Math Pac 2

Owner's Manual Supplement

For the HP-71

Math Pac 2

Owner's Manual Supplement

For Use with the HP-71

March 2020

NOTICE

The information contained in this document is subject to change without notice.

This document and the product described in this document have not been produced by Hewlett-Packard. Information on the origin of the product described in this document are provided in the Web site indicated below.

Web site: http://www.jeffcalc.hp41.eu/emu71/mathrom.html

Edition History

Preliminary Edition . . . . March 2020

Introducing the Math Pac 2

The Math Pac 2 is an enhanced version of the 1984 Math Pac for the HP-71 Computer.
It provides the same set of powerful tools than the previous version for solving a wide range of mathematical, scientific and engineering problems.

The Math Pac 2 adds several complements to get a comprehensive set of mathematical operations by integrating the best features from other computers/calculators such as the HP-75 Computer, the Series 80 Desktop Computers, the Series 200/300 HP Basic, the 28C/S Advanced Scientific Calculators and the HP-42S RPN Scientific Calculator, to reach state-of-the-art levels in term of mathematical tools in a single Basic environment. You never will have to use a RPN/RPL calculator or a bulky HP desktop computer for special computing purposes [1].

The Math Pac 2 adds the following capabilities to your HP-71:

- Full set of the Math Pac 1A capabilities.
- Additional real- and complex-valued functions.
- Additional array functions.
- Additional matrix operations.
- Improvements of several functions.
- New keywords for functional compatibility with other computers and calculators.

1. This should not be taken too seriously, RPN/RPL calculators are very good and the HP Basic (also known as Rocky Mountain Basic) is a powerful language for technical and instrumentation applications still today.

Contents

How to Use This Manual

This manual supplement assumes that you are familiar with the operation of the 1984 Math Pac 1A, which is not described again in this manual. Only the new capabilities and keywords provided in the Math Pac 2 are described here. Please refer to the 1984 Math Pac Owner's Manual for a reference to the original capabilities. However, the keyword index at the end of this manual summarizes all the capabilities of the Math Pac 2 including the original Math Pac capabilities.

Additional Information

Additional information can be found in the manuals for the HP-71, HP-75, Series 80, Series 200/300 Basic and HP calculators, specifically:

- The HP-71 Math Pac Owner's Manual.
- The HP-75 Math Pac Owner's Manual.
- The Series 80 Matrix ROM Manual.
- The Series 200/300 Basic Language Reference Manual.

Section 1

Installing the Math 2 LEX File

The Math 2 LEX file is installed from the Math 2 Distribution Pac.

CAUTIONS

- Don't install the Math 2 LEX together with the HP71 Math ROM. This will definitively cause problems.
- When installing the Math 2 LEX for the first time, and if a Math ROM was previously installed, be sure to turn on the HP71 AFTER removing the Math ROM, but BEFORE installing the Math Pac 2 (that is with none of the Math ROM or Math 2 LEX present). This will guarantee that the new Math 2 LEX will be correctly initialized.
  This step is not needed when later installing new versions of the Math 2 LEX.
- The Math Pac 2 is backward compatible with the Math ROM (old programs will run correctly on the Math Pac 2), but programs using the new Math Pac 2 capabilities should not be run with the previous Math ROM installed. See Appendix for details.

Please refer to the Math 2 Distribution Pac documentation for how to install the Math 2 LEX file in your HP-71 or in a HP-71 emulator.

Section 2

New Real- and Complex-Value Functions

Real-Value Trigonometric Functions

The three functions described below are for HP-75 and Series 80 compatibility.

SEC **Secant**

`SEC(X)` Where `X` is a real-value numeric expression. The secant is defined by `SEC(X)=1/COS(X)` in the current angle mode. Can be used in CALC mode.

CSC **Cosecant**

`CSC(X)` Where `X` is a real-value numeric expression. The cosecant is defined by `CSC(X)=1/SIN(X)` in the current angle mode. Can be used in CALC mode.

COT **Cotangent**

`COT(X)` Where `X` is a real-value numeric expression. The cotangent is defined by `COT(X)=1/TAN(X)` in the current angle mode. Can be used in CALC mode.

Complex-Value Inverse Hyperbolic Functions

The Math Pac 2 adds the support of complex arguments in the inverse hyperbolic functions.

ASINH **Inverse Hyperbolic Sine**

`ASINH(Z)`

Where `Z` is a complex-value numeric expression.
Returns the complex principal value of the inverse hyperbolic sine of `Z`.
The complex inverse hyperbolic sine is defined by:
`ASINH(Z)=LOG(Z+SQRT(1+Z^2))` [1]

Can be used in CALC mode.

ACOSH **Inverse Hyperbolic Cosine**

`ACOSH(Z)`

Where `Z` is a complex-value numeric expression.
Returns the complex principal value of the inverse hyperbolic cosine of `Z`.
The complex inverse hyperbolic cosine is defined by:
`ACOSH(Z)= SGN(IMPT(Z))*LOG(Z+(0,1)*SQRT(1-Z^2))` [1]

Can be used in CALC mode.

ATANH **Inverse Hyperbolic Tangent**

`ATANH(Z)`

Where `Z` is a complex-value numeric expression. `Z≠(±1,0)`
Returns the complex principal value of the inverse hyperbolic tangent of `Z`.
The complex inverse tangent is defined by:
`ATANH(Z)=LOG((1+Z)/(1-Z))/2` [1]

Can be used in CALC mode.

1. The Math Pac 2 internally uses special algorithms to provide accurate results over the whole argument range.

Complex-Value Inverse Trigonometric Functions

The Math Pac 2 adds the support of complex arguments in the inverse trigonometric functions. `ASN/ACS/ATN` are for compatibility with the Series 200/300 HP Basic.

ASIN/ASN **Inverse Sine**

`ASIN(Z)` or `ASN(Z)`

Where `Z` is a complex-value numeric expression.
Returns the complex principal value of the inverse sine of `Z`.
The complex inverse sine is defined by:
`ASIN(Z)=(0,-1)*ASINH((0,1)*Z)` [1]

Can be used in CALC mode.

ACOS/ACS **Inverse Cosine**

`ACOS(Z)` or `ACS(Z)`

Where `Z` is a complex-value numeric expression.
Returns the complex principal value of the inverse cosine of `Z`.
The complex inverse cosine is defined by:
`ACOS(Z)= SGN(IMPT(Z))*(0,-1)*ACOSH(Z)` [1]

Can be used in CALC mode.

ATAN/ATN **Inverse Tangent**

`ATAN(Z)` or `ATN(Z)`

Where `Z` is a complex-value numeric expression, `Z≠(0,±1)`
Returns the complex principal value of the inverse tangent of `Z`.
The complex inverse tangent is defined by:
`ATAN(Z)=(0,-1)*ATANH((0,1)*Z)` [1,2]

Can be used in CALC mode.

1. The Math Pac 2 internally uses special algorithms to provide accurate results over the whole argument range.
2. Error "ATANH=Inf" is generated for `Z=(0,1)` and `Z=(0,-1)`.

Complex-Value Base-10 Logarithm Function

The Math Pac 2 adds the support of complex arguments in the base-10 logarithm function for compatibility with Series 200/300 HP Basic `LGT` function and the RPN/RPL calculators `LOG` function.

LGT/LOG10 **Base-10 Logarithm**

`LGT(Z)` or `LOG10(Z)`

Where `Z` is a complex-value numeric expression, `Z≠(0,0)`
The complex base-10 logarithm is defined by `LGT(Z)=LOG(Z)/LOG(10)`

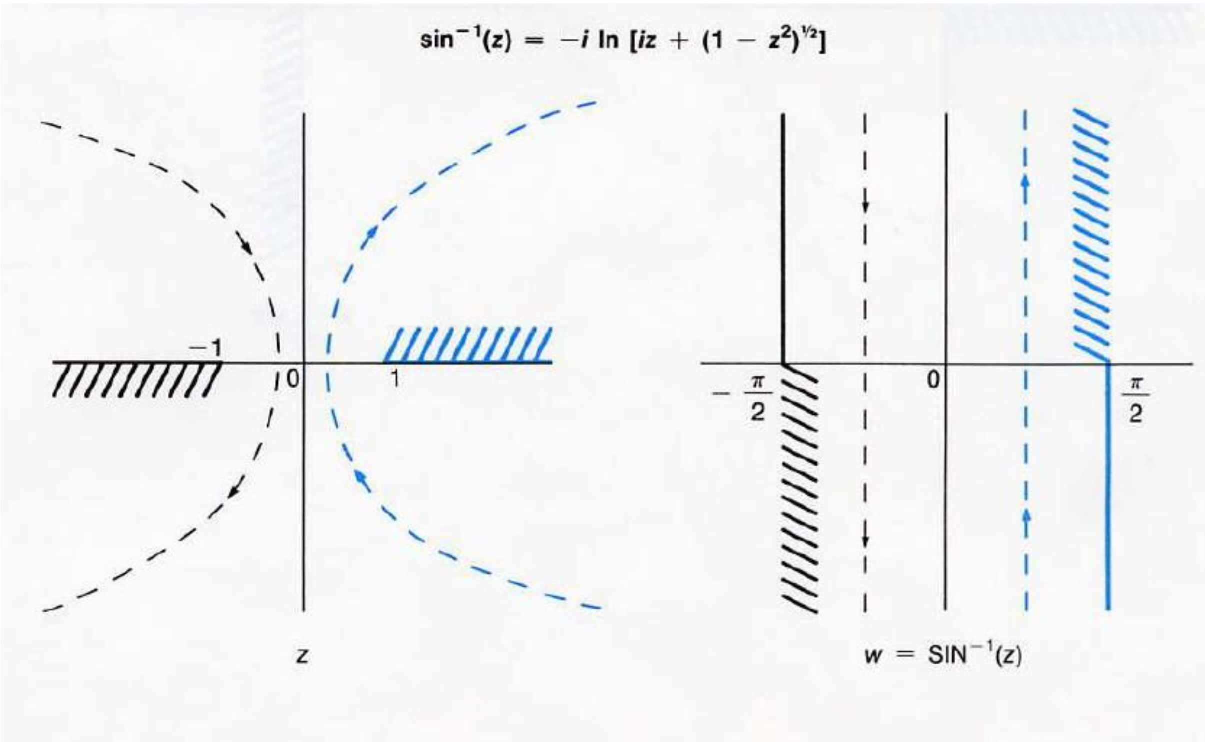
Can be used in CALC mode.

Additional Information [1]

For some inverse functions, the definitions of the principal branches are not universally agreed upon. The branches used by the Math Pac were chosen such as they are analytic in the regions where their real-valued counterparts are defined; that is, the branch cut occurs where the real-value inverse is undefined. In addition, most of the important symmetries are preserved. For example:

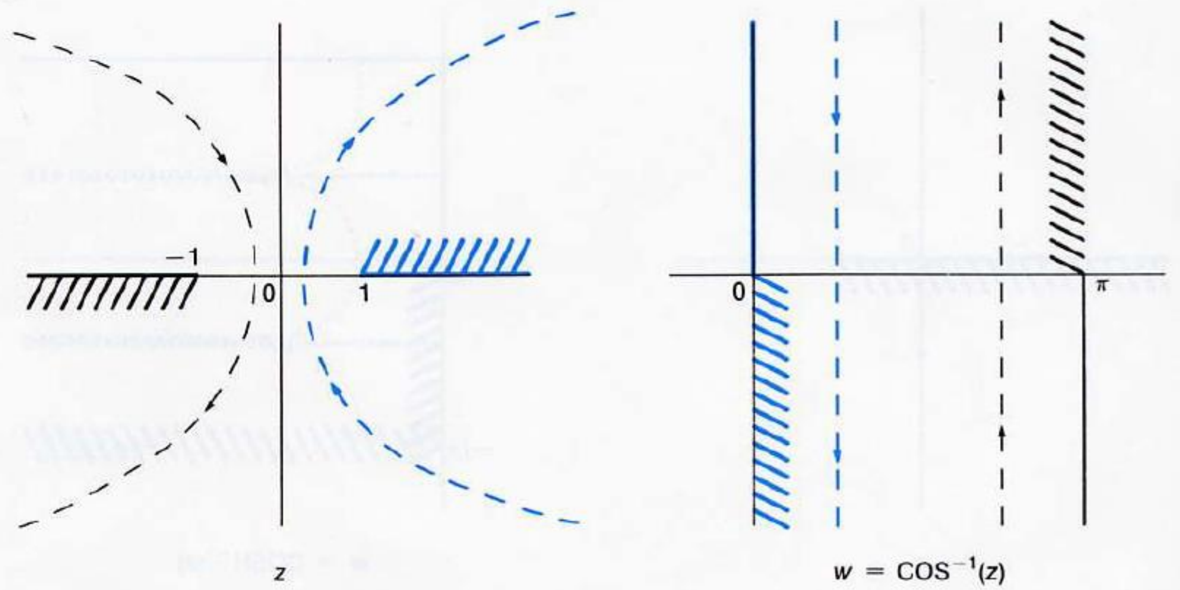
$$\mathrm{SIN}^{-1}(-z) = -\mathrm{SIN}^{-1}(z) \text{ for all } z.$$

The illustrations that follow show the principal branches of the new inverse functions that the Math Pac 2 calculates.

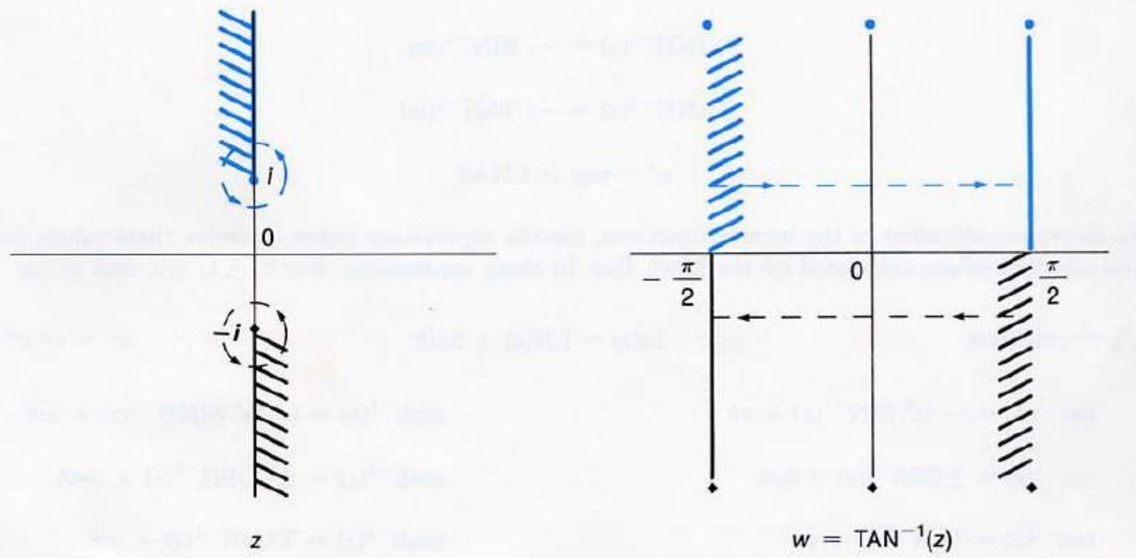


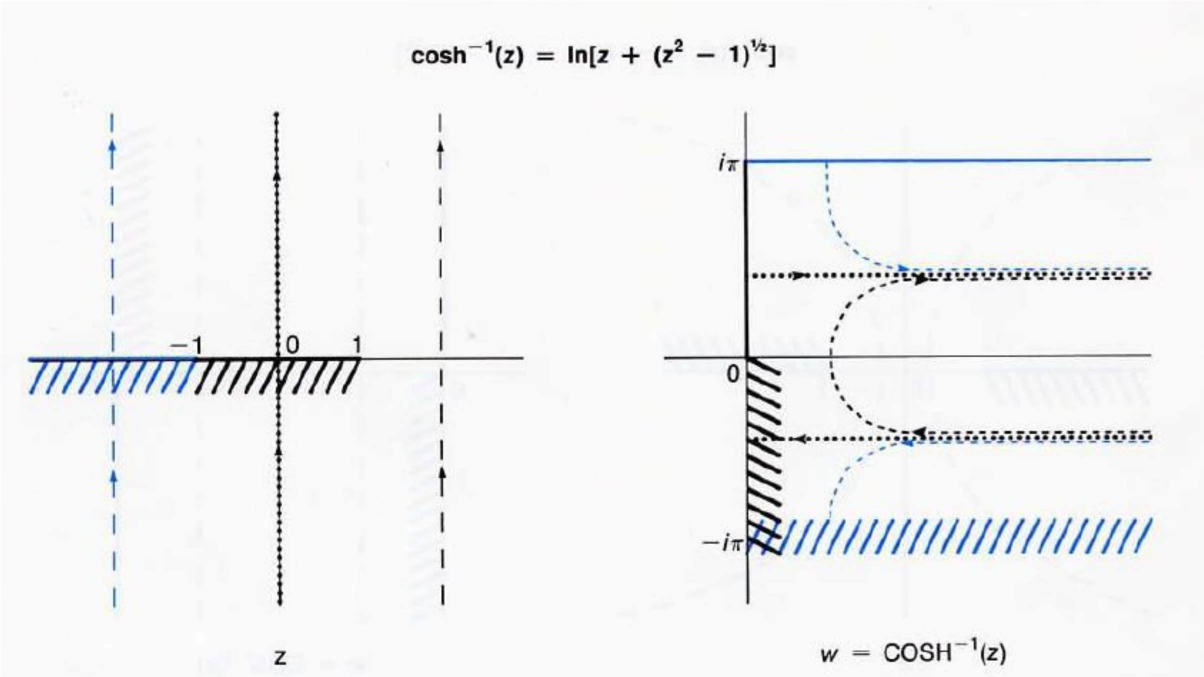
1. The information in this section is adapted from the HP-75 Math Pac Owner's Manual.

$$\cos^{-1}(z) = -i \ln [z + (z^2 - 1)^{1/2}]$$



$$\tan^{-1}(z) = \frac{i}{2} \ln \frac{i + z}{i - z}$$





The principal branches in these four graphs above are obtained from the equations shown, but don't necessarily use the principal branches of ln(z) and sqr(z).

The remaining inverse functions may be determined from the illustrations above and the following equations:

$$\text{SINH}^{-1}(z) = -i\ \text{SIN}^{-1}(iz)$$

$$\text{TANH}^{-1}(z) = -i\ \text{TAN}^{-1}(iz)$$

To determine all values of the inverse functions, use the expressions below to derive these values from the principal values calculated by the Math Pac 2. In these expressions, k=0, ±1, ±2, and so on.

$$\sin^{-1}(z) = (-1)^k\ \text{SIN}^{-1}(z) + \pi k$$

$$\cos^{-1}(z) = \pm\text{COS}^{-1}(z) + 2\pi k$$

$$\tan^{-1}(z) = \text{TAN}^{-1}(z) + \pi k$$

$$\sinh^{-1}(z) = (-1)^k\ \text{SINH}^{-1}(z) + \pi i k$$

$$\cosh^{-1}(z) = \pm\text{COSH}^{-1}(z) + 2\pi i k$$

$$\tanh^{-1}(z) = \text{TANH}^{-1}(z) + \pi i k$$

Section 3

New Scalar-Valued Array Functions

Array Element Sum Functions

The SUM function provides compatibility with the HP-75, Series 80 and 200/300 HP Basic. The ABSUM function is for HP-75 and Series 80 compatibility.

SUM **Array Element Sum**

`SUM(A)`

Where `A` is a real- or complex-type array.
Returns the sum of the values of all elements in the array.

To halt operation, press [ATTN] twice.
Not usable in CALC mode.

ABSUM **Array Element Absolute Value Sum**

`ABSUM(A)`

Where `A` is a real-type array.
Returns the sum of the absolute values of all elements in the array.

To halt operation, press [ATTN] twice.
Not usable in CALC mode.

Array Minimum/Maximum Element Functions

The four functions described below are for HP-75 and Series 80 compatibility.

AMIN **Array Element Minimum**

```
AMIN(A)
```

Where A is a real-type array.
Returns the value of the minimum element in the array.
Store the numbers of the row and column where the element is found for AROW and ACOL.

To halt operation, press [ATTN] twice.
Not usable in CALC mode.

AMAX **Array Element Maximum**

```
AMAX(A)
```

Where A is a real-type array.
Returns the value of the maximum element in the array.
Store the numbers of the row and column where the element is found for AROW and ACOL.

To halt operation, press [ATTN] twice.
Not usable in CALC mode.

MINAB **Array Element Minimum Absolute Value**

```
MINAB(A)
```

Where A is a real-type array.
Returns the value of the smallest element (in absolute value) in the array.
Store the numbers of the row and column where the element is found for AROW and ACOL.

To halt operation, press [ATTN] twice.
Not usable in CALC mode.

MAXAB **Array Element Maximum Absolute Value**

```
MAXAB(A)
```

Where A is a real-type array.
Returns the value of the largest element (in absolute value) in the array.
Store the numbers of the row and column where the element is found for AROW and ACOL.

To halt operation, press [ATTN] twice.
Not usable in CALC mode.

Array Row/Column Functions

The two functions described below are for Series 80 compatibility.

AROW **Array Row**

`AROW` Returns the value of the row set by the last execution of one of the functions `RNORM`, `AMIN`, `AMAX`, `MINAB` or `MAXAB`. Usable in CALC mode.

ACOL **Array Column**

`ACOL` Returns the value of the column set by the last execution of one of the functions `CNORM`, `AMIN`, `AMAX`, `MINAB` or `MAXAB`. Usable in CALC mode.

Additional Information

The `AROW` function combines the Series 80 `RNORMROW/AMINROW/AMAXROW/MAXABROW` functions in a single keyword. The `ACOL` function combines the `CNORMCOL/AMINCOL/AMAXCOL/MAXABCOL` functions in a single keyword.

If more than one element is found in the `AMIN/AMAX/MINAB/MAXAB` functions, the element in the lowest-numbered row is chosen, and this number is returned as the value of the function `AROW`. If both such elements are in the same row, the element in the lowest-numbered column is chosen, and this number is returned as the value of the function `ACOL`.

The error "`No Data`" is generated if the `AROW` or `ACOL` function is executed before resp. the `RNORM` or `CNORM` function, and before any of the `AMIN`, `AMAX`, `MINAB` or `MAXAB` functions, since the last installation of the Math Pac 2.

Example [1]

```
5 ! Example adapted from the HP-85 Matrix ROM Manual
10 OPTION BASE 1
20 DIM A(3,3),B(3,3),V1(3),V2(3)
30 DATA -3,2,3,5,-3,5,2,5,-1
40 DATA 2,1,3,1,4,2
70 READ A(,)
80 READ V1(),V2()
90 MAT PRINT A;V1,V2
100 PRINT ABSUM(A)
110 PRINT AMAX(A)
120 PRINT ACOL
130 PRINT AROW
140 PRINT AMIN(A)
150 PRINT ACOL
160 PRINT AROW
170 PRINT CNORM(A)
180 PRINT ACOL
190 PRINT DET(A)
200 MAT B=INV(A)
210 PRINT DETL
220 PRINT DOT(V1,V2)
230 PRINT FNORM(A)
240 PRINT LBND(A,1)
250 PRINT MAXAB(A)
260 PRINT ACOL
270 PRINT AROW
280 PRINT RNORM(A)
290 PRINT AROW
300 PRINT SUM(A)
310 N=2
320 PRINT UBND(A,N)
330 END
```

```
>RUN

-3  2  3    Array A
 5 -3  5
 2  5 -1

 2          Vector V1
 1
 3
```

1. Except from the lines with the `AROW`/`ACOL` keywords, this program can also be run on the HP75 with the Math Pac.

1 Vector V2
4
2

29 `ABSUM(A)`: the sum of the absolute values of the elements in A.

5 `AMAX(A)`: the value of the largest element in A.

1 `ACOL`: the lowest-numbered column containing `AMAX(A)`.

2 `AROW`: the lowest-numbered row containing `AMAX(A)`.

-3 `AMIN(A)`: the value of the smallest element in A.

1 `ACOL`: the lowest-numbered column containing `AMIN(A)`.

1 `AROW`: the lowest-numbered row containing `AMIN(A)`.

10 `CNORM(A)` : the largest sum of the absolute values of the elements in each column of A.

1 `ACOL`: the lowest-numbered column with the largest sum of absolute values.

189 `DET(A)`: Determinant of A.

189 `DETL`: Determinant of matrix A inverted in preceding `MAT...INV` statement.

12 `DOT(V1,V2)`: the sum of the products of corresponding elements of V1 and V2.

10.54 `FNORM(A)`: the square root of the sum of the squares of the elements in A.

1 `LBND(A,1)`: the lower bound of the first subscript of A.

5 `MAXAB(A)`: the largest absolute value of any element in A.

1 `ACOL`: the lowest-numbered column containing the element with largest absolute value.

2 `AROW`: the lowest-numbered row containing the element with largest absolute value.

13 `RNORM(A)`: the largest sum of the absolute values of the elements in each row of A.

2 `AROW`: the lowest-numbered row with the largest sum of absolute values.

15 `SUM(A)`: the sum of the elements in A.

3 `UBND(A,2)`: the upper bound of the second subscript of A.

Section 4

New Matrix Operations

Matrix Operations

The new operations provide compatibility with the HP-75, Series 80 and 200/300 HP Basic.

TBA

Section 5

Other Pac 2 Improvements

Complex Square Operation

With the Math Pac 1A, a square operation on a complex number such as Z^2 is internally computed as exp(2*ln(Z)).
With complex numbers, this can limit the accuracy in some cases, such as:

```
> (1E-9,1)^2
(-1,2.00000769645E-9)
```

With the Math Pac 2, the complex square operation Z^2 is now internally computed as Z*Z to provide the same performance of speed and accuracy for complex numbers than the [x²] key on RPN/RPL calculators (15C, 28C/S, 42S and others).

```
> (1E-9,1)^2
(-1,2E-9)
```

It is still possible to use the general exponentiation formula by forcing a complex power:

```
> (1E-9,1)^(2,0)
(-1,2.00000769645E-9)
```

Moreover, the special case is not based on a pattern detected during the expression parsing but at execution, so an expression like:

`X=2 @ Z=(1E-9,1)^X` is handled as a multiplication rather than an exponentiation.

Note that the new square functionality operates only on complex numbers, a real square operation like X^2 is still computed by the HP-71 as exp(2*ln(X)). This generally doesn't affect the accuracy of the result.

A side improvement of the new complex square functionality is that the signed 0 (a specific HP71 feature) is now conserved by the complex square operation:

```
> (-0,1)^(2,0)
(-1,0)             the sign of 0 is not conserved with the Math 1A square function [1]
```

```
> (-0,1)^2
(-1,-0)            the sign of 0 is now conserved
```

1. This limitation was mentioned by Prof. W. Kahan in his article: "Branch Cuts for Complex Elementary Functions", 1986

RNORM Row and CNORM Column

The RNORM and CNORM functions now store the number of the row (for RNORM) or column (for CNORM) that has the largest sum of absolute values (the norm), these numbers are then available with the new AROW and ACOL functions, until the execution of another function that changes the stored row/column numbers (see section 3).

This functionality is similar to the RNORMROW and CNORMCOL functions of the Series 80.

If more than one row/column (resp. for RNORM/CNORM) has the largest sum of absolute values, the lowest-numbered row/column is chosen and this number is returned as the value of the function AROW/ACOL.