALC mode.

PRINT USING

Print Using Custom Format

MAT PRINT USING $\begin{matrix}\textit{format string}\\ \textit{line number}\end{matrix}$;

$\mathbf{A}\left[\begin{matrix},\\ ;\end{matrix}\ \mathbf{B}\right]\ldots\left[\begin{matrix},\\ ;\end{matrix}\right]$

where **A** (and **B**) are real- or complex-type array(s).

Operation is identical to **MAT DISP USING**, except that the output is sent to the **PRINTER IS** device which requires HP-IL. If no **PRINTER IS** device is present, output is sent to the display, or to the HP-IL **DISPLAY IS** device. Also, you can override the CR/LF normally generated by **MAT PRINT USING** with the **ENDLINE** statement.

Not usable in CALC mode.

Array Arithmetic

Operators

= −

Negation

MAT A=−B

where **A** and **B** are both vectors or both matrices.

Array **B** may be either real or complex type.

If **B** is complex, then **A** must be complex.

If **B** is real, then **A** may be real or complex; if complex, all imaginary parts of all elements in **A** are set to zero.

Implicitly redimensions **A** to be the same size as **B** and assigns to each element of **A** the negative of the corresponding element of **B**.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

+ Addition

MAT A=B+C

where **A**, **B**, and **C** are all vectors or all matrices, and **B** and **C** are conformable for addition.

Arrays **B** and **C** may be either real or complex type.

If either **B** or **C** is complex, then **A** must be complex.

If both **B** and **C** are real, then **A** may be real or complex; if complex, all imaginary parts of all elements in **A** are set to zero.

Implicitly redimensions **A** to be the same size as **B** and **C**, and assigns to each element of **A** the sum of the values of the corresponding elements of **B** and **C**.

To halt operation, press ATTN twice.

Not usable in CALC mode.

− Subtraction

MAT A=B−C

where **A**, **B**, and **C** are all vectors or all matrices, and **B** and **C** are conformable for addition.

Arrays **B** and **C** may be either real or complex type.

If either **B** or **C** is complex, then **A** must be complex.

If both **B** and **C** are real, then **A** may be real or complex; if complex, all imaginary parts of all elements in **A** are set to zero.

Implicitly redimensions **A** to be the same size as **B** and **C**, and assigns to each element of **A** the difference of the values of the corresponding elements of **B** and **C**.

To halt operation, press ATTN twice.

Not usable in CALC mode.

()* Multiplication by a Scalar

MAT A=(*X*)*B

where **A** and **B** are both vectors or both matrices and *X* is a numeric expression.

Array **B** may be either real or complex type and expression *X* may be either real or complex valued.

If either **B** or *X* is complex, then **A** must be complex.

If both **B** and *X* are real, then **A** may be real or complex; if complex, all imaginary parts of all elements in **A** are set to zero.

Implicitly redimensions **A** to be the same size as **B** and assigns to each element of **A** the product of the value of *X* and the value of the corresponding element of **B**.

To halt operation, press ATTN twice.

Not usable in CALC mode.

* Matrix Multiplication

MAT A=B*C

where **B** is a matrix, **A** and **C** are both vectors or both matrices, and **B** and **C** are conformable for multiplication.

Arrays **B** and **C** may be either real or complex type.

If either **B** or **C** is complex, then **A** must be complex.

If both **B** and **C** are real, then **A** may be real or complex; if complex, all imaginary parts of all elements in **A** are set to zero.

Implicitly redimensions **A** to have the same number of rows as **B** and the same number of columns as **C**. The values of the elements of **A** are determined by the usual rules of matrix multiplication.

To halt operation, press ATTN twice.

Not usable in CALC mode.

TRN * — Transpose Multiplication

MAT A= TRN(B)*C

where **B** is a matrix, **A** and **C** are both vectors or both matrices, and **B** and **C** are conformable for transpose multiplication.

Arrays **B** and **C** may be either real or complex type.

If either **B** or **C** is complex, then **A** must be complex.

If both **B** and **C** are real, then **A** may be real or complex; if complex, all imaginary parts of all elements in **A** are set to zero.

Implicitly redimensions **A** to have the same number of rows as the number of columns in **B** and the same number of columns as **C**.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

Scalar-Valued Array Functions

Determinant Functions

DET — Determinant

DET(A)

where **A** is a square real-type matrix.

Returns the determinant of the matrix **A**.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

DETL — Determinant of Last Matrix

DETL or **DET**

Returns the determinant of the last real-type matrix that was:

- Inverted in a **MAT...INV** statement.
- Used as the first argument of a **MAT...SYS** statement.

DETL retains its value (even if the HP-71 is turned off) until another **MAT...INV** (with a real type argument) or a **MAT...SYS** (with a real type first argument) is executed.

Not usable in CALC mode.

Array Norms

CNORM — One-Norm (Column Norm)

CNORM(A)

where **A** is a real- or complex-type array.

Returns the maximum value (over all columns of **A**) of the sums of the absolute values of all elements in a column.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

RNORM — Infinity Norm (Row Norm)

RNORM(A)

where **A** is a real- or complex-type array.

Returns the maximum value (over all rows of **A**) of the sums of the absolute values of all elements in a row.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

FNORM

Frobenius (Euclidean) Norm

FNORM(A)

where **A** is a real- or complex-type array.

Returns the square root of the sum of the squares of the absolute values of all elements of **A**.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

Inner Product

DOT

Inner (Dot) Product

DOT(X,Y)

where **X** and **Y** are real- or complex-type vectors with the same number of elements.

Returns **X•Y**, the inner product of **X** and **Y**. If both **X** and **Y** are real, then the result is real. If either **X** or **Y** is complex, then the result is complex.

If **X** is a complex vector, then the complex conjugates of the elements of **X** are used to compute the inner product.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

Subscript Bounds

UBND

Subscript Upper Bound

UBND(A,*N*) or **UBOUND(A,*N*)**

where **A** is a real- or complex-type array and *N* is a numeric expression whose rounded integer value is 1 or 2.

Returns the upper bound of the *N*th (first or second) subscript of **A**. If **A** is a vector, **UBND(A,2)** = −1.

Not usable in CALC mode.

LBND

Subscript Lower Bound

LBND(A,*N*) or **LBOUND(A,*N*)**

where **A** is a real- or complex-type array and *N* is a numeric expression whose rounded integer value is 1 or 2.

Returns the **OPTION BASE** setting in effect when **A** was last dimensioned. If **A** is a vector, **LBND(A,2)** = −1.

Not usable in CALC mode.

Inverse, Transpose, and System Solution

Operations

INV

Matrix Inverse

MAT A=INV(B)

where **A** is a matrix and **B** is a square matrix.

Array **B** may be either real or complex type.

If **B** is complex, then **A** must be complex.

If **B** is real, then **A** may be real or complex; if complex, all imaginary parts of all elements in **A** are set to zero.

Implicitly redimensions **A** to be the same size as **B** and assigns to **A** the value of the matrix inverse of **B**.

To halt operation, press ATTN twice.

Not usable in CALC mode.

TRN

Matrix Transpose or Matrix Conjugate Transpose

MAT A=TRN(B)

where **A** and **B** are matrices.

Array **B** may be either real or complex type.

If **B** is complex, then **A** must be complex.

If **B** is real, then **A** may be real or complex; if complex, all imaginary parts of all elements in **A** are set to zero.

Implicitly redimensions **A** to be the same size as the matrix transpose of **B**. If **B** is real, assigns to **A** the value of the matrix transpose of **B**. If **B** is complex, assigns to **A** the values of the matrix conjugate transpose of **B**.

To halt operation, press ATTN twice.

Not usable in CALC mode.

Solving a System of Equations

SYS

System Solution

MATX=SYS(A,B)

where **A** is a square matrix, **X** and **B** are both vectors or both matrices, and **A** and **B** are conformable for multiplication.

Arrays **A** and **B** may be either real or complex type.

If either **A** or **B** is complex, then **X** must be complex.

If both **A** and **B** are real, then **X** may be real or complex; if complex, all imaginary parts of all elements in **X** are set to zero.

Implicitly redimensions **X** to be the same size as **B** and assigns to **X** the computed solution to the matrix equation **AX**=**B**.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

Solving *f*(*x*) = 0

FNROOT

Function Root

FNROOT(*A*,*B*,*F*)

where *A*, *B*, and *F* are real numeric expressions.

Returns the first value found (starting with guesses *A* and *B*) that is a root of the specified function or is the best approximation available.

Not usable in CALC mode

FVAR

Function Variable

FVAR

Represents the variable x in $f(x)$, the variable whose value **FNROOT** seeks.

Also returns the most current guess generated by a running **FNROOT**.

Can be used in CALC mode.

FVALUE

Function Value

FVALUE

Returns the value of the function *F* (the third argument of **FNROOT**) at the result generated by the most recently completed **FNROOT**. **FVALUE** retains its value, even if your HP-71 is turned off, until **FNROOT** is again completed.

Can be used in CALC mode.

FGUESS — Previous Estimate of Function Root

FGUESS

Returns the next-to-last value tried as a solution in the most recently completed **FNROOT** statement. **FGUESS** retains its value, even if your HP-71 is turned off, until **FNROOT** is again executed.

Can be used in CALC mode.

Numerical Integration

INTEGRAL — Definite Integral

INTEGRAL(*A*,*B*,*E*,*F*)

where *A*, *B*, *E*, and *F* are real numeric expressions.

Returns an approximation to the integral from *A* to *B* of *F*. The relative error *E* (rounded to the range 1E−12<=E<=1) indicates the accuracy of *F* and is used to calculate the acceptable error in the approximation of the integral.

Not usable in CALC mode.

IVAR — Integration Variable

IVAR

Represents the variable of integration in the formula defining *F*, the last argument of **INTEGRAL**.

Also contains the most recent sampling point used by a running **INTEGRAL**.

Can be used in CALC mode.

IVALUE — Last Result of INTEGRAL

```
IVALUE
```

Returns the last approximation computed by the **INTEGRAL** keyword. If the $\boxed{\text{ATTN}}$ key was pressed or the operation of **INTEGRAL** was otherwise interrupted, then **IVALUE** returns the value of the current approximation to the integral. Otherwise, **IVALUE** returns the same value that **INTEGRAL** last returned.

IVALUE retains its value (even if your HP-71 is turned off) until another **INTEGRAL** is computed.

Can be used in CALC mode.

IBOUND — Error Approximation for INTEGRAL

```
IBOUND
```

Returns the final absolute error estimate for the definite integral most recently computed by **INTEGRAL**.

- A positive value for **IBOUND** means that the approximations converged.
- A negative value for **IBOUND** means that the approximations didn't converge, so that the value returned by **INTEGRAL** may not be representative of the true value.

Like **IVALUE**, **IBOUND** retains its value (even if the HP-71 is turned off) until another **INTEGRAL** is computed. Unlike **IVALUE**, the value of **IBOUND** has no relation to the current approximation to the integral if the operation of **INTEGRAL** is interrupted.

Can be used in CALC mode.

Finding Roots of Polynomials

PROOT

Roots of a Polynomial

MAT R=PROOT(P)

where **P** is a real vector or matrix with *N* + 1 elements, where *N* = degree of polynomial whose roots are sought, and **R** is a complex vector or matrix.

If **R** is a vector, implicitly redimensions **R** to have *N* elements. If **R** is a matrix, implicitly redimensions **R** to have *N* rows and one column. **R** will be assigned the (complex) values of the solutions of the equation $P(x) = 0$ (where *P* is the polynomial of degree *N* whose coefficients are the values of the elements of **P**).

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

Finite Fourier Transform

FOUR

Finite Fourier Transform

MAT W=FOUR(Z)

where **Z** is an *N*-element complex array, either a vector or matrix, *N* is the number of complex data points, which must be a non-negative integer power of 2, and **W** is a complex array, either a vector or matrix.

If **W** is a vector, implicitly redimensions **W** to have *N* elements; if **W** is a matrix, implicitly redimensions **W** to have *N* rows and one column. **W** will be assigned the complex values of the finite Fourier transform of the data points represented by **Z**.

To halt operation, press [ATTN] twice.

Not usable in CALC mode.

Error Conditions

Math Pac Error Messages

Number **Error Message and Condition**

1 **#DIMS**

- **DOT(A,B)**: **A** or **B** is a matrix.
- **DET(A)**, **MAT B=INV(A)**, **MAT B=TRN(A)**, **MAT A=IDN**, **MAT X=SYS(A,Y)**: **A** or **B** is a vector.
- **MAT A=IDN(*i*)**: only one redimensioning subscript specified.
- **MAT A=*operation*(*operand array(s)*)**: number of subscripts of **A** not the same as the number of subscripts required for the result of the operation.

2 **Not Square**

- **DET(A)**, **MAT A=IDN**, **MAT B=INV(A)**, **MAT X=SYS(A,B)**: **A** is a matrix but the number of rows of **A** is not equal to the number of columns.
- **MAT A=IDN(*i*,*j*)**: $i \neq j$.

3 **Conformability**

- **MAT A=B+C**, **MAT A=B−C**: **B** and **C** not conformable for addition (the number of rows are unequal or the number of columns are unequal).
- **MAT A=B*C**: **B** and **C** not conformable for multiplication (**B** is a vector or the number of columns of **B** is not equal to the number of rows of **C**).
- **MAT A=TRN(B)*C**: **B** and **C** not conformable for transpose multiplication (**B** is a vector or the number of rows of **B** is not equal to the number of rows of **C**).

- **MAT X=SYS(A,B)**: Although **A** is a square matrix, **A** and **B** are not conformable for multiplication.
- **DOT(A,B)**: Although **A** and **B** are vectors, the number of elements of **A** is not equal to the number of elements of **B**.

4 **Parameter Redim**

- The result array of a **MAT** statement is a subprogram parameter. The statement requires array redimensioning, which changes the number of array elements.

5 **Nesting Error**

- More than five **FNROOT** or **INTEGRAL** keywords are nested.

6 **Kybd FN in FNROOT/INTEGRAL**

- Attempting to execute **FNROOT** or **INTEGRAL** from the keyboard in BASIC mode, and the function whose root or integral is sought is a user-defined function.
- Attempting to execute a user-defined function from the keyboard while an **FNROOT** or **INTEGRAL** execution is suspended during the evaluation of the function whose root or integral is sought.

7 **Function Interrupted**

- Interrupting **DET(A)**, **CNORM(A)**, **RNORM(A)**, **FNORM(A)**, or **DOT(A,B)** by pressing [ATTN] twice.

8 **Bad Array Size**

- **MAT B=FOUR(A)** where the number of elements of **A** is not a non-negative integral power of two.
- **MAT B=PROOT(A)** where **A** has only one element.

9 **PROOT Failure**

- **PROOT** failed to find a root.

10 **GAMMA=Inf**

- **GAMMA(X)** where **X** is a non-positive integer.

11 **ATANH(+−1)**

- **ATANH(1)** or **ATANH(−1)**

No error number **Initialization**

- The Math ROM cannot initialize due to insufficient memory.

  This ROM requires 43.5 bytes of user memory for its own use.

  This memory must be available before plugging in the module.

HP-71 Error Messages

Number **Error Message and Condition**

11 **Invalid Arg**

- **BVAL(*B$*,*R*)**, **BSTR$(*X*,*R*)**: rounded integer value of R not equal to 2, 8, or 16.
- **BVAL(*B$*,*R*)**: *B$* not a valid string representation of a number in base *R*.
- **BSTR$(*X*,*R*)**: The rounded integer value of *X* is not in the interval [0,1E12).
- **BVAL(*B$*,*R*)**: value > 999,999,999,999.
- **LBND(A,*N*)**, **UBND(A,*N*)**: rounded integer value of *N* is not equal to 1 or 2.
- An illegal subscript in a **MAT CON**, **MAT IDN**, **MAT ZER**, **COMPLEX**, or **COMPLEX SHORT** statement.

24 **Insufficient Memory**

- Appendix B in the Math Pac Owner's Manual gives the memory requirements for various Math Pac operations.

31 **Data Type**

- A scalar (real or complex) has been used where an array is required or vice-versa.
- A complex type (scalar or array) has been used where a real type (scalar or array) is required or vice-versa.

32 **No Data**

- Attempting to execute **DETL** before the first completion of **MAT...INV** with a real-type argument or **MAT...SYS** with a real-type first argument.
- Attempting to execute **FVALUE** or **FGUESS** before the first completion of an **FNROOT** keyword.
- Attempting to execute **IVALUE** or **IBOUND** before any **INTEGRAL** keyword has completed the first evaluation of the function whose integral is sought.
- Attempting to execute **FVAR** while no **FNROOT** is evaluating the function whose root is sought.
- Attempting to execute **IVAR** while no **INTEGRAL** is evaluating the function whose integral is sought.

46 **Invalid USING**

- Formatting a real expression with a complex **IMAGE** field or vice-versa.

79 **Illegal Context**

- Attempting to execute **INTEGRAL** or **FNROOT** from CALC mode in any way except by direct execution.

80 **Invalid Parameter**

- **MAT INPUT** attempts to execute an expression in the **MAT INPUT** response line where that expression calls a user-defined function.

HEWLETT
PACKARD

Portable Computer Division
1000 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.

82480-90002 English Printed in S'pore 3/84