Chapter 5
Algebraic and arithmetic operations

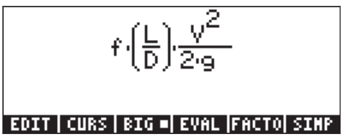
An algebraic object, or simply, algebraic, is any number, variable name or algebraic expression that can be operated upon, manipulated, and combined according to the rules of algebra. Examples of algebraic objects are the following:

- A number: 12.3, 15.2_m, 'π', 'e', 'i'
- A variable name: 'a', 'ux', 'width', etc.
- An expression: 'p*D^2/4','f*(L/D)*(V^2/(2*g))'
- An equation: 'Q=(Cu/n)*A(y)*R(y)^(2/3)*So^0.5'

Entering algebraic objects

Algebraic objects can be created by typing the object between single quotes directly into stack level 1 or by using the equation writer ⇀ EQW . For example, to enter the algebraic object 'π*D^2/4' directly into stack level 1 use: ' ← π × ALPHA D y^x 2 ÷ 4 ENTER . The resulting screen is shown next for both the ALG mode (left-hand side) and the RPN mode (right-hand side):

An algebraic object can also be built in the Equation Writer and then sent to the stack. The operation of the Equation Writer was described in Chapter 2. As an exercise, build the following algebraic object in the Equation Writer:



After building the object, press to show it in the stack (ALG and RPN modes shown below):

Simple operations with algebraic objects

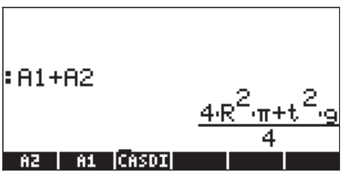
Algebraic objects can be added, subtracted, multiplied, divided (except by zero), raised to a power, used as arguments for a variety of standard functions (exponential, logarithmic, trigonometry, hyperbolic, etc.), as you would any real or complex number. To demonstrate basic operations with algebraic objects, let's create a couple of objects, say 'π*R^2' and 'g*t^2/4', and store them in variables A1 and A2 (See Chapter 2 to learn how to create variables and store values in them). Here are the keystrokes for storing variables A1 in ALG mode: ['] [←] π [×] [ALPHA] [R] [yˣ] [2] [▶] [STO▶] [ALPHA] [A] [1] [ENTER], resulting in:

The keystrokes corresponding to RPN mode are: ['] [←] π [×] [ALPHA] [R] [yˣ] [2] [ENTER] [ALPHA] [A] [1] [STO▶]

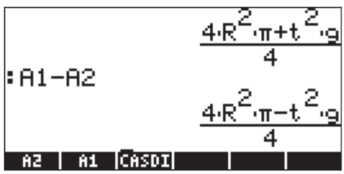
After storing the variable A2 and pressing the key, the screen will show the variables as follows:

In ALG mode, the following keystrokes will show a number of operations with the algebraics contained in variables [A1] and [A2] (press [VAR] to recover variable menu):

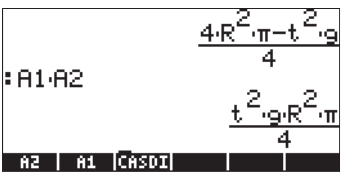
[A1] [+] [A2] [ENTER]



[A1] [−] [A2] [ENTER]

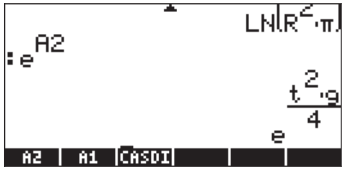


[A1] [×] [A2] [ENTER]



[A1] [÷] [A2] [ENTER]

RightShift LN **A1** LeftShift e^x **A2**



The same results are obtained in RPN mode if using the following keystrokes:

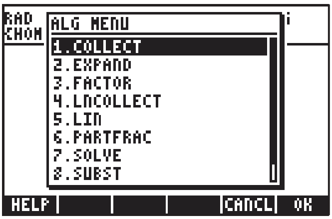
A1 **A2** + EVAL **A1** **A2** − EVAL

A1 **A2** × EVAL **A1** **A2** ÷ EVAL

A1 RightShift LN EVAL **A2** LeftShift e^x EVAL

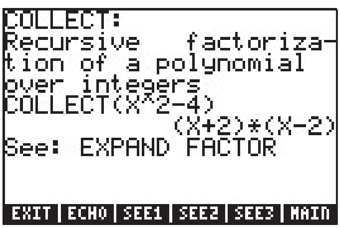
Functions in the ALG menu

The ALG (Algebraic) menu is available by using the keystroke sequence RightShift ALG (associated with the 4 key). With system flag 117 set to *CHOOSE boxes*, the ALG menu shows the following functions:



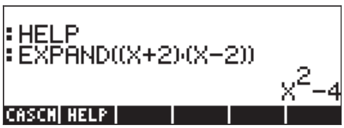
Rather than listing the description of each function in this manual, the user is invited to look up the description using the calculator's help facility: TOOL NXT **HELP** ENTER . To locate a particular function, type the first letter of the function. For example, for function COLLECT, we type ALPHA C , then use the up and down arrow keys, ▲ ▼ , to locate COLLECT within the help window.

To complete the operation press **OK**. Here is the help screen for function COLLECT:



We notice that, at the bottom of the screen, the line See: EXPAND FACTOR suggests links to other help facility entries, the functions EXPAND and FACTOR. To move directly to those entries, press the soft menu key SEE1 for EXPAND, and SEE2 for FACTOR. Pressing SEE1, for example, shows the following information for EXPAND:

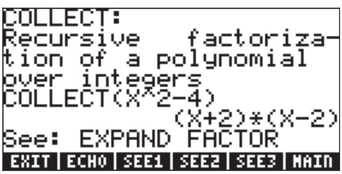
Help facility
A help facility accessible via TOOL NEXT CASCMD allows you to browse through all the CAS commands. It provides not only information on each command, but also provides an example of its application. This example can be copied onto your stack by pressing the ECHO soft menu key. For example, for the EXPAND entry shown above, press the ECHO soft menu key to get the following example copied to the stack (press ENTER to execute the command):



We leave for the user to explore the list of CAS functions available. Here are a couple of examples:

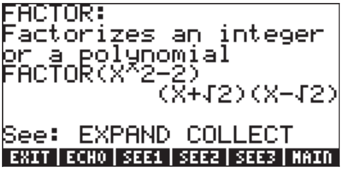
The help facility will show the following information on the commands:

COLLECT:

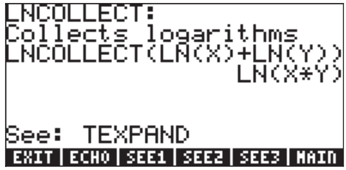


EXPAND:

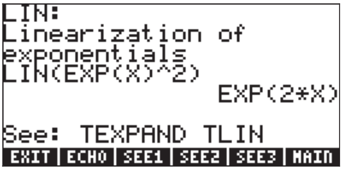
FACTOR:



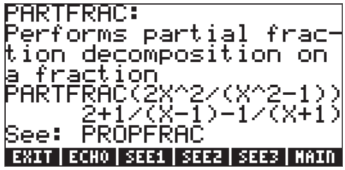
LNCOLLECT:



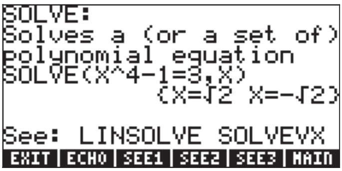
LIN:



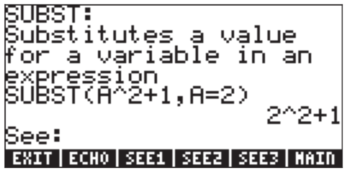
PARTFRAC:



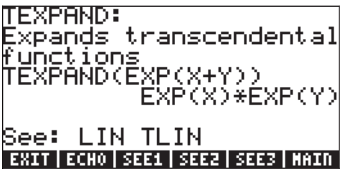
SOLVE:



SUBST:



TEXPAND:



Note: Recall that, to use these, or any other functions in the RPN mode, you must enter the argument first, and then the function. For example, the example for TEXPAND, in RPN mode will be set up as:

['] [←] e^x [+] [ALPHA] [X] [+] [ALPHA] [Y] [ENTER]

At this point, select function TEXPAND from menu ALG (or directly from the catalog [→] CAT), to complete the operation.

Other forms of substitution in algebraic expressions

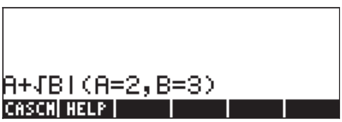
Functions SUBST, shown above, is used to substitute a variable in an expression. A second form of substitution can be accomplished by using the [→] [|] (associated with the I key). For example, in ALG mode, the following entry will substitute the value *x = 2* in the expression $x+x^2$. The figure to the left shows the way to enter the expression (the substituted value, x=2, must be enclosed in parentheses) before pressing [ENTER]. After the [ENTER] key is pressed, the result is shown in the right-hand figure:

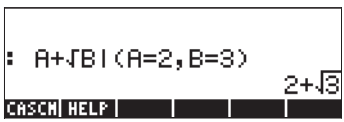
In RPN mode, this can be accomplished by entering first the expression where the substitution will be performed ($x+x^2$), followed by a list (see Chapter 8) containing the substitution variable, an space, and the value to be substituted, i.e., {x 2}. The final step is to press the keystroke combination: [→] [|] .

The required keystrokes are the following:

['] [ALPHA] [←] [X] [+] [ALPHA] [←] [X] [y^x] [2] [ENTER]
[←] [{}] [ALPHA] [←] [X] [SPC] [2] [ENTER] [→] [|] [ENTER]

In ALG mode, substitution of more than one variable is possible as illustrated in the following example (shown before and after pressing [ENTER])

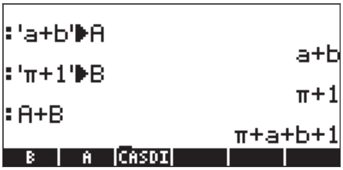




In RPN mode, it is also possible to substitute more than one variable at a time, as illustrated in the example below. Recall that RPN mode uses a list of variable names and values for substitution.

A different approach to substitution consists in defining the substitution expressions in calculator variables and placing the name of the variables in the original expression. For example, in ALG mode, store the following variables:

Then, enter the expression A+B:



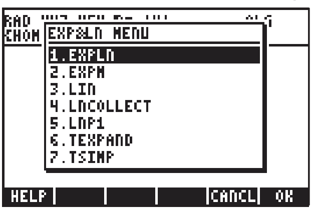
The last expression entered is automatically evaluated after pressing the ENTER key, producing the result shown above.

Operations with transcendental functions

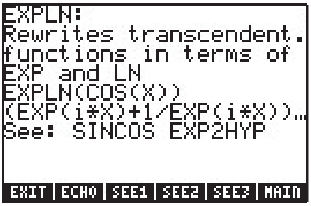
The calculator offers a number of functions that can be used to replace expressions containing logarithmic, exponential, trigonometric, and hyperbolic functions in terms of trigonometric identities or in terms of exponential functions. The menus containing functions to replace trigonometric functions can be obtained directly from the keyboard by pressing the right-shift key followed by the 8 key, i.e., ↱ TRIG . The combination of this key with the left-shift key, i.e., ↱ EXP&LN , produces a menu that lets you replace expressions in terms of exponential or natural logarithm functions. In the next sections we cover those menus in more detail.

Expansion and factoring using log-exp functions

The ↰ EXP&LN produces the following menu:

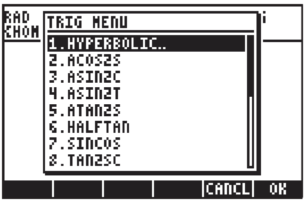


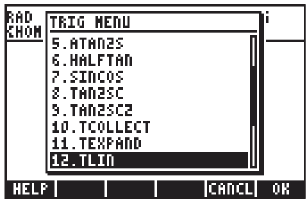
Information and examples on these commands are available in the help facility of the calculator. Some of the command listed in the EXP&LN menu, i.e., LIN, LNCOLLECT, and TEXPAND are also contained in the ALG menu presented earlier. Functions LNP1 and EXPM were introduced in menu HYPERBOLIC, under the MTH menu (See Chapter 2). The only remaining function is EXPLN. Its description is shown in the left-hand side, the example from the help facility is shown to the right:



Expansion and factoring using trigonometric functions

The TRIG menu, triggered by using ⇁ TRIG, shows the following functions:





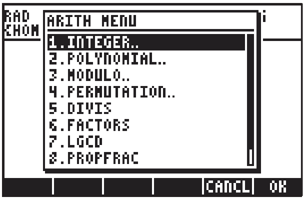
These functions allow to simplify expressions by replacing some category of trigonometric functions for another one. For example, the function ACOS2S allows to replace the function *arccosine* (acos(x)) with its expression in terms of *arcsine* (asin(x)).

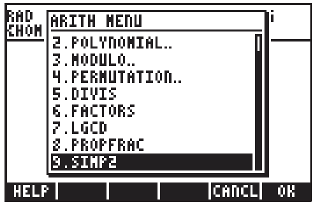
Description of these commands and examples of their applications are available in the calculator's help facility (TOOL NXT HELP). The user is invited to explore this facility to find information on the commands in the TRIG menu.

Notice that the first command in the TRIG menu is the HYPERBOLIC menu, whose functions were introduced in Chapter 2.

Functions in the ARITHMETIC menu

The ARITHMETIC menu contains a number of sub-menus for specific applications in number theory (integers, polynomials, etc.), as well as a number of functions that apply to general arithmetic operations. The ARITHMETIC menu is triggered through the keystroke combination ← ARITH (associated with the / key). With system flag 117 set to *CHOOSE boxes*, ← ARITH shows the following menu:

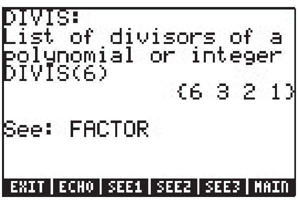




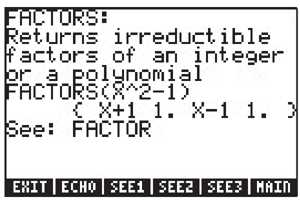
Out of this menu list, options 5 through 9 (*DIVIS, FACTORS, LGCD, PROPFRAC, SIMP2*) correspond to common functions that apply to integer numbers or to polynomials. The remaining options (*1. INTEGER*, *2. POLYNOMIAL*, *3. MODULO*, and *4. PERMUTATION*) are actually sub-menus of functions that apply to specific mathematical objects. This distinction between sub-menus (options 1 through 4) and plain functions (options 5 through 9) is made clear when system flag 117 is set to *SOFT menus*. Activating the ARITHMETIC menu (← ARITH), under these circumstances, produces:

Following, we present the help facility entries for the functions of options 5 through 9 in the ARITHMETIC menu (TOOL NXT HELP):

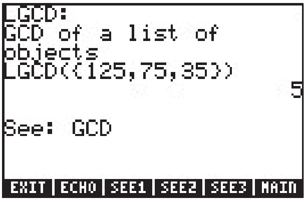
DIVIS:



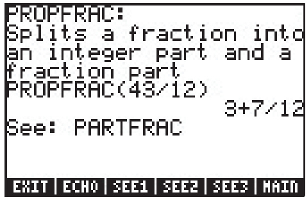
FACTORS:



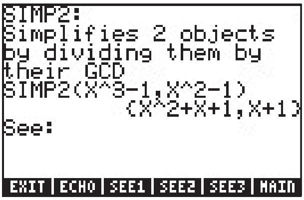
LGCD (Greatest Common Denominator):



PROPFRAC (proper fraction)



SIMP2:



The functions associated with the ARITHMETIC submenus: INTEGER, POLYNOMIAL, MODULO, and PERMUTATION, are the following:

INTEGER menu

EULER	Number of integers < n, co -prime with n
IABCUV	Solves au + bv = c, with a,b,c = integers
IBERNOULLI	n-th Bernoulli number
ICHINREM	Chinese reminder for integers
IDIV2	Euclidean division of two integers
IEGCD	Returns u,v, such that au + bv = gcd(a,b)
IQUOT	Euclidean quotient of two integers
IREMAINDER	Euclidean remainder of two integers
ISPRIME?	Test if an integer number is prime
NEXTPRIME	Next prime for a given integer number
PA2B2	Prime number as square norm of a complex number
PREVPRIME	Previous prime for a given integer number

POLYNOMIAL menu

ABCUV	Bézout polynomial equation (au+bv=c)
CHINREM	Chinese remainder for polynomials
CYCLOTOMIC	n-th cyclotomic polynomial
DIV2	Euclidean division of two polynomials
EGDC	Returns u,v, from au+bv=gcd(a,b)

FACTOR	Factorizes an integer number or a polynomial
FCOEF	Generates fraction given roots and multiplicity
FROOTS	Returns roots and multiplicity given a fraction
GCD	Greatest common divisor of 2 numbers or polynomials
HERMITE	n-th degree Hermite polynomial
HORNER	Horner evaluation of a polynomial
LAGRANGE	Lagrange polynomial interpolation
LCM	Lowest common multiple of 2 numbers or polynomials
LEGENDRE	n-th degree Legendre polynomial
PARTFRAC	Partial-fraction decomposition of a given fraction
PCOEF	(help-facility entry missing)
PTAYL	Returns Q(x-a) in Q(x-a) = P(x), Taylor polynomial
QUOT	Euclidean quotient of two polynomials
RESULTANT	Determinant of the Sylvester matrix of 2 polynomials
REMAINDER	Euclidean reminder of two polynomials
STURM	Sturm sequences for a polynomial
STURMAB	Sign at low bound and number of zeros between bounds

MODULO menu

ADDTMOD	Add two expressions modulo current modulus
DIVMOD	Divides 2 polynomials modulo current modulus
DIV2MOD	Euclidean division of 2 polynomials with modular coefficients
EXPANDMOD	Expands/simplify polynomial modulo current modulus
FACTORMOD	Factorize a polynomial modulo current modulus
GCDMOD	GCD of 2 polynomials modulo current modulus
INVMOD	inverse of integer modulo current modulus
MOD	(not entry available in the help facility)
MODSTO	Changes modulo setting to specified value
MULTMOD	Multiplication of two polynomials modulo current modulus
POWMOD	Raises polynomial to a power modulo current modulus
SUBTMOD	Subtraction of 2 polynomials modulo current modulus

Applications of the ARITHMETIC menu

This section is intended to present some of the background necessary for application of the ARITHMETIC menu functions. Definitions are presented next regarding the subjects of polynomials, polynomial fractions and modular arithmetic. The examples presented below are presented independently of the calculator setting (ALG or RPN)

Modular arithmetic

Consider a counting system of integer numbers that periodically cycles back on itself and starts again, such as the hours in a clock. Such counting system is called a *ring*. Because the number of integers used in a ring is finite, the arithmetic in this ring is called *finite arithmetic*. Let our system of finite integer numbers consists of the numbers *0, 1, 2, 3, …, n-1, n*. We can also refer to the arithmetic of this counting system as *modular arithmetic of modulus n*. In the case of the hours of a clock, the modulus is 12. (If working with modular arithmetic using the hours in a clock, however, we would have to use the integer numbers 0, 1, 2, 3, …, 10, 11, rather than 1, 2, 3,…,11, 12).

Operations in modular arithmetic

Addition in modular arithmetic of modulus *n,* which is a positive integer, follow the rules that if *j* and *k* are any two nonnegative integer numbers, both smaller than *n*, if $j+k\geq n$, then $j+k$ is defined as $j+k-n$. For example, in the case of the clock, i.e., for $n = 12$, 6+9 "=" 3. To distinguish this 'equality' from infinite arithmetic equalities, the symbol ≡ is used in place of the equal sign, and the relationship between the numbers is referred to as a *congruence* rather than an equality. Thus, for the previous example we would write *6+9 ≡ 3 (mod 12),* and read this expression as "*six plus nine is congruent to three, modulus twelve*." If the numbers represent the hours since midnight, for example, the congruence 6+9 ≡ 3 (mod 12), can be interpreted as saying that "six hours past the ninth hour after midnight will be three hours past noon." Other sums that can be defined in modulus 12 arithmetic are: 2+5 ≡ 7 (mod 12); 2+10 ≡ 0 (mod 12); 7+5 ≡ 0 (mod 12); etcetera.

The rule for *subtraction* will be such that if $j - k < 0$, then j-k is defined as j-k+n. Therefore, *8-10 ≡ 2 (mod 12),* is read "*eight minus ten is congruent to two, modulus twelve*." Other examples of subtraction in modulus 12 arithmetic would be 10-5 ≡ 5 (mod 12); 6-9 ≡ 9 (mod 12); 5 – 8 ≡ 9 (mod 12); 5 –10 ≡ 7 (mod 12); etcetera.

Multiplication follows the rule that if $j\cdot k > n$, so that $j\cdot k = m\cdot n + r$, where *m* and *r* are nonnegative integers, both less than *n*, then $j\cdot k \equiv r$ (mod *n*). The result of multiplying *j* times *k* in modulus *n* arithmetic is, in essence, the integer remainder of $j \cdot k/n$ in infinite arithmetic, if $j \cdot k>n$. For example, in modulus 12 arithmetic we have $7 \cdot 3 = 21 = 12 + 9$, (or, $7 \cdot 3/12 = 21/12 = 1 + 9/12$, i.e., the integer reminder of 21/12 is 9). We can now write $7 \cdot 3 \equiv 9 \pmod{12}$, and read the latter result as "*seven times three is congruent to nine, modulus twelve*."

The operation of *division* can be defined in terms of multiplication as follows, $r/k \equiv j \pmod{n}$, if, $j \cdot k \equiv r \pmod{n}$. This means that *r* must be the remainder of $j \cdot k/n$. For example, $9/7 \equiv 3 \pmod{12}$, because $7 \cdot 3 \equiv 9 \pmod{12}$. Some divisions are not permitted in modular arithmetic. For example, in modulus 12 arithmetic you cannot define 5/6 (mod 12) because the multiplication table of 6 does not show the result 5 in modulus 12 arithmetic. This multiplication table is shown below:

6*0 (mod 12)	0
6*1 (mod 12)	6
6*2 (mod 12)	0
6*3 (mod 12)	6
6*4 (mod 12)	0
6*5 (mod 12)	6

6*6 (mod 12)	0
6*7 (mod 12)	6
6*8 (mod 12)	0
6*9 (mod 12)	6
6*10 (mod 12)	0
6*11 (mod 12)	6

Formal definition of a finite arithmetic ring

The expression $a \equiv b \pmod{n}$ is interpreted as "a is congruent to b, modulo n," and holds if $(b-a)$ is a multiple of *n*. With this definition the rules of arithmetic simplify to the following:

If $a \equiv b \pmod{n}$ and $c \equiv d \pmod{n}$,

then

$$a+c \equiv b+d \pmod{n},$$
$$a-c \equiv b - d \pmod{n},$$
$$a\times c \equiv b\times d \pmod{n}.$$

For division, follow the rules presented earlier. For example, $17 \equiv 5 \pmod{6}$, and $21 \equiv 3 \pmod{6}$. Using these rules, we can write:

$17 + 21 \equiv 5 + 3 \pmod{6} \Rightarrow 38 \equiv 8 \pmod{6} \Rightarrow 38 \equiv 2 \pmod{6}$

$17 - 21 \equiv 5 - 3 \pmod{6} \Rightarrow -4 \equiv 2 \pmod{6}$

$17 \times 21 \equiv 5 \times 3 \pmod{6} \Rightarrow 357 \equiv 15 \pmod{6} \Rightarrow 357 \equiv 3 \pmod{6}$

Notice that, whenever a result in the right-hand side of the "congruence" symbol produces a result that is larger than the modulo (in this case, n = 6), you can always subtract a multiple of the modulo from that result and simplify it to a number smaller than the modulo. Thus, the results in the first case *8 (mod 6)* simplifies to *2 (mod 6*), and the result of the third case, *15 (mod 6)* simplifies to *3 (mod 6)*. Confusing? Well, not if you let the calculator handle those operations. Thus, read the following section to understand how finite arithmetic rings are operated upon in your calculator.

Finite arithmetic rings in the calculator

All along we have defined our finite arithmetic operation so that the results are always positive. The modular arithmetic system in the calculator is set so that the ring of modulus n includes the numbers *-n/2+1, …,-1, 0, 1,…,n/2-1, n/2*, if n is even, and *–(n-1)/2, -(n-3)/2,…,-1,0,1,…,(n-3)/2, (n-1)/2*, if n is odd. For example, for n = 8 (even), the finite arithmetic ring in the calculator includes the numbers: (-3,-2,-1,0,1,3,4), while for n = 7 (odd), the corresponding calculator's finite arithmetic ring is given by (-3,-2,-1,0,1,2,3).

Modular arithmetic in the calculator

To launch the modular arithmetic menu in the calculator select the MODULO sub-menu within the ARITHMETIC menu (← ARITH). The available menu includes functions: ADDTMOD, DIVMOD, DIV2MOD, EXPANDMOD, FACTORMOD, GCDMOD, INVMOD, MOD, MODSTO, MULTMOD, POWMOD, and SUBTMOD. Brief descriptions of these functions were provided in an earlier section. Next we present some applications of these functions.

Setting the modulus (or MODULO)

The calculator contains a variable called MODULO that is placed in the {HOME CASDIR} directory and will store the magnitude of the modulus to be used in modular arithmetic.

The default value of MODULO is 13. To change the value of MODULO, you can either store the new value directly in the variable MODULO in the sub-directory {HOME CASDIR} Alternatively, you can store a new MODULO value by using function MODSTO.

Modular arithmetic operations with numbers

To add, subtract, multiply, divide, and raise to a power using modular arithmetic you will use the functions ADDTMOD, SUBTMOD, MULTMOD, DIV2MOD and DIVMOD (for division), and POWMOD. In RPN mode you need to enter the two numbers to operate upon, separated by an [ENTER] or an

[SPC] entry, and then press the corresponding modular arithmetic function. For example, using a modulus of 12, try the following operations:

ADDTMOD examples

6+5 ≡ -1 (mod 12)	6+6 ≡ 0 (mod 12)	6+7 ≡ 1 (mod 12)
11+5 ≡ 4 (mod 12)	8+10 ≡ -6 (mod 12)	

SUBTMOD examples

5 - 7 ≡ -2 (mod 12)	8 – 4 ≡ 4 (mod 12)	5 –10 ≡ -5 (mod 12)
11 – 8 ≡ 3 (mod 12)	8 - 12 ≡ -4 (mod 12)	

MULTMOD examples

6·8 ≡ 0 (mod 12)	9·8 ≡ 0 (mod 12)	3·2 ≡ 6 (mod 12)
5·6 ≡ 6 (mod 12)	11·3 ≡ -3 (mod 12)	

DIVMOD examples

12/3 ≡ 4 (mod 12)	12/8 (mod 12) does not exist
25/5 ≡ 5 (mod 12)	64/13 ≡ 4 (mod 12)
66/6 ≡ -1 (mod 12)	

DIV2MOD examples

2/3 (mod 12) does not exist
26/12 (mod 12) does not exist
125/17 (mod 12) ½ 1 with remainder = 0
68/7 ½ -4 (mod 12) with remainder = 0
7/5 ½ -1 (mod 12) with remainder = 0

Note: DIVMOD provides the quotient of the modular division j/k (mod n), while DIMV2MOD provides no only the quotient but also the remainder of the modular division j/k (mod n).

POWMOD examples

2^3≡ -4 (mod 12)	3^5≡ 3 (mod 12)	5^{10}≡ 1 (mod 12)
11^8 ≡ 1 (mod 12)	6^2 ≡ 0 (mod 12)	9^9 ≡ -3 (mod 12)

In the examples of modular arithmetic operations shown above, we have used numbers that not necessarily belong to the ring, i.e., numbers such as 66, 125, 17, etc. The calculator will convert those numbers to ring numbers before operating on them. You can also convert any number into a ring number by using the function EXPANDMOD. For example,

$$\text{EXPANDMOD}(125) \equiv 5 \pmod{12}$$
$$\text{EXPANDMOD}(17) \equiv 5 \pmod{12}$$
$$\text{EXPANDMOD}(6) \equiv 6 \pmod{12}$$

The modular inverse of a number

Let a number *k* belong to a finite arithmetic ring of modulus *n*, then the modular inverse of *k*, i.e., $1/k$ (mod *n*), is a number *j*, such that $j \cdot k \equiv 1$ (mod *n*). The modular inverse of a number can be obtained by using the function INVMOD in the MODULO sub-menu of the ARITHMETIC menu. For example, in modulus 12 arithmetic:

1/6 (mod 12) does not exist.	$1/5 \equiv 5$ (mod 12)
$1/7 \equiv -5$ (mod 12)	1/3 (mod 12) does not exist.
$1/11 \equiv -1$ (mod 12)	

The MOD operator

The MOD operator is used to obtain the ring number of a given modulus corresponding to a given integer number. On paper this operation is written as *m* mod *n* = *p*, and is read as "*m* modulo *n* is equal to *p*". For example, to calculate 15 mod 8, enter:

- ALG mode: 1 5 MOD 8 ENTER
- RPN mode: 1 5 ENTER 8 ENTER MOD

The result is 7, i.e., 15 mod 8 = 7. Try the following exercises:

18 mod 11 = 7	23 mod 2 =1	40 mod 13 = 1
23 mod 17 = 6	34 mod 6 = 4	

One practical application of the MOD function for programming purposes is to determine when an integer number is odd or even, since *n* mod 2 = 0, if *n* is even, and *n* mode 2 = 1, if *n* is odd. It can also be used to determine when an integer *m* is a multiple of another integer *n*, for if that is the case *m* mod *n* = 0.

Note: Refer to the help facility in the calculator for description and examples on other modular arithmetic. Many of these functions are applicable to polynomials. For information on modular arithmetic with polynomials please refer to a textbook on number theory.

Polynomials

Polynomials are algebraic expressions consisting of one or more terms containing decreasing powers of a given variable. For example, 'X^3+2*X^2-3*X+2' is a third-order polynomial in X, while 'SIN(X)^2-2' is a second-order polynomial in SIN(X). A listing of polynomial-related functions in the ARITHMETIC menu was presented earlier. Some general definitions on polynomials are provided next. In these definitions *A(X), B(X), C(X), P(X), Q(X), U(X), V(X),* etc., are polynomials.

- Polynomial fraction: a fraction whose numerator and denominator are polynomials, say, C(X) = A(X)/B(X)
- Roots, or zeros, of a polynomial: values of X for which P(X) = 0
- Poles of a fraction: roots of the denominator
- Multiplicity of roots or poles: the number of times a root shows up, e.g., P(X) = $(X+1)^2(X-3)$ has roots {-1, 3} with multiplicities {2,1}
- Cyclotomic polynomial ($P_n(X)$): a polynomial of order EULER(*n*) whose roots are the primitive *n*-th roots of unity, e.g., $P_2(X) = X+1$, $P_4(X) = X^2+1$
- Bézout's polynomial equation: A(X) U(X) + B(X)V(X) = C(X)

Specific examples of polynomial applications are provided next.

Modular arithmetic with polynomials

The same way that we defined a finite-arithmetic ring for numbers in a previous section, we can define a finite-arithmetic ring for polynomials with a given polynomial as modulo. For example, we can write a certain polynomial *P(X)* as $P(X) = X \pmod{X^2}$, or another polynomial $Q(X) = X + 1 \pmod{X-2}$.

A polynomial, *P(X)* belongs to a finite arithmetic ring of polynomial modulus *M(X),* if there exists a third polynomial *Q(X),* such that *(P(X) – Q(X))* is a multiple of *M(X)*. We then would write: $P(X) \equiv Q(X) \pmod{M(X)}$. The later expression is interpreted as "*P(X) is congruent to Q(X), modulo M(X)"*.

The CHINREM function

CHINREM stands for CHINese REMainder. The operation coded in this command solves a system of two congruences using the Chinese Remainder Theorem. This command can be used with polynomials, as well as with integer numbers (function ICHINREM). The input consists of two vectors *[expression_1, modulo_1] and [expression_2, modulo_2]*. The output is a vector *containing [expression_3, modulo_3]*, where *modulo_3* is related to the product *(modulo_1)·(modulo_2)*. Example: CHINREM([X+1, X^2-1],[X+1,X^2]) = [X+1,-(X^4-X^2)]

Statement of the Chinese Remainder Theorem for integers

If $m_1, m_2, \ldots, m_r$ are natural numbers every pair of which are relatively prime, and $a_1, a_2, \ldots, a_r$ are any integers, then there is an integer x that simultaneously satisfies the congruences: $x \equiv a_1 \pmod{m_1}$, $x \equiv a_2 \pmod{m_2}$, …, $x \equiv a_r \pmod{m_r}$. Additionally, if $x = a$ is any solution then all other solutions are congruent to a modulo equal to the product $m_1 \cdot m_2 \cdot \ldots m_r$.

The EGCD function

EGCD stands for Extended Greatest Common Divisor. Given two polynomials, A(X) and B(X), function EGCD produces the polynomials C(X), U(X), and V(X), so that C(X) = U(X)*A(X) + V(X)*B(X). For example, for A(X) = X^2+1, B(X) = X^2-1, EGCD(A(X),B(X)) = {2, 1, -1}. i.e., 2 = 1*(X^2+1')-1*(X^2-1). Also, EGCD('X^3-2*X+5','X') = { 5,1,-(X^2-2)}, i.e., 5 = – (X^2-2)*X + 1*(X^3-2*X+5).

The GCD function

The function GCD (Greatest Common Denominator) can be used to obtain the greatest common denominator of two polynomials or of two lists of polynomials of the same length. The two polynomials or lists of polynomials will be placed in stack levels 2 and 1 before using GCD. The results will be a polynomial or a list representing the greatest common denominator of the two polynomials or of each list of polynomials. Examples, in RPN mode, follow (calculator set to Exact mode):

'X^3-1' ENTER 'X^2-1' ENTER GCD Results in: 'X-1'

{'X^2+2*X+1','X^3+X^2'} ENTER {'X^3+1','X^2+1'} ENTER GCD results in {'X+1' 1}

The HERMITE function

The function HERMITE [HERMI] uses as argument an integer number, k, and returns the Hermite polynomial of k-th degree. A Hermite polynomial, $He_k(x)$ is defined as

$$He_0 = 1, \quad He_n(x) = (-1)^n e^{x^2/2} \frac{d^n}{dx^n}(e^{-x^2/2}), \quad n = 1,2,\ldots$$

An alternate definition of the Hermite polynomials is

$$H_0 * = 1, \quad H_n * (x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2}), \quad n = 1,2,...$$

where d^n/dx^n = n-th derivative with respect to x. This is the definition used in the calculator.

Examples: The Hermite polynomials of orders 3 and 5 are given by:

HERMITE(3) = '8*X^3-12*X',

And HERMITE(5) = '32*x^5-160*X^3+120*X'.

The HORNER function

The function HORNER produces the Horner division, or synthetic division, of a polynomial P(X) by the factor (X-*a*). The input to the function are the polynomial P(X) and the number *a*. The function returns the quotient polynomial Q(X) that results from dividing P(X) by (X-*a*), the value of *a*, and the value of P(*a*), in that order. In other words, P(X) = Q(X)(X-a)+P(a). For example, HORNER('X^3+2*X^2-3*X+1',2) = {'X^2+4*X+5', 2, 11}. We could, therefore, write $X^3+2X^2-3X+1 = (X^2+4X+5)(X-2)+11$. A second example: HORNER('X^6-1',-5)= {'X^5-5*X^4+25*X^3-125*X^2+625*X-3125',-5, 15624}
i.e., $X^6-1 = (X^5-5*X^4+25X^3-125X^2+625X-3125)(X+5)+15624$.

The variable VX

A variable called VX exists in the calculator's {HOME CASDIR} directory that takes, by default, the value of 'X'. This is the name of the preferred independent variable for algebraic and calculus applications. Avoid using the variable VX in your programs or equations, so as to not get it confused with the CAS' VX. If you need to refer to the x-component of velocity, for example, you can use vx or Vx. For additional information on the CAS variable see Appendix C.

The LAGRANGE function

The function LAGRANGE requires as input a matrix having two rows and *n* columns. The matrix stores data points of the form $[[x_1,x_2, \ldots, x_n]\ [y_1, y_2, \ldots, y_n]]$. Application of the function LAGRANGE produces the polynomial expanded from

$$p_{n-1}(x) = \sum_{j=1}^{n} \frac{\prod_{k=1,k\neq j}^{n}(x - x_k)}{\prod_{k=1,k\neq j}^{n}(x_j - x_k)} \cdot y_j.$$

For example, for n = 2, we will write:

$$p_1(x) = \frac{x - x_2}{x_1 - x_2} \cdot y_1 + \frac{x - x_1}{x_2 - x_1} \cdot y_2 = \frac{(y_1 - y_2) \cdot x + (y_2 \cdot x_1 - y_1 \cdot x_2)}{x_1 - x_2}$$

Check this result with your calculator:
LAGRANGE([[x1,x2],[y1,y2]]) = '((y1-y2)*X+(y2*x1-y1*x2))/(x1-x2)'.

Other examples: LAGRANGE([[1, 2, 3][2, 8, 15]]) = '(X^2+9*X-6)/2'
LAGRANGE([[0.5,1.5,2.5,3.5,4.5][12.2,13.5,19.2,27.3,32.5]]) =
'-(.1375*X^4+ -.7666666666667*X^3+ - .74375*X^2 +
1.991666666667*X-12.92265625)'.

Note: Matrices are introduced in Chapter 10.

The LCM function

The function LCM (Least Common Multiple) obtains the least common multiple of two polynomials or of lists of polynomials of the same length. Examples:

LCM('2*X^2+4*X+2' ,'X^2-1') = '(2*X^2+4*X+2)*(X-1)'.
LCM('X^3-1','X^2+2*X') = '(X^3-1)*(X^2+2*X)'

The LEGENDRE function

A Legendre polynomial of order n is a polynomial function that solves the differential equation $(1 - x^2) \cdot \frac{d^2y}{dx^2} - 2 \cdot x \cdot \frac{dy}{dx} + n \cdot (n+1) \cdot y = 0$

To obtain the *n*-th order Legendre polynomial, use LEGENDRE(*n*), e.g.,

LEGENDRE(3) = '(5*X^3-3*X)/2'
LEGENDRE(5) = '(63*X ^5-70*X^3+15*X)/8'

The PCOEF function

Given an array containing the roots of a polynomial, the function PCOEF generates an array containing the coefficients of the corresponding polynomial. The coefficients correspond to decreasing order of the independent variable. For example: PCOEF([-2,–1,0,1,1,2]) = [1. –1. –5. 5. 4. –4. 0.], which represents the polynomial X^6-X^5-5X^4+5X^3+4X^2-4X.

The PROOT function

Given an array containing the coefficients of a polynomial, in decreasing order, the function PROOT provides the roots of the polynomial. Example, from X^2+5X-6 =0, PROOT([1, –5, 6]) = [2. 3.].

The PTAYL function

Given a polynomial P(X) and a number *a*, the function PTAYL is used to obtain an expression Q(X-*a*) = P(X), i.e., to develop a polynomial in powers of (X- *a*). This is also known as a Taylor polynomial, from which the name of the function, Polynomial & TAYLor, follow:

For example, PTAYL('X^3-2*X+2',2) = 'X^3+6*X^2+10*X+6'.

In actuality, you should interpret this result to mean

'(X-2) ^3+6*(X-2) ^2+10*(X-2) +6'.

Let's check by using the substitution: 'X = x – 2'. We recover the original polynomial, but in terms of lower-case x rather than upper-case x.

The QUOT and REMAINDER functions

The functions QUOT and REMAINDER provide, respectively, the quotient Q(X) and the remainder R(X), resulting from dividing two polynomials, P_1(X) and P_2(X). In other words, they provide the values of Q(X) and R(X) from P_1(X)/P_2(X) = Q(X) + R(X)/P_2(X). For example,

QUOT(X^3-2*X+2, X-1) = X^2+X-1

REMAINDER(X^3-2*X+2, X-1) = 1.

Thus, we can write: (X^3-2X+2)/(X-1) = X^2+X-1 + 1/(X-1).

Note: you could get the latter result by using PROPFRAC:
PROPFRAC('(X^3-2*X+2)/(X-1)') = 'X^2+X-1 + 1/(X-1)'.

The EPSX0 function and the CAS variable EPS

The variable ε (epsilon) is typically used in mathematical textbooks to represent a very small number. The calculator's CAS creates a variable EPS, with default value 0.0000000001 = 10^{-10}, when you use the EPSX0 function. You can change this value, once created, if you prefer a different value for EPS. The function EPSX0, when applied to a polynomial, will replace all coefficients whose absolute value is less than EPS with a zero. Function EPSX0 is not available in the ARITHMETIC menu, it must be accessed from the function catalog (N). Example:

EPSX0('X^3-1.2E-12*X^2+1.2E-6*X+6.2E-11)=
'X^3-0*X^2+.0000012*X+0'.

With EVAL: 'X^3+.0000012*X'.

The PEVAL function

The functions PEVAL (Polynomial EVALuation) can be used to evaluate a polynomial $p(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots + a_2 \cdot x^2 + a_1 \cdot x + a_0$, given an array of coefficients $[a_n, a_{n-1}, \ldots a_2, a_1, a_0]$ and a value of x_0. The result is the evaluation $p(x_0)$. Function PEVAL is not available in the ARITHMETIC menu, it must be accessed from the function catalog (,N). Example:

PEVAL([1,5,6,1],5) = 281.

The TCHEBYCHEFF function

The function TCHEBYCHEFF(*n*) generates the Tchebycheff (or Chebyshev) polynomial of the first kind, order n, defined as $T_n(X) = \cos(n \cdot \arccos(X))$. If the integer n is negative (n < 0), the function TCHEBYCHEFF(*n*) generates the Tchebycheff polynomial of the second kind, order n, defined as $T_n(X) = \sin(n \cdot \arccos(X))/\sin(\arccos(X))$. Examples:

TCHEBYCHEFF(3) = 4*X^3-3*X

TCHEBYCHEFF(-3) = 4*X^2-1

Fractions

Fractions can be expanded and factored by using functions EXPAND and FACTOR, from the ALG menu (‚×). For example:

EXPAND('(1+X)^3/((X-1)*(X+3))') = '(X^3+3*X^2+3*X+1)/(X^2+2*X-3)'
EXPAND('(X^2)*(X+Y)/(2*X-X^2)^2)') = '(X+Y)/(X^2-4*X+4)'
EXPAND('X*(X+Y)/(X^2-1)') = '(X^2+Y*X)/(X^2-1)'
EXPAND('4+2*(X-1)+3/((X-2)*(X+3))-5/X^2') =

'(2*X^5+4*X^4-10*X^3-14*X^2-5*X+30)/(X^4+X^3-6*X^2)'

FACTOR('(3*X^3-2*X^2)/(X^2-5*X+6)') = 'X^2*(3*X-2)/((X-2)*(X-3))'
FACTOR('(X^3-9*X)/(X^2-5*X+6)') = 'X*(X+3)/(X-2)'
FACTOR('(X^2-1)/(X^3*Y-Y)') = '(X+1)/((X^2+X+1)*Y)'

The SIMP2 function

Functions SIMP2 and PROPFRAC are used to simplify a fraction and to produce a proper fraction, respectively. Function SIMP2 takes as arguments two numbers or polynomials, representing the numerator and denominator of a rational fraction, and returns the simplified numerator and denominator. For example: SIMP2('X^3-1','X^2-4*X+3') = { 'X^2+X+1','X-3'}.

The PROPFRAC function

The function PROPFRAC converts a rational fraction into a "proper" fraction, i.e., an integer part added to a fractional part, if such decomposition is possible. For example:

PROPFRAC('5/4') = '1+1/4'
PROPFRAC('(x^2+1)/x^2') = '1+1/x^2'

The PARTFRAC function

The function PARTFRAC decomposes a rational fraction into the partial fractions that produce the original fraction. For example:

PARTFRAC('(2*X^6-14*X^5+29*X^4-37*X^3+41*X^2-16*X+5)/(X^5-7*X^4+11*X^3-7*X^2+10*X)') =

'2*X+(1/2/(X-2)+5/(X-5)+1/2/X+X/(X^2+1))'

This technique is useful in calculating integrals (see chapter on calculus) of rational fractions.

If you have the Complex mode active, the result will be:

'2*X+(1/2/(X+i)+1/2/(X-2)+5/(X-5)+1/2/X+1/2/(X-i))'

The FCOEF function

The function FCOEF is used to obtain a rational fraction, given the roots and poles of the fraction.

Note: If a rational fraction is given as F(X) = N(X)/D(X), the roots of the fraction result from solving the equation N(X) = 0, while the poles result from solving the equation D(X) = 0.

The input for the function is a vector listing the roots followed by their multiplicity (i.e., how many times a given root is repeated), and the poles followed by their multiplicity represented as a negative number. For example, if we want to create a fraction having roots 2 with multiplicity 1, 0 with multiplicity 3, and -5 with multiplicity 2, and poles 1 with multiplicity 2 and –3 with multiplicity 5, use:

FCOEF([2, 1, 0, 3, –5, 2, 1, –2, –3, –5]) = '(X--5)^2*X^3*(X-2)/(X+3)^5*(X-1)^2'

If you press EVAL ← ANS ENTER (or, simply EVAL, in RPN mode) you will get:

'(X^6+8*X^5+5*X^4-50*X^3)/(X^7+13*X^6+61*X^5+105*X^4-45*X^3-297*X^2-81*X+243)'

The FROOTS function

The function FROOTS obtains the roots and poles of a fraction. As an example, applying function FROOTS to the result produced above, will result in: [1 –2. –3 –5. 0 3. 2 1. –5 2.]. The result shows poles followed by their multiplicity as a negative number, and roots followed by their multiplicity as a positive number. In this case, the poles are (1, -3) with multiplicities (2,5) respectively, and the roots are (0, 2, -5) with multiplicities (3, 1, 2), respectively.

Another example is: FROOTS('(X^2-5*X+6)/(X^5-X^2)')= [0 –2. 1 –1. 3 1. 2 1.]. i.e., poles = 0 (2), 1(1), and roots = 3(1), 2(1). If you have had Complex

mode selected, then the results would be:
[0 –2. 1 –1. – ((1+i*√3)/2) –1. – ((1–i*√3)/2) –1. 3 1. 2 1.].

Step-by-step operations with polynomials and fractions

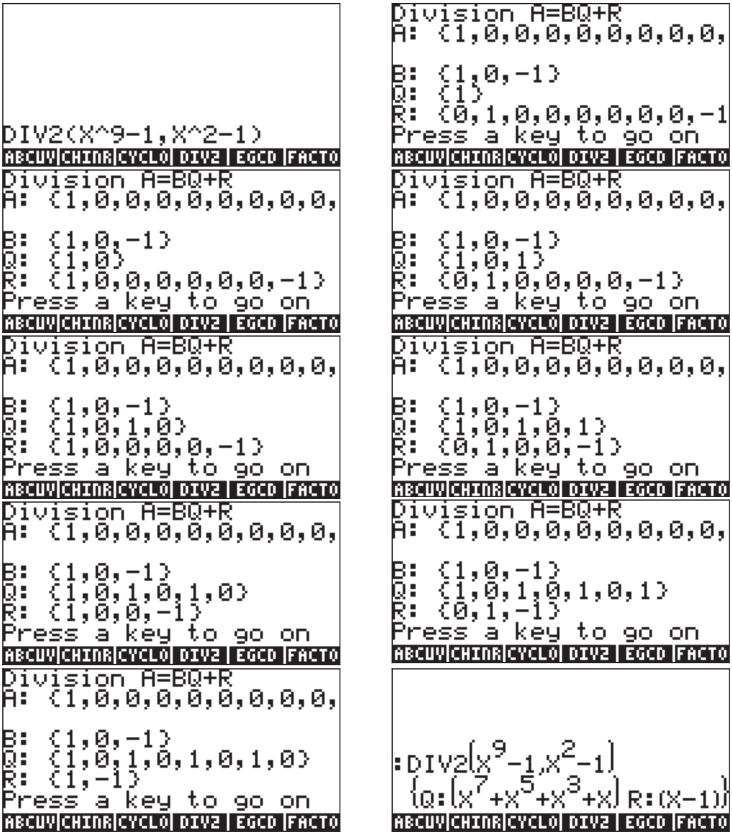
By setting the CAS modes to Step/step the calculator will show simplifications of fractions or operations with polynomials in a step-by-step fashion. This is very useful to see the steps of a synthetic division. The example of dividing

$$\frac{X^3 - 5X^2 + 3X - 2}{X - 2}$$

is shown in detail in Appendix C. The following example shows a lengthier synthetic division:

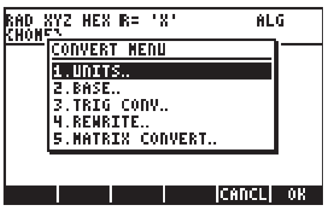
$$\frac{X^9 - 1}{X^2 - 1}$$

Note that DIV2 is available from the ARITH/POLYNOMIAL menu.



The CONVERT Menu and algebraic operations

The CONVERT menu is activated by using ← CONVERT key (the 6 key). This menu summarizes all conversion menus in the calculator. The list of these menus is shown next:



The functions available in each of the sub-menus are shown next.

UNITS convert menu (Option 1)

This menu is the same as the UNITS menu obtained by using → UNITS. The applications of this menu are discussed in detail in Chapter 3.

BASE convert menu (Option 2)

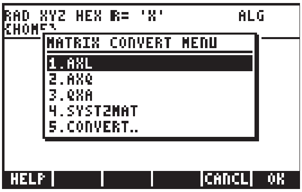
This menu is the same as the UNITS menu obtained by using ↱ BASE . The applications of this menu are discussed in detail in Chapter 19.

TRIGONOMETRIC convert menu (Option 3)

This menu is the same as the TRIG menu obtained by using ↱ TRIG . The applications of this menu are discussed in detail in this Chapter.

MATRICES convert menu (Option 5)

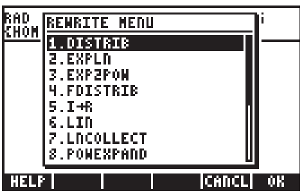
This menu contains the following functions:

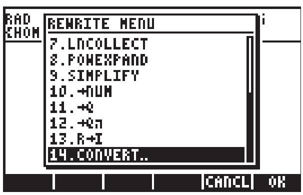


These functions are discussed in detail in Chapter 10.

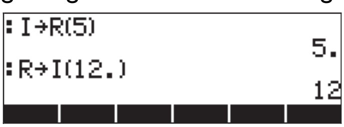
REWRITE convert menu (Option 4)

This menu contains the following functions:

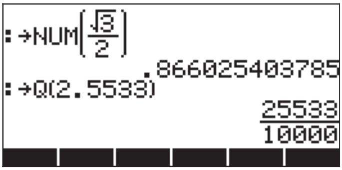


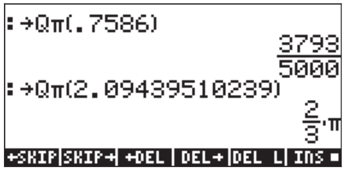


Functions I→R and R→I are used to convert a number from integer (I) to real (R), or vice versa. Integer numbers are shown without trailing decimal points, while real numbers representing integers will have a trailing decimal point, e.g.,



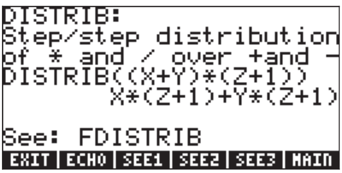
Function →NUM has the same effect as the keystroke combination ↱ →NUM (associated with the ENTER key). Function →NUM converts a symbolic result into its floating-point value. Function →Q converts a floating-point value into a fraction. Function →Qπ converts a floating-point value into a fraction of π, if a fraction of π can be found for the number; otherwise, it converts the number to a fraction. Examples are of these three functions are shown next.



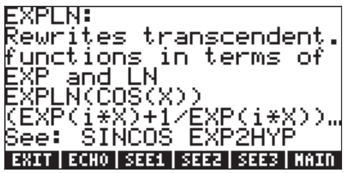


Out of the functions in the REWRITE menu, functions DISTRIB, EXPLN, EXP2POW, FDISTRIB, LIN, LNCOLLECT, POWEREXPAND, and SIMPLIFY apply to algebraic expressions. Many of these functions are presented in this Chapter. However, for the sake of completeness we present here the help-facility entries for these functions.

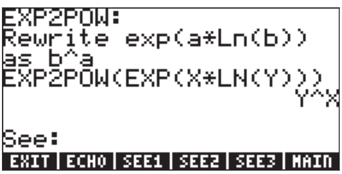
DISTRIB



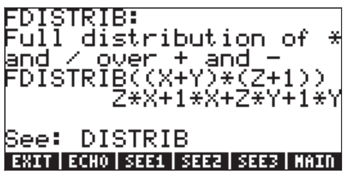
EXPLN



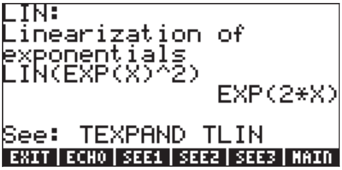
EXP2POW



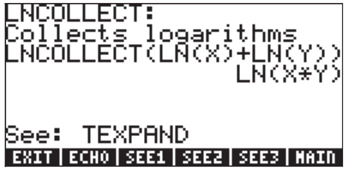
FDISTRIB



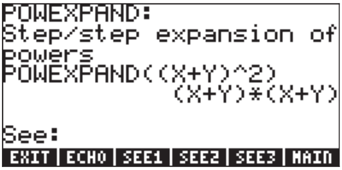
LIN



LNCOLLECT



POWEREXPAND



SIMPLIFY