12

Using Matrices

Introduction

You can perform matrix calculations in HOME and in programs. The matrix *and each row* of a matrix appear in brackets, and the elements and rows are separated by commas. For example, for the following matrix:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

is displayed in the history as:
[[1,2,3],[4,5,6]]

(If the Decimal Mark in MODES is set to Comma, then the row separators are periods.)

You can enter matrices directly in the command line, or create them in the matrix editor.

Vectors

Vectors are one-dimensional arrays. They are composed of just one row. A vector is represented with single brackets; for example, [1,2,3]. A vector can be a real number vector or a complex number vector, for example [(1,2), (7,3)].

Matrices

Matrices are two-dimensional arrays. They are composed of more than one row and more than one column. Two-dimensional matrices are represented with nested brackets; for example, [[1,2,3],[4,5,6]]. You can create complex matrices, for example, [[(1,2), (3,4)], [(4,5), (6,7)]].

Matrix Variables

There are ten matrix variables available, named M0 through M9. You can use them in calculations or manipulations in HOME or in a program. You can retrieve the matrix names from the VARS menu, or just type their names from the keyboard.

Creating and storing matrices

You can create, edit, delete, send, and receive matrices in the Matrix catalog.

To open the Matrix catalog, press SHIFT *MATRIX*.

You can also create and store matrices—named or unnamed—-in HOME. For example, the command:

POLYROOT([1,0,-1,0])►M1

stores the complex vector of length 3 into the M1 variable. M1 now contains the three roots of
$x^3 - x = 0$

You use the Matrix catalog to:

- create a new matrix
- edit an existing matrix
- send and receive matrices to a PC or another calculator
- delete the contents of a matrix

Matrix Catalog keys

The table below summarizes the operations of the menu keys in the Matrix Catalog, as well as the use of Delete (DEL) and Clear (SHIFT *DEL*).

Key	Meaning
EDIT	Opens the highlighted matrix for editing.
NEW	Prompts for a matrix type, then opens an empty matrix with the highlighted name.
SEND	Transmits the highlighted matrix to another HP 39G/40G or a disk drive. See “Sending and receiving aplets” on page 16-242.
RECV	Receives a transmitted matrix from another HP 39G/40G or a disk drive. See “Sending and receiving aplets” on page 16-242.
DEL	Clears the highlighted matrix.
SHIFT *CLEAR*	Clears all matrices.
SHIFT ▼ or ▲	Moves to the end or the beginning of the catalog.

To create a matrix in the matrix catalog

1. Press SHIFT *MATRIX* to open the Matrix catalog. The Matrix catalog lists the 10 available matrix variables, M0 through M9.
2. Highlight the matrix variable name you want to use and press **NEW**. (The dimensions change automatically after you define the matrix.)
3. Select the type of matrix to create.

- **For a vector (one-dimensional array)**, select `Real vector` or `Complex vector`. Certain operations (+, -, `CROSS`) do not recognize a one-dimensional matrix as a vector, so this selection is important.

- **For a matrix (two-dimensional array),** select `Real matrix` or `Complex matrix`.

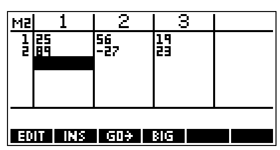
4. For each element in the matrix, type a number or an expression, and press ENTER. (The expression may not contain symbolic variable names.)

- **For complex numbers**, enter each number in complex form; that is, (*a* +*bi*), where *a* is the real part and *b* is the imaginary part. You must include the parentheses and the comma.

5. Use the cursor keys to move to a different row or column. You can change the direction of the highlight bar by pressing **GO.** The **GO** menu key toggles between the following three options:

- **GO**↓ specifies that the cursor moves to the cell below the current cell when you press ENTER.
- **GO**→ specifies that the cursor moves to the cell to the right of the current cell when you press ENTER.
- **GO** specifies that the cursor stays in the current cell when you press ENTER.

6. When done, press SHIFT *MATRIX* to see the Matrix catalog, or press HOME to return to HOME. The matrix entries are automatically stored.



A matrix is listed with two dimensions, even if it is 3×1. A vector is listed with the number of elements, such as 3.

To transmit a matrix

You can send and receive matrices between calculators just as you can send and receive aplets, programs, lists, and notes.

After aligning the HP 39G calculators' infrared ports, open the Matrix catalogs on both calculators. On the sending calculator, highlight the matrix to send, press **SEND**, then and press **RECV** on the receiving calculator.

Matrices can also be transmitted to or from a computer using the separately supplied cable and Connectivity Kit. *Note: The HP 40G is supplied with a PC adapter and unit to unit cable.*

Working with matrix objects

To edit a matrix

In the Matrix catalog, highlight the matrix name you want to edit and press **EDIT** instead of **NEW**.

Matrix edit keys

The following table summarizes the matrix edit key operations.

Key	Meaning
EDIT	Copies the highlighted element into the edit line.
INS	Inserts a row of zeros above, or a column of zeros to the left, of the highlighted cell. (You are prompted to choose row or column.)
GO	A three-way toggle for cursor advancement in the Matrix editor. **GO**→advances to the right, **GO**↓¸ advances downward, and **GO** does not advance at all.
BIG	Switches between larger and smaller font sizes.
DEL	Deletes the highlighted cells row or column (you are prompted to make a choice).
SHIFT *CLEAR*	Clears all elements from the matrix.
SHIFT *cursor key*	Moves to the first or last row or column.

To display a matrix

- In the Matrix catalog (SHIFT *MATRIX*), highlight the matrix name and press **EDIT**.
- In HOME, enter the name of the matrix variable and press ENTER.

To display one element

In HOME, enter *matrixname*(*row*,*column*). For example, if M2 is [[3,4],[5,6]], then M2(1,2) ENTER returns 4.

To delete a matrix

In the Matrix catalog, highlight the matrix to delete and press DEL. The name remains. The matrix is redimensioned to 1×1 with a zero element.

To delete all matrices

In the Matrix catalog, press SHIFT *CLEAR*.

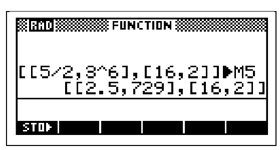
To delete a row or column

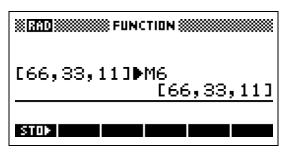
Pressing DEL clears one row or column in a matrix. (You are prompted to choose.) *Note: This operation works in edit mode only. The others work in the catalog view.*

To create a matrix in HOME

1. Enter the matrix in the edit line. Start and end the matrix *and each row* with square brackets (the shifted 5 and 6 keys).
2. Separate each element *and each row* with a comma. Example: [[1,2],[3,4]].
3. Press ENTER to enter and display the matrix.

The left screen below shows the matrix [[2.5,729],[16,2]] being stored into M5. The screen on the right shows the vector [66,33,11] being stored into M6. Note that you can enter an expression (like 5/2) for an element of the matrix, and it will be evaluated.

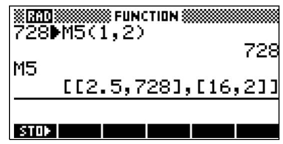




To store one element

In HOME, enter :
value **STO▶***matrixname*(*row*,*column*)
For example, to change the element in the first row and second column of M5 to 728, then display the resulting matrix:

728 **STO▶**
[ALPHA] M5 [(] 1 [,] 2 [)]
[ENTER] [ALPHA] M5
[ENTER].



An attempt to **STO▶** to an element beyond the size of the matrix, as it is currently defined, results in an error message.

Matrix arithmetic

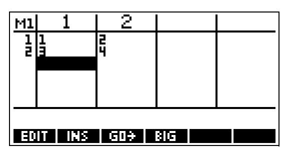
You can use the arithmetic functions (+, –, ×, /) with matrix arguments. Division left multiplies by the inverse of the divisor. You can enter the matrices themselves or enter the names of stored matrix variables. The matrices can be real or complex.

For the next four examples, store [[1,2],[3,4]] into M1 and [[5,6],[7,8]] into M2.

Example

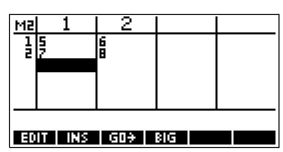
1. Create the first matrix.

[SHIFT] *MATRIX* **NEW** **OK**
1 [ENTER] 2 [ENTER] [▼]
3 [ENTER] 4 [ENTER]



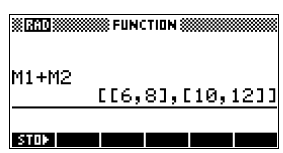
2. Create the second matrix.

[SHIFT] *MATRIX* [▼] **NEW**
OK 5 [ENTER] 6 [ENTER]
[▼] 7 [ENTER] 8 [ENTER]



3. Add the matrices that you created.

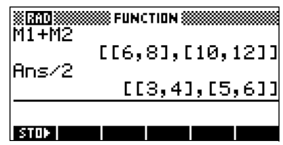
[HOME] [ALPHA] M 1 [+]
[ALPHA] M 2
STO▶ [ENTER]



To multiply and divide by a scalar

For division by a scalar, enter the matrix first, then the operator, then the scalar. For multiplication, the order of the operands does not matter. The matrix and the scalar can be real or complex. For example, to divide the result from the previous example by 2, use the following key presses:

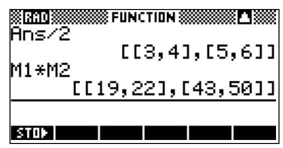
[/] 2 [ENTER]



To multiply two matrices

To multiply the two matrices M1 and M2 that you created for the previous example, use the following keystrokes:

[ALPHA]M 1 [*] [ALPHA]M 2 [ENTER]



To multiply a matrix by a vector, enter the matrix first, then the vector. The number of elements in the vector must equal the number of columns in the matrix.

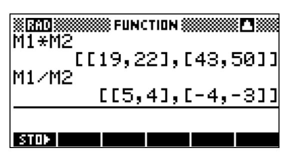
To divide by a square matrix

For division of a matrix or a vector by a square matrix, the number of rows of the dividend (or the number of elements, if it is a vector) must equal the number of rows in the divisor.

This operation is not a mathematical division, it is a left multiplication by the inverse of the divisor. M1/M2 is equivalent to $M2^{-1} * M1$.

To divide the two matrices M1 and M2 that you created for the previous example, use the following keystrokes:

[ALPHA]M 1 [/] [ALPHA]M 2 [ENTER]



To invert a matrix

You can invert a *square matrix* in HOME by typing the matrix (or its variable name) and pressing [SHIFT]x^{-1} [ENTER]. Or you can use the matrix INVERSE command. Enter INVERSE(*matrixname*) in HOME.

To negate each element

You can change the sign of each element in a matrix by pressing [(-)] before the matrix name.

Solving systems of linear equations

Example

Solve the following linear system:

$$\begin{aligned} 2x + 3y + 4z &= 5 \\ x + y - z &= 7 \\ 4x - y + 2z &= 1 \end{aligned}$$

1. Open the Matrix catalog and create a new vector object in the M1 variable. The Create New choose list is displayed.

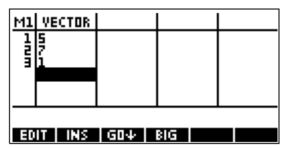
   [SHIFT] *MATRIX* **NEW**

   [▼] [ENTER]

2. Create the vector of the constants.

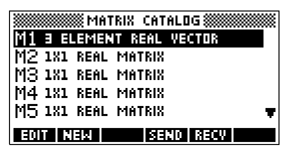
   5 [ENTER] 7 [ENTER]
   1 [ENTER]

   .



3. Return to the Matrix catalog. The vector you created is listed as M1.

   [SHIFT] *MATRIX*



4. Select the M2 variable and create a new matrix.

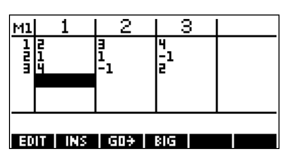
   [▼] **NEW**

   Select `Real matrix`

   OK

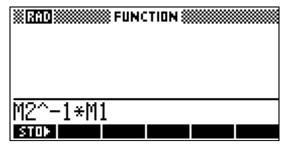
5. Create a new matrix and enter the constants.

   2 [ENTER] 3
   [ENTER] 4 [ENTER] [▼] 1
   [ENTER] 1 [ENTER]
   [(-)] 1 [ENTER]
   4 [ENTER]
   [(-)] 1 [ENTER] 2
   [ENTER]



6. Return to HOME and enter the calculation to left multiply the constants vector by the inverse of the coefficients matrix.

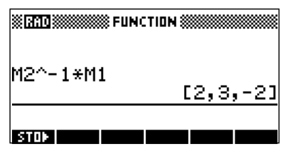
   HOME ALPHA M 2
   SHIFT x^{-1} *
   ALPHA M 1



   An alternative method, is to use the RREF function. See “RREF” on page 12-179.

7. Evaluate the calculation.

   ENTER



The result is a vector of the solutions. That is, from the vector:

- x=2
- y=3
- z=-2

Matrix functions and commands

About functions

- Functions can be used in any aplet or in HOME. They are listed in the MATH menu under the Matrix category. They can be used in mathematical expressions—primarily in HOME, as well as in programs.
- Functions always produce *and display* a result. They *do not change any stored variables*, such as a matrix variable.
- Functions have arguments that are enclosed in parentheses and separated by commas. Example: CROSS(*vector1*,*vector2*). The matrix input can be either a matrix variable name (such as M1) or the actual matrix data inside brackets. For example, CROSS(M1,[1,2]).

About commands

Matrix commands are listed in the CMDS menu (SHIFT *CMDS*), category `Matrix`.

See “Matrix commands” on page 15-220 for details on the matrix commands available for use in programming.

Argument Conventions

- For *row#* or *column#*, supply the number of the row (counting from the top, starting with 1) or the number of the column (counting from the left, starting with 1).
- The argument *matrix* can refer to either a vector or a matrix.

Matrix functions

The matrix functions are given in the following table.

See “Matrix commands” on page 15-220 for details on the matrix commands available for use in programming.

COLNORM

Column Norm. Finds the maximum value (over all columns) of the sums of the absolute values of all elements in a column.

COLNORM(*matrix*)

COND

Condition Number. Finds the 1-norm (column norm) of a square *matrix.*

COND(*matrix*)

CROSS

Cross Product of *vector1* with *vector2*.

CROSS(*vector1*, *vector2*)

DET

Determinant of a square *matrix*.

DET(*matrix*)

DOT

Dot Product of two arrays, *matrix1 matrix2*.

DOT(*matrix1*, *matrix2*)

EIGENVAL

Displays the eigenvalues in vector form for *matrix*.

EIGENVAL(*matrix*)

EIGENVV

Eigenvectors and Eigenvalues for a square *matrix*. Displays a list of two arrays. The first contains the eigenvectors and the second contains the eigenvalues.

EIGENVV(*matrix*)

IDENMAT

Identity matrix. Creates a square matrix of dimension *size* × *size* whose diagonal elements equal 1 and off-diagonal elements equal zero.

IDENMAT(*size*)

INVERSE

Inverts a square matrix (real or complex).

INVERSE(*matrix*)

LQ

LQ Factorization. Factors *m* × *n matrix* into three matrices: {[[*m* × *n lowertrapezoidal*]],[[*n* × *n orthogonal*]], [[*m* × *m permutation*]]}.

LQ(*matrix*)

LSQ

Least Squares. Displays the minimum norm least squares matrix (or vector).

LSQ(*matrix1*, *matrix2*)

LU

LU Decomposition. Factors a square *matrix* into three matrices:
{[[*lowertriangular*]],[[*uppertriangular*]],[[*permutation*]]}
The *uppertriangular* has ones on its diagonal.

LU(*matrix*)

MAKEMAT

Make Matrix. Creates a matrix of dimension *rows* × *columns*, using *expression* to calculate each element. If *expression* contains the variables I and J, then the calculation for each element substitutes the current row number for I and the current column number for J.

MAKEMAT(*expression*, *rows*, *columns*)

Example

MAKEMAT(0,3,3) returns a 3×3 zero matrix, [[0,0,0],[0,0,0],[0,0,0]].

The `EDITMAT` command can also be used to create matrices.

1. Press [SHIFT] *CMDS* [)] [▶] [SIN] **OK**
2. Press [ALPHA] *M* 1, and then press [ENTER].

3. The Matrix catalog opens with M1 available for editing.

   `EDITMAT` *matrixname* is a shortcut to opening the matrix editor with *matrixname*.

QR

QR Factorization. Factors an $m \times n$ *matrix* into three matrices: {[[$m \times m$ *orthogonal*]],[[$m \times n$ *uppertrapezoidal*]],[[$n \times n$ *permutation*]]}.

QR(*matrix*)

RANK

Rank (an integer) of a rectangular *matrix*.

RANK(*matrix*)

ROWNORM

Row Norm. Finds the maximum value (over all rows) of the sums of the absolute values of all elements in a row.

ROWNORM(*matrix*)

RREF

Reduced Row-Echelon. Changes a rectangular *matrix* to its reduced row-echelon form.

RREF(*matrix*)

This function takes an augmented matrix of size n by n+1 and transforms it into reduced row echelon form, with the final column containing the solution.

SCHUR

Schur Decomposition. Factors a square *matrix* into two matrices. If *matrix* is real, then the result is {[[*orthogonal*]],[[*upper-quasi triangular*]]}.
If *matrix* is complex, then the result is {[[*unitary*]],[[*upper-triangular*]]}.

SCHUR(*matrix*)

SIZE

Dimensions of *matrix*. Returned as a list: {rows,columns}.

SIZE(*matrix*)

SPECNORM

Spectral Norm of the specified array.

SPECNORM(*matrix*)

SPECRAD

Spectral Radius of a square *matrix*.

SPECRAD(*matrix*)

SVD

Singular Value Decomposition. Factors an $m \times n$ *matrix* into two matrices and a vector :

{[[$m \times m$ *square orthogonal*]],[[$n \times n$ *square orthogonal*]], [*real*]}.

SVD(*matrix*)

SVL

Singular Values. Returns a vector containing the singular values of *matrix.*

SVL(*matrix*)

TRACE

Finds the trace of a square *matrix.* The trace is equal to the sum of the diagonal elements. (It is also equal to the sum of the eigenvalues.)

TRACE(*matrix*)

TRN

Transposes *matrix.* For a complex matrix, TRN finds the conjugate transpose.

TRN(*matrix*)

Examples

Filling Matrices

You can create and fill a matrix with 0 for each diagonal element and 1 for each off-diagonal element using the MAKEMAT (*make matrix*) function. For example, entering MAKEMAT(I≠J,4,4) creates a 4 × 4 matrix showing the numeral 1 for all elements except zeros on the diagonal. The logical operator ≠ returns 0 when I (the row number) and J (the column number) are equal, and returns 1 when they are not equal.

Identity Matrix

You can create an identity matrix with the IDENMAT function. For example, IDENMAT(2) creates the 2×2 identity matrix [[1,0],[0,1]].

Transposing a Matrix

The TRN function swaps the row-column and column-row elements of a matrix. For instance, element 1,2 (row 1, column 2) is swapped with element 2,1; element 2,3 is swapped with element 3,2; and so on.

For example, TRN([[1,2],[3,4]]) creates the matrix [[1,3],[2,4]].

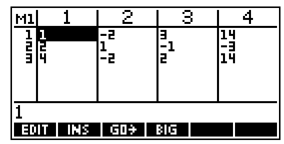
RREF example

The following set of equations

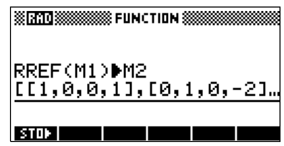
$$x - 2y + 3z = 14$$
$$2x + y - z = -3$$
$$4x - 2y + 2z = 14$$

is written as the augmented matrix $\left[\begin{array}{ccc|c} 1 & -2 & 3 & 14 \\ 2 & 1 & -1 & -3 \\ 4 & -2 & 2 & 14 \end{array}\right]$

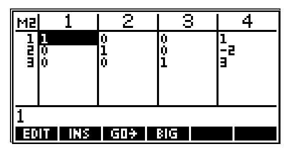
which is then stored as a 3 **x** 4 real matrix M1.



We now use the function RREF to change this to reduced row echelon form, storing it as M2 simply for convenience.



This gives the final result shown in the matrix M2 on the right, giving a solution of (1, -2, 3).



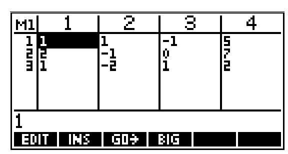
An advantage of using the RREF function is that it will also work with inconsistent matrices resulting from systems of equations which have no solution or infinite solutions.

RREF example

The following set of equations

$$\begin{aligned} x + y - z &= 5 \\ 2x - y &= 7 \\ x - 2y + z &= 2 \end{aligned}$$

has an infinite number of solutions.



The final row of zeros indicates an inconsistency.