Algebraic Operations With the HP 49 G

By

Gilberto E. Urroz, Ph.D., P.E.

Distributed by

infoClearinghouse.com

ALGEBRAIC OPERATIONS WITH THE HP 49 G 3

The CAS and RPN mode 3

Some transcendental functions 3
- Trigonometric functions 3
- Trigonometric identities 4
- What is a logarithm? 5
- Logarithms of base 10 5
- Natural logarithms and the exponential function 5
- Properties of the exponential function 6
- Properties of logarithms 6
- Converting logarithms of different bases 7
- Hyperbolic functions 7
- Euler formula and complex arguments for logarithmic, exponential, and hyperbolic functions 8

Operations with algebraic objects 10
- Entering algebraic objects 10
- Basic operations 11
- Expanding expressions 12
- Factoring expressions 12
- Complex mode factorization 13
- Substituting expressions in algebraic objects 13
- Substitution by using HP variables 14

Expansion and factorization using transcendental functions 15
- Expansion and factorization in terms of exponential and logarithmic functions 15
- Expansion and factorization in terms of trigonometric functions 17

Fraction Manipulation 20
- SIMP2 20
- PROPFAC 20
- PARTFRAC 20
- FCOEF 21
- FROOTS 21

Modular arithmetic 21
- Operations in modular arithmetic 21
- Formal definition of a finite arithmetic ring 22
- Finite arithmetic rings in the HP 49 G 23
- Modular arithmetic in the HP 49 G 23
  - Setting the modulus (or MODULO) 23
- Modular arithmetic operations with numbers 24
  - The modular inverse of a number 25
  - The MOD function 25
  - Other modular arithmetic functions 25

Polynomials 26
- Modular arithmetic with polynomials 26
- ABCUV 26
- CHINREM 27
- DIV2 27
- EGCD 27
- FACTOR 27
- FCOEF 27

FROOTS 28
GCD 28
HERMITE 28
HORNER 29
The variable VX, or “Why do you use only X in your examples?” 29
LAGRANGE 30
Entering matrices directly in the stack 30
LCM 30
LEGENDRE 31
Checking the solution to Legendre’s equation 31
PARTFRAC 32
PCOEF 32
PROOT 32
Direct access to polynomials numerical solution using NUM.SLV 32
PTAYL 33
QUOTIENT and REMAINDER 33
EPSX0 and the CAS variable EPS 33
PEVAL 33
TCHEBYCHEFF 34

REFERENCES (for all HP49 documents at InfoClearinghouse.com) 35

Algebraic operations with the HP 49 G

The Calculator Algebraic System or CAS allows the user to manipulate algebraic expressions including expansion, factorization, operations with fractions, substitutions, polynomial manipulation, and solution of single and multiple algebraic equations. This chapter introduces you to the basic of algebraic manipulation with the HP 49 G calculator. Before getting into the details of algebraic operations, however, we will formally define the logarithmic, exponential, hyperbolic and trigonometric functions, so as to take advantage of the HP 49 G algebraic menu commands that use such functions.

The CAS and RPN mode

The CAS allows the symbolic manipulation of algebraic objects (simply referred to as *algebraics*), but it does not mean that you need to operate your calculator in algebraic mode. This chapter will show you how to use algebraic manipulations with your calculator in RPN mode. To get started, make sure that your calculator CAS is set to `exact` mode (i.e., uncheck the `_Approx` mode indicator in the CAS MODES display. This display is accessed by using [MODES][CAS]).

Some transcendental functions

In Chapter 4 we presented some simple calculations utilizing exponential, logarithmic, trigonometric, and hyperbolic functions. Sometimes, these are referred to as transcendental functions, for they transcend the realm of algebraic functions. In this section we will introduce these formally and explain their relationships in order to take advantage of the many functions available in the HP 49 G for expanding, factoring, and replacing one type of function with others.

Trigonometric functions

Consider a unit radius circle (referred to as the *unit circle*) centered at the origin in a two-dimensional Cartesian coordinate system, as shown in the left-hand side figure below.

Let θ be the length of an arc of the circle measured from point A(1,0) and reaching to point $P(x,y)$ on the unit circle. We define the following *trigonometric functions*: sine (*sin*), cosine (*cos*), tangent (*tan*), cotangent (*cot*), secant (sec), cosecant (*csc*):

$$\sin\theta = y,\ \cos\theta = x,\ \tan\theta = y/x,\ \cot\theta = x/y,\ \sec\theta = 1/x,\ \csc\theta = 1/y.$$

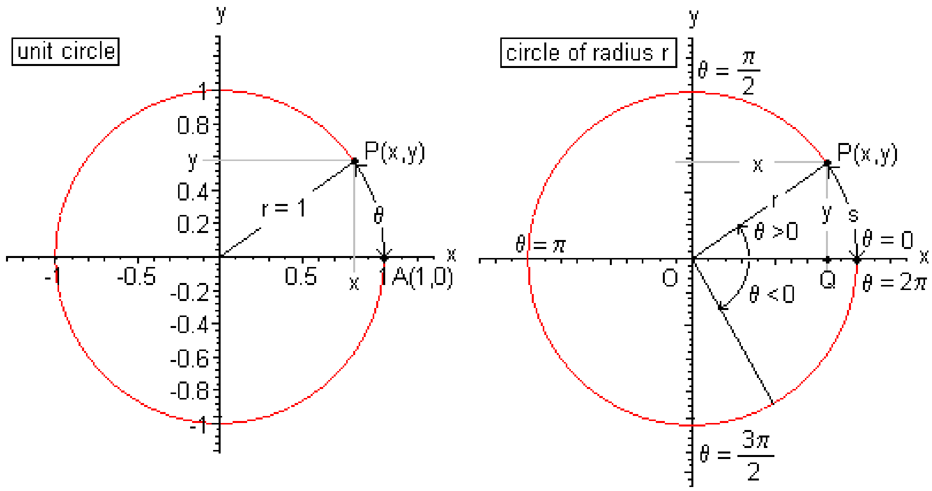
Recall, that the coordinates x and y used in the definition above *must* be on the unit circle, and that θ, as defined here, is an angle measured in radians. The angle θ is defined as positive if it is measured counterclockwise from the point the positive x-axis, and as negative if measured clockwise from that axis. The figure also shows typical values of the angle θ corresponding to the main axes directions.

Extending the definitions to a circle of radius $r\neq 1$, as shown in the right-hand side figure above, we define the angle in radians corresponding to an arc of length s as,

$$\theta = s/r.$$

The trigonometric functions, based on the coordinates of point P at the end of the arc, are defined as:

$$\sin\theta = y/r,\ \cos\theta = x/r,\ \tan\theta = y/x,\ \cot\theta = x/y,\ \sec\theta = r/x,\ \csc\theta = r/y.$$



The trigonometric functions sine, cosine, and tangent have their own main-function key in the HP 49 keyboard (fifth row, third, fourth, and fifth columns), i.e., [SIN], [COS], and [TAN]. The left-hand option ([↰]) for those keys represent the ***inverse trigonometric*** functions,

$$asin(x) = \sin^{-1}(x),\ acos(x) = \cos^{-1}(x),\ \text{and}\ atan(x) = \tan^{-1}(x).$$

The trigonometric functions cotangent, secant, and cosecant, can be calculated using the trigonometric identities:

$$\cot x = 1/\sin x,\ \sec x = 1/\cos x,\ \csc x = 1/\sin x.$$

Trigonometric identities

Trigonometric identities are relationships between the trigonometric functions of an angle that can be used to simplify algebraic expressions involving such functions. Some identities follow from the definition of the functions themselves, for example,

$$\tan\theta = \sin\theta/\cos\theta,\ \ \cot\theta = \cos\theta/\sin\theta.$$

Other identities follow from applying the Pythagorean Theorem to the right triangle formed by points OPQ, i.e., $x^2+y^2=r^2$. Dividing the expression by r^2 and using the definitions of $\sin\theta$ and $\cos\theta$ given above, we get

$$\sin^2\theta + \cos^2\theta = 1.$$

If we divide this identity by $\cos^2\theta$ and use the identities $1/\cos\theta = \sec\theta$, and $\sin\theta/\cos\theta = \tan\theta$, we get

$$\tan^2\theta + 1 = \sec^2\theta.$$

If we divide the identity $\sin^2\theta + \cos^2\theta = 1$ by $\sin^2\theta$, and use the identities $1/\sin\theta = \csc\theta$, and $\cos\theta/\sin\theta = \cot\theta$, we get

$$1+\cot^2\theta = \csc^2\theta.$$

Some identities are related to trigonometric functions of the sum and difference of angles, for example,

$$\sin(\alpha+\beta) = \sin\alpha \cdot \cos\beta + \cos\alpha \cdot \sin\beta$$

$$\sin(\alpha-\beta) = \sin\alpha \cdot \cos\beta - \cos\alpha \cdot \sin\beta$$

$$\cos(\alpha+\beta) = \cos\alpha \cdot \cos\beta - \sin\alpha \cdot \sin\beta$$

$$\cos(\alpha-\beta) = \cos\alpha \cdot \cos\beta + \sin\alpha \cdot \sin\beta$$

$$\tan(\alpha+\beta) = (\tan\alpha + \tan\beta)/(1 - \tan\alpha \cdot \tan\beta)$$

$$\tan(\alpha-\beta) = (\tan\alpha - \tan\beta)/(1 + \tan\alpha \cdot \tan\beta)$$

From the latter set of identities we can obtain some identities related to the double angle, i.e.,

$$\sin 2\theta = 2\sin\theta \cdot \cos\theta$$

$$\cos 2\theta = \cos^2\theta - \sin^2\theta = 1 - 2\sin^2\theta = 2\cos^2\theta - 1$$

$$\tan 2\theta = 2\tan\theta/(1 - \tan^2\theta)$$

And, from these, in turn, we can get some identities related to the half-angle, i.e.,

$$\sin^2(\theta/2) = (1 - \cos\theta)/2$$

$$\cos^2(\theta/2) = (1 + \cos\theta)/2$$

$$\tan^2(\theta/2) = (1 - \cos\theta)/(1 + \cos\theta).$$

What is a logarithm?

Let a, b, and r be three real numbers. If $a = b^r$, then r is said to be the logarithm of base b of a, and written

$$r = \log_b(a).$$

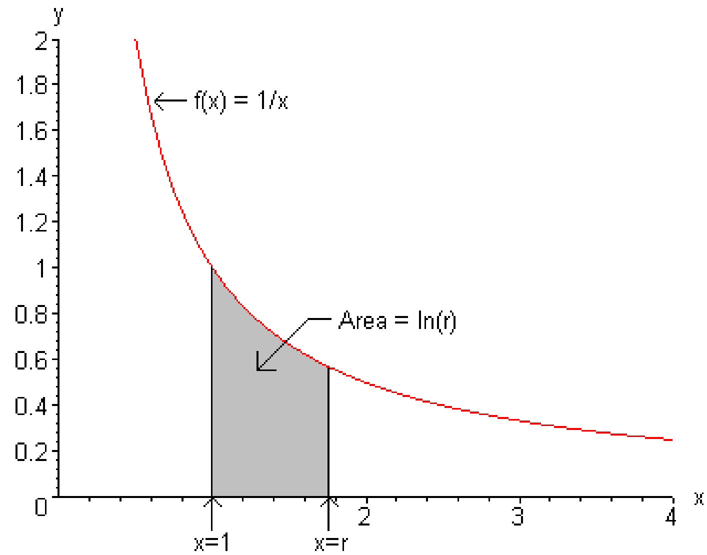
For example, since $32 = 2^5$, then $\log_2(32) = 5$.

Logarithms of base 10

Because our numerical system is based on powers of 10, logarithms of base 10 were the most commonly used logarithms before calculators and computers became readily available. Anyone who was serious about performing complex calculations before, say, the early 1970's, would keep in his library a reliable table of logarithms. This meant, of course, logarithms of base 10. When referring to the logarithm of base 10 of a number *a*, people would simply write '*log a*' or '*log(a)*'. For that reason, the HP 49 G provides the function LOG, as the [↦] option for the [EEX] key. The inverse of the LOG function is, of course, the function 10^x. This function is available as the [↦] option for the [EEX] key.

Natural logarithms and the exponential function

Natural logarithms have for base the irrational number e = 2.718281828…. [An irrational number, as opposite to a rational number, is a number that cannot be expressed as the ration of two integers. Examples of other famous irrational numbers are √2, √3, and π.] They are called natural logarithms because they follow naturally from some properties of the function *1/x*. As a matter of fact, you could define the ***natural logarithm*** of a real number $r>1$, as the area under the curve $f(x) = 1/x$, between $x = 1$ and $x = r$, as illustrated in the figure below. The natural logarithm of a number *r* is written as '*ln r*' or '*ln(r)*'.



The inverse of the *ln(x)* function is the ***exponential function***, $exp(x) = e^x$. The exponential and natural logarithm functions are defined as the left-shift [↰] and right-shift [↱] options, respectively, corresponding to the [y^x] key in your calculator's keyboard.

Properties of the exponential function

The exponential function, $exp(x) = e^x$, has the following properties:

$\exp(0) = e^0 = 1$	$\exp(-x) = e^{-x} = 1/e^x = 1/\exp(x)$
$\exp(x)+\exp(y) = e^{x+y} = \exp(x)\cdot\exp(y) = e^x\cdot e^y$	$\exp(x)-\exp(y) = e^{x-y} = \exp(x)/\exp(y) = e^x/e^y$
$[\exp(x)]^n = (e^x)^n = e^{xn} = \exp(nx)$	$[\exp(x)]^{1/n} = (e^x)^{1/n} = e^{xn} = \exp(nx)$

Properties of logarithms

Let *x*, *y*, *b*, be real numbers, then

$log_b b = 1,$	$log_b 1 = 0,$
$log_b(x\cdot y) = log_b x + log_b y$	$log_b(x/y) = log_b x - log_b y$
$log_b(y^x) = x log_b y$	$log_b(y^{1/x}) = log_b y/x$

In terms of natural logarithms, with $b = e$, these properties are written as:

$\ln e = 1$,	$\ln 1 = 0$,
$\ln(x \cdot y) = \log_b x + \log_b y$	$\ln(x/y) = \log_b x - \log_b y$
$\ln(y^x) = x \log_b y$	$\ln(y^{1/x}) = \log_b y/x$

Converting logarithms of different bases

Let x, y, a, b, and r, be real numbers. Let $x = \log_a r$ and $y = \log_b r$. Therefore, $r = a^x$, and $r = b^y$, and $a^x = b^y$. Taking, logarithms of base a on both sides of this equation, we get $\log_a(a^x) = \log_a(b^y)$, or $x \cdot \log_a a = y \cdot \log_a b$, and, $\log_a r = \log_b r \cdot \log_a b$. From this result it follows that,

$$\log_b r = \frac{\log_a r}{\log_a b}.$$

For example, let $b = e$, $a = 10$, then

$$\ln r = \frac{\log r}{\log e} = \frac{\log r}{0.4343} = 2.303 \cdot \log r.$$

and,

$$\log r = 0.4343 \cdot \ln r.$$

Hyperbolic functions

Hyperbolic functions are defined in terms of exponential functions as follows:

- Hyperbolic sine (*sinh*):

$$\sinh x = \frac{1}{2}(e^x - e^{-x}).$$

- Hyperbolic cosine (*cosh*):

$$\cosh x = \frac{1}{2}(e^x + e^{-x}).$$

- Hyperbolic tangent (*tanh*):

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

The *inverse hyperbolic functions* are defined as:

- Hyperbolic arcsin (*asinh*)

$$\mathrm{a}\sinh(x) = \ln(x + \sqrt{x^2 + 1})$$

- Hyperbolic arccos (*acosh*)

$$\mathrm{a}\cosh(x) = \ln(x + \sqrt{x^2 - 1})$$

- Hyperbolic arctan (*atanh*)

$$\mathrm{atanh}(x) = \ln\left(\frac{1+x}{1-x}\right)$$

These hyperbolic functions are available in the HP 49 G through the keystroke sequence [↰][MTH][HYP].

Other hyperbolic functions not available directly in the calculator are the hyperbolic cotangent (*coth*), hyperbolic secant (*sech*), and hyperbolic cosecant (*csch*), defined as:

coth x = 1/tanh x, sech x = 1/cosh x, csch x = 1/sinh x.

Euler formula and complex arguments for logarithmic, exponential, and hyperbolic functions

So far we have defined logarithmic, exponential, and hyperbolic functions for real numbers only. We can extend the definition of these functions to complex arguments by using Euler's formula:

$$e^{i\theta} = \cos\theta + i \cdot \sin\theta.$$

So that, if we use as argument for the functions previously defined the complex variable $z = x + i \cdot y = r \cdot e^{i\theta}$, we can extend their definitions as follows:

- natural logarithm: $ln(z) = ln(r \cdot e^{i\theta}) = r + i\theta = (x^2+y^2)^{1/2} + i \cdot atan(y/x).$
- exponential: $exp(z) = e^z = e^{x+iy} = e^x\, e^{iy} = e^x\,(cos\ y + i \cdot sin\ y).$
- hyperbolic sine: $sinh(z) = (e^z - e^{-z})/2 = sinh\ x \cdot cos\ y + i \cdot cosh\ x \cdot sin\ y$
- hyperbolic cosine: $cosh(z) = (e^z + e^{-z})/2 = cosh\ x \cdot cos\ y + i \cdot sinh\ x \cdot sin\ y$

✚ hyperbolic tangent:	$\tanh(z) = (e^z - e^{-z})/(e^z + e^{-z}) =$ $(\sinh x\cdot\cosh x + i\cdot\sin y\cdot\cos y)/(\cosh^2 x + \cos^2 y - 1)$	

Euler's formula also permits us to find expressions for the trigonometric functions in terms of complex arguments,

- sine: $\sin(z) = -i\cdot(e^{i\cdot z} - e^{-i\cdot z})/2 = \sin x\cdot\cosh y + i\cdot\cos x\cdot\sinh y$
- cosine: $\cos(z) = (e^{i\cdot z} + e^{-i\cdot z})/2 = \cos x\cdot\cosh y - i\cdot\sin x\cdot\sinh y$
- tangent: $\tan(z) = -i\cdot(e^{i\cdot z} - e^{-i\cdot z})/(e^{i\cdot z} + e^{-i\cdot z}) = (\sin x\cdot\cos x + i\cdot\sinh y\cdot\cosh y)/(\cos^2 x + \sinh^2 y)$

Inverse trigonometric and hyperbolic functions with complex arguments can be calculated by the expressions, where z = x + i·y:

- arcsine: $asin(z) = -i\cdot\ln(\sqrt{(1-z^2)}+i\cdot z)$
- arccosine: $acos(z) = -i\cdot\ln(z + i\cdot\sqrt{(1-z^2)})$
- arctangent: $atan(z) = i/2\cdot\ln((1-i\cdot z)/(1+i\cdot z))$
- hyperbolic arcsine: $asinh(z) = \ln(z+\sqrt{(z^2+1)})$
- hyperbolic arccosine: $acosh(z) = \ln(z+\sqrt{(z^2-1)})$
- hyperbolic arctangent: $atanh(z) = \ln((1+z)/(1-z))$.

The advantage of having in your hands a calculator that can calculate standard functions with complex arguments, is that you don't have to evaluate the definitions shown above to obtain values such as trigonometric and hyperbolic functions of complex variables. The following exercises will let you check the calculator's ability to compute standard functions using complex arguments. Before attempting thee exercises, check that your calculator CAS is set to ✓`Complex`. Use your HP 49 G calculator to obtain the values of:

ln(-5):	[5][+/-][↱][LN] [↱][→NUM]	Result: (1,6094,3.1416)
ln(3-2i)	[↰][( )][3][↱][,][2][+/-] [↱][LN]	Result: (1.2825, -0.5880)
exp(iπ/2):	[↰][i][SPC][↰][π][×][2][÷][↰][e^x]	Result: '*i*'
sin(2-3i):	[↰][( )][2][↱][,][3][+/-][SIN]	Result: (9.1544, 4.1689)
cos(-5+i):	[↰][( )][5] [+/-] [↱][,][1][COS]	Result: (0.4377,-1.269)
tan((1+i)/2)):	[1] [SPC][↰][i][+][2][÷][TAN][↱][→NUM]	Result: (0.4039,0.5641)

cot(-i/2):[↰][i][+/-][2][÷][TAN][↱][→NUM][1/x] Result: (0.,2.1640)

sec(2*i): [2][↰][i][×][COS][↱][→NUM][1/x] Result: (0.2658,0)

csc(-2+3i): [↰][( )][2][+/-][↱][,][3][SIN][1/x] Result: (-9.0473E-02, 4.1200E-02)

arcsin(5): [5][↰][ASIN] [↱][→NUM] Result: (1.5707,-2.2924)

arcsin(-5-2i): [↰][( )][5] [+/-] [↱][,][2][+/-][↰][ASIN] Result: (-1,1842,-2.3705)

arccos(-10): [1][0][+/-][↰][ACOS] [↱][→NUM] Result: (3.1416,-2.9932)

arccos(-2+4:i) [↰][( )][2] [+/-] [↱][,][4][↰][ACOS] Result: (2.0247,-2.1986)

arctan(-7+i): [↰][( )][7] [+/-] [↱][,][1][↰][ATAN] Result: (-1.4316,1.9617E-2)

For the next exercises, activate the menu [↰][MTH][HYP]:

sinh(-5+3i): [↰][( )][5] [+/-] [↱][,][3][SINH] Result: (73.46,10.47)

cosh(7-2i): [↰][( )][7] [↱][,][3] [+/-][COSH] Result: (73.46,10.47)

tanh(-2+7i): [↰][( )][2] [+/-] [↱][,][7][TANH] Result: (-0.9944,3.6094E-2)

arcsinh(-2+0.5i):[↰][( )][2] [+/-] [↱][,][.][5][ASINH] Result: (-1.4657,0.2211)

arccosh(-7+6i): [↰][( )][7] [+/-] [↱][,][6][ACOSH] Result: (2.9140,2.4300)

arctanh(-3-i): [↰][( )][3] [↱][,][1][ENTER][+/-][ATANH] Result: (2.9140,2.4300)

> **Note**: The menu [↰][MTH][HYP][NXT] contains two additional functions related to the exponential and natural logarithm functions: [EXPM] and [LNP1] defined as
>
> $$EXPM(x) = EXP(x)-1,$$
>
> and
>
> $$LNP1(x) = LN(x+1).$$

These functions are useful when writing programs to save programming steps.

Operations with algebraic objects

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. An algebraic object is an object of type 9 in the calculator. 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: ‘π*D^2/4’, ‘f*(L/D)*(V^2/(2*g))’, ‘y+Q^2/(2*g*A(y)^2)’
- An equation: ‘p*V=n*R*T’, ‘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]

Or, if using the equation writer:

[EQW] [↰][π][×][ALPHA][D][y^x][2] [▲][▲][▲] [÷][4][ENTER]

Basic operations

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 place them in a list so that they are available for algebraic manipulation at any time. Let's get started:

[↱][‘][↰][π][×][ALPHA][R][y^x][2][ENTER]	Enter ‘π*R^2’
[↱][‘][ALPHA][↰][G][×][ALPHA][↰][T][y^x][2][÷][4][ENTER]	Enter ‘g*t^2/4’
[2][↰][PRG][LIST][→LIST]	Create a list with two algebraics

The list will look like this in your calculator's display: `{ ‘π*R^2‘ ‘g*t^2/4‘ }`

Press [ENTER] several times, say 10 times, to make copies of the list of algebraic objects, so that we can use them for demonstrating operations. To decompose the list into its components, simply use [↱][EVAL].

Now, try the following *exercises*:

[↱][EVAL][+]	To add the algebraics, result: ‘π*R^2 + g*t^2/4‘
[⇦]	Drop result from display
[↱][EVAL][-]	To subtract the algebraics, result: ‘π*R^2 - g*t^2/4‘
[⇦]	Drop result from display
[↱][EVAL][×]	To multiply the algebraics, result: ‘π*R^2 *(g*t^2/4)‘
[⇦]	Drop result from display
[↱][EVAL][÷]	To divide the algebraics, result: ‘π*R^2 /(g*t^2/4)‘
[⇦]	Drop result from display
[↱][EVAL][2][y^x]	To square an algebraic, result: ‘(g*t^2/4)^2‘
[↱][ALG][EXPAN]	To ‘expand’ the square, result: ‘.0625*t^4.*g^2’
[⇦]	Drop result from display
[√x]	Square root of second algebraic, result: ‘√ (π*R^2)‘
[↱][ALG][EXPAN]	To ‘expand’ the square root, result: ‘1.77245385091*R’.
	Note: the constant π was replaced by its numerical value
[⇦]	Drop result from display
[↱][EVAL][SIN]	Sine function, result: ‘SIN(g*t^2/4)‘
[▶]	Swap levels 1 and 2
[↰][MTH][HYP][TANH]	Hyperbolic tangent function, result: ‘TANH(π*R^2)‘
[+]	Add the two previous results, result: ‘SIN(g*t^2/4)+ TANH(π*R^2)‘
[⇦]	Drop result from display

[↱][EVAL][↱][LN]	Natural log function, ‘LN(g*t^2/4)‘
[▶]	Swap levels 1 and 2
[↰][e^x]	Exponential function, result: ‘EXP(π*R^2)‘
[÷]	Divide the two previous results, result: ‘LN(g*t^2/4)/ EXP(π*R^2)‘
[⇦]	Drop result from display

As you can see operations with algebraic objects is not different than operations with numbers.

NOTE: Make sure your calculator is in the EXACT mode before attempting the following exercises.

Expanding expressions

Who among you does not still remember with great joy those long hours spent expanding algebraic expressions such as $(x+y)^3$ or $(a+b+c)^2$ in High School? Well, now you can let the calculator have all the fun by using the keystroke combination: [↱][ALG](the key for number 4)[EXPAN]. Try the following exercises:

[↱][‘][↰][()] [ALPHA][X] [+][ALPHA][Y] [▶][y^x][2] [ENTER]	Enter ‘(X+Y)^2’
[↱][ALG][EXPAN]	Result: ‘X^2+2*X*Y+Y^2’

Nice, isn’t it? No need to remember anymore that “… the square of X + Y is equal to the square of X plus twice the product of X times Y plus the square of Y.”

Now, let’s use the calculator to verify the following expansions. At this point I will assume that you know how to enter algebraic expressions in the stack, therefore, I will skip the detail keystroke sequences and provide the resulting algebraic only:

‘(X+Y)^3’[EXPAN]	Result: ’X^3+3*Y*X^2+3*Y^2*X+Y^3’
‘a*(X+Y)’[EXPAN]	Result: ‘a*X+a*Y’
‘(2+A)*A^2*(1+A)’[EXPAN]	Result: ‘A^4+3*A^3+2*A^2’
‘(a+b+c)^2’[EXPAN]	Result: ‘a^2+(2*b+2*c)*a+(b^2+2*c*b+c^2)’
‘2*(a+b)*(2+a)’[EXPAN]	Result: ‘2*a^2+(2*b+4)*a+4*b’

Notice that, in the last two exercises, the EXPAND function does not give you the full expansion than one expects. This may be a shortfall of the current version of the calculator’s CAS, or a deliberate attempt by the author(s) of the CAS to let you finish the calculation by hand. (The exercises in this chapter were developed using ROM version 1.16). The moral of the story is that , in some instances, the calculator will not carry operations to the ultimate expected expressions. (Even if you press [EXPAN] repeatedly). Keep a paper and pencil handy, and use it, when necessary, to finish some calculations by hand. For example,

$$(a+b+c)^2 = a^2+b^2+c^2+2ab+2ac+2cb,$$

and

$$2(a+b)(2+a) = 2a^2+4a+2ab+4b.$$

Factoring expressions

The function FACTOR, available through [↱][ALG][FACTO], can be used to obtain factorization of algebraic expressions. As in the case of EXPAN, however, there will be instances when the calculator will not be able to factor an expression down to its simplest factors, in which case, it will need some help from you.

Try the following *exercises*:

‘T^2+5*T+6’ [FACTO] Result: ‘(T+3)*(T+2)’

‘H^2-4’ [FACTO] Result: ‘(H+2)*(H-2)’

‘Z^4-8*Z^2+16’ [FACTO] Result: ‘(Z+2)^2*(Z-2)^2‘’

Note: If you work in *Approx* mode, your result would have been:‘(Z+2)*(Z-2)*(Z+2)*(Z-2)‘.

‘M^3+M^2-6*M’[FACTO] Result:‘(M+0)*(M-2)*(M+3)’ = M(M-2)(M+3)

‘X^2+Y^2+Z^2+2*X*Y+2*X*Z+2*Y*Z’[FACTO] Result: ‘(X+(Y+Z))*(X+(Y+Z))’=$(X+Y+Z)^2$

Complex mode factorization

If your CAS is set to ✓Complex mode, FACTOR will try to factor an algebraic expression down to their simplest factors. For example, working with real numbers only, the expression ‘pH2+4’ cannot be factored at all:

‘pH^2+4’ [FACTO] Result: ‘pH^2+4’

However, in CAS Complex mode you will get:

‘pH^2+4’ [FACTO] Result: ‘(pH,2)*(pH-2*i)’

The first factor in this result, ‘(pH,2)’, is the complex number ‘pH+2*i’ represented as an ordered pair.

Try a few more *examples* of Complex mode factorization:

‘xK^3+2*xK^2+4*xK+8’ [FACTO] Result:’(xK+2)*(xK,2)*(xK-2*i)’

‘Y^4+(m+2)*Y^3+(2*m+4)*Y^2+(4*m+8)*Y+8*m’ [FACTO]
Result: ’(Y+2)*(Y+m)*(Y,2)*(Y-2*i)’

‘p^3-β*p^2+2*α*p-2*α*β’ [FACTO] Result: ‘(p-b)*(p+i*√ (2*a))*(p-i*√2*√a)’

‘SIN(q)^2+2’ [FACTO] Result: ‘(SIN(q)+i*√2)*(SIN(q)-i*√2)’

Substituting expressions in algebraic objects

Substitution of algebraic expressions or numerical values in algebraic objects can be accomplished in two different ways. The first way uses the command [SUBS], available through the keystroke sequence [↦][ALG][SUBS]. Application of this command requires placing the algebraic object, where the substitution will take place, in stack level 2, and the substitution, in the form ’variable = value’ in stack level 1. For example, enter the algebraic object: ‘x^2+x+1’ [ENTER], and then enter ‘x = y+1’[ENTER], and press [SUBS]. The result is: ‘(y+1)^2+(y+1)+1’. Press [EXPAN] to expand and simplify the expression. The result is now: ‘y^2+3*y+3’. This mode of substitution allows substituting only one variable at a time. Other examples follow:

‘h = 30 + 2.5*t – 16.6*t^2’ [ENTER] ‘t = 4’ [ENTER][↦][ALG][SUBS][↦][EVAL] Results in ‘h=-225.6’
‘h = h0 + v0*t-g*t^2/2’ [ENTER] ‘t = tf’ [ENTER] [SUBS] Results in ‘h = h0 + v0*tf-g*tf^2/2’
Next, enter ‘tf = 2’ [SUBS] to get ‘h = h0 + v0*2-g*2^2/2’.
Next, enter ‘h0 = 10’ to get ‘h = 10 + v0*2-g*2^2/2’
Next, enter ‘h0 = 10’ to get ‘h = 10 + v0*tf-g*2^2/2’
Next, enter ‘v0 = 5’ to get ‘h = 10 + 5*2-g*2^2/2’

Next, enter ‘g = 9.806’ to get ‘h = 10 + 5*2-9.806*2^2/2’
Finally, to evaluate h, use [↦][EVAL]. The result is ‘h=.388’
‘V = Q/A’ [ENTER] ‘A = (b+m*y)*y’ [SUBS] Results in ‘V=Q/((b+m*y)*y)’.

A second form of substitution can be accomplished by using the [↦][|] (TOOL) key. This requires that the algebraic object, where the substitution will take place, be placed in stack level 2, while stack level 1 contains a list, of the form { variable1 value1 variable2 value2 …}. For example, enter the algebraic object ‘a+b^2+c’[ENTER], and then enter the list {a -1 b ‘x+1‘ c ‘π‘ }[ENTER]. Press [↦][|] [ENTER] to obtain ‘-1+((x+1)^2+π)’. Using [EXPAN] produces ‘π+(x^2+2*x)’. This mode of substitution allows substituting more than one variable at a time. The function [|] follows from the substitution notation:

$$(a+b^2+c)\,|_{\{a=-1,\,b=x+1,\,c=\pi\}}$$

Other examples:

‘h = h0 + v0*t-g*t^2/2’ [ENTER] { t 2 v0 5 h0 10 g 9.806 } [↦][|] [ENTER] Produces:
‘h = (2*10+(2*2*5-2^2*9.806))/2’. Use [↦][EVAL] to obtain ‘h =.388’

‘F = CONST(G)*m1*m2/r^2’ [ENTER] { m1 10_kg m2 20_kg r 12_m } [↦][|] [ENTER] Produces:
‘F=((6.67259E-11_m^3/(s^2*kg))*(20_kg)*10_kg)/(12_m)^2’ Use [↦][EVAL] to obtain
‘F =.9.26748611111E-11_m*kg/s^2’. The units are m*kg/s^2’ = N (newtons).

Substitution by using HP variables

The two substitution approaches presented above require that the original expression (where the substitution is to take place) and the substitution expressions be available in the stack. A different approach to substitution consists in defining the substitution expressions in variables and placing the name of the variables in the original expression. For example, store the following variables:

‘(b+m*y)*y’ [ENTER] ‘A’ [ENTER] [STO▶]
‘b+2*y*√ (1+m^2)’ [ENTER] ‘P’ [ENTER] [STO▶]
‘A/P’ [ENTER] ‘R’ [STO▶]

Then enter the expression:

‘Q = Cu*R^(2/3)*A*√S/n’ [ENTER] [ENTER]

(to keep an additional copy of the original expression). Enter [↦][EVAL] to evaluate the expression. The result is:

‘Q = (y*b+y^2*m)*√S*Cu*((y*b+y^2*m)/(b+2*y*√(m^2+1)))^.666666666667/n’

As you can see, the resulting expression is given in terms of the most primitive variables, i.e., b, m, y, which come from the definitions of A, P, and R.

Now, drop the last result from the stack by using the backspace key [], and purge the variables defined:
{ A P Q } [TOOL][PURGE]

With the expression ‘Q = Cu*R^(2/3)*A*√S/n’ in stack level 1, try [↦][EVAL]. The result is
‘Q = √S*A*Cu*R^.666666666667/n’

In the latter case, because there is no definition available for A or R, no substitution occurs. The only change, after evaluating the expression, is in the order of the terms in the expression, and the replacement of (2/3) by its approximation .666666666667, if the approximate mode is selected. In exact mode, the value (2/3) is kept.

Expansion and factorization using transcendental functions

The HP 49 G 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 left-shift key followed by the [8] key, i.e., [↦][TRIG]. The combination of this key with the right-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 factorization in terms of exponential and logarithmic functions

Here we use the EXP&LN menu obtained from the keystroke combination [↤][EXP&LN]. This produces the following menu:

[EXPLN][EXPM][LIN][LNCOL][LNP1][TEXPA]

And, after pressing [NXT], it shows as the second menu:

[TSIMP][ ][ ][ ][ ][ ]

We have described the operation of the functions [EXPM] (=exp(x)+1) and [LNP1] (= ln(x+1)) in an earlier note. Here, we describe the operation of the remaining functions in this menu.

- The function [EXPLN] transforms trigonometric functions into expressions involving exponential and natural logarithms, without linearizing the expression. Consider the following examples:

`'SIN(X)^2+1'` [EXPLN] Result: `'((EXP(i*X)-1/EXP(i*X))/(2*i))^2+1'`

`'SIN(X)+TAN(X)'` Result:
`'(EXP(i*X)-1/EXP(i*X))/(2*i)+(EXP(i*(2*x))1)/(i*(exp(i*(2*x))+1)'`

`'SIN(X+Y)*COS(X)'`[EXPLN] Result:
`'(EXP(I*(X+Y))-1/EXP(I*(X+Y))/(2*I)+(EXP(I*x)+1/EXP(I*X))/2'`

`'ASIN(X)'` `[EXPLN]` Result: `'i*LN(EXP(LN(X^2-1)/2)+X)+π/2'`

> **Note**: The expression obtained above include the function pair EXP and LN, which are inverse of each other, therefore, it can be simplified, by hand, to read $i\cdot\ln((x^2-1)^{1/2}+x)+\pi/2$. Earlier on, we defined the arcsin(z) as $-i\cdot\ln(\sqrt{1-z^2})+i\cdot z)$. Although, the two expressions are different, you can check that they produce the same result by calculating the two expressions with the same complex argument. For example, with the expression obtained earlier in stack level 1, enter the following:
>
> {X (2.,3.)} [ENTER][↦][|][ENTER] [↦][→NUM], to get (0.57065, 1.98338).
>
> Next, type in:
>
> '-i*LN(√(1-X^2)+i*X)' [ENTER] {X (2.,3.)} [ENTER][↦][|][ENTER] [↦][→NUM],
>
> to get (0.57065, 1.98338).

The function [LIN] linearizes expressions containing exponential or logarithmic terms. For example:

‘SIN(X)’ [LIN] results in ‘-(i/2*EXP(i*x))+i/2*EXP(-(i*X))’

‘π*EXP(2*X)*EXP(Y)’ [LIN] results in ‘π*EXP(2*X+Y)’

‘SIN(LN(X^2-1))’ [LIN] results in ‘-(i/2*EXP(i*LN(X^2-1)))+i/2*EXP(-(i*LN(X^2-1)))’

‘SIN(X)-COSH(X)’ [LIN] results in ‘-1/2*EXP(X)+ –1/2*EXP(-X)-i/2*EXP(i*X)+i/2*EXP(-(i*X))’

The function [LNCOL] (LNCOLLECT) collects logarithmic terms in an expression. Examples:

‘LN(X) + LN(X-1)-LN(X+1)’ [LNCOL] results in ‘LN(X*(X-1)/(X+1))’

‘(1/2)*LN(X^2-1)+LN(X)’ using [LNCOL] on this expression has no effect. However, if you press [LIN], the expression gets transformed into ‘(LN(X^2-1)-2*LN(X))/2’. Using [LNCOL] on this new version of the expression produces ‘LN(X^2-1)/X^2)/2’. Since $ln(a)/n = ln(a^{1/n})$, we can write the latter result as

$$\frac{1}{2}\cdot\ln\left(\frac{X^2-1}{X^2}\right)=\ln\left(\frac{X^2-1}{X^2}\right)^{1/2}=\ln\left(\sqrt{\frac{X^2-1}{X^2}}\right)=\ln\left(\frac{\sqrt{X^2-1}}{X}\right).$$

‘2+(1/3)*LN(Y)-LN(k)’ [LIN] results in ‘(LN(Y)-(3*LN(k)-6)/3’. Using [LNCOL] on this result, produces ‘(6+LN(Y/k^3))/3’. This result can be simplified even further if we replace 6 with a LN expression. Since 6 = LN(EXP(6))=LN(403.43), we can write the latter expression as ‘(LN(403.43)+LN(Y/k^3))/3’. Press [LNCOL] to get ‘‘(LN(403.43*(Y/k^3))/3’.

‘(1/2)*LN(Y)-(1/3)*LN(X+1)’ [LIN] results in ‘-((2*LN(X+1)-3*LN(Y))/6’. Using [LNCOL] on this result produces ‘-LN((X+1)^2/Y^3)/6’.

‘LOG(X)-LOG(3)-(1/2)*LOG(Y)’ [LIN] results in ‘(2*LN(X)-(LN(Y)+2*LN(3)))/(2*LN(10))’. Using [LNCOL] on this result produces ‘-(LN(Y*9/X^2)/LN(100))’.

The function [TEXPA] (TEXPAND) expands expressions containing transcendental functions (i.e., exponential, logarithmic, hyperbolic, and trigonometric functions). Some examples follow:

‘SIN(EXP(X*Y))’ [TEXPA] results in ‘COS(LN(Y))*SIN(LN(X))+SIN(LN(Y))*COS(LN(X))’.

‘COS(A-B)’ [TEXPA] results in ‘COS(B)*COS(A)+SIN(B)*SIN(A)’

‘LN((2*X-1)/(2*X+1)^(1/3))’ [TEXPA] results in ‘LN(2*X-1)-LN(XROOT(3,2*X+1))’. Now, apply the function [LNCOL] to get ‘-LN(XROOT(3,2*X+1)/(2*X-1))’.

Note: XROOT(x,y) represents $\sqrt[x]{y} = y^{1/x}$.

‘EXP(X+Y^2-A)’ [TEXPA] results in ‘EXP(X)*EXP(Y^2)/EXP(A)’.

The function [TSIMP] (visible if you press [NXT]) simplifies expressions involving natural logarithms and exponential functions. For example, expressions involving trigonometric or hyperbolic functions get replaced by their equivalent in exponential functions. On the other hand, expressions involving inverse trigonometric and hyperbolic functions get replaced by their equivalent in logarithmic functions. Some examples of the use of [TSIMP] are shown below:

'SIN(X)'[TSIMP] results in 'EXP(i*X)-1/EXP(i*X))/(2*i)'. Press [NXT][LIN] to transform the latter expression into '-(i/2*EXP(I*X)+i/2*EXP(-(I*X))'.

'ATAN(X)' [TSIMP] produces
'i/2*((LN(X^2+1)/2,ATAN(1/X)+ - 2*(π/2))+-1*(LN(X^2+1)/2,ATAN(X)+-1*(π/2)))'
If we use the function [LIN] on this expression it gets simplified to
'(-(i*LN((i*X+1)/(i*X-1)))+ (i *LN(-((X+1)/(X-i)))+π))/4'.

'SIN(X) + COS(X)' [TSIMP] results in '(EXP(i*X)-1/EXP(i*X))/(2*I)+(EXP(i*X)+1/EXP(i*X))/2' . Using the function [LIN] of this result produces '(1-i)/2*EXP(i*X)+(1+i)/2*EXP(-(i*X))'.

'ASIN(X)+LN(X)' [TSIMP] results in 'i*LN(X+ √ (X^2-1))+π/2+LN(X)'. Next, use [LIN] to produce '(2*i*LN(X+EXP(LN(X^2-1)/2))+(2*LN(X)+π))/2'. Using [LNCOL] produces '(2*i*LN(X+EXP(LN(X^2-1)/2))+ π + LN(X^2))/2'.

Expansion and factorization in terms of trigonometric functions

In this section we present functions that let you convert trigonometric functions into other trigonometric functions. These menus are accessed through the keystroke combination [↦][TRIG], which produces the following:

Menu 1: [HYP][ACOS2][ASIN2][ASIN2][ATAN2][HALFT].

Use [NXT] to get

Menu 2: [SINCO][TAN2S][TAN2S][TCOLL][TEXPA][TLIN]

Use [NXT] once more to get

Menu 3: [TRIG][TRIGC][TRIGS][TRIGT][TSIMP][ ]

The [HYP] menu provides access to the hyperbolic function menu as you would using [↤][MTH][HYP].

The functions [TEXPA] and [TSIMP] were presented in the previous section.

The function [ACOS2] (ACOS2S) replaces occurrences of ACOS(X) with 'π/2 – ASIN(X)'. Examples:

'ACOS(X^2-1)' [ACOS2]	Results in 'π/2 – ASIN(X^2-1)'.
'ACOS(SIN(3*π/2))' [ACOS2]	Results in 'π/2 + π/2'
'ACOS(1/X)' [ACOS2]	Results in 'π/2 – ASIN(1/X)'.

You will notice that in the first menu there are two keys labeled [ASIN2]. In fact, the two [ASIN2] labels represent the functions [ASIN2C] and [ASIN2T], in that order. The function ASIN2C replaces occurrences of ASIN(X) with ‘π/2 – ACOS(X)’, while the function ASIN2T replaces occurrences of ASIN(X) with ‘ATAN(X/√ (1-X^2))’. Examples:

‘ASIN(Y/2)’ [ASIN2C]	Results in ‘π/2 – ACOS(Y/2)’.
‘ASIN(Y/2)’ [ASIN2T]	Results in ‘ATAN(Y/2/(√ -(Y^2-4)/2))’
‘ASIN(X)+ASIN(Y)’ [ASIN2C]	Results in ‘π/2-ACOS(X)+(π/2-ACOS(Y))’
‘ASIN(TAN(X))’ [ASIN2T]	Results in ‘ATAN(TAN(X) /(√ -(TAN(X)^2-1))

The function [ATAN2](ATAN2S) replaces occurrences of ATAN(X) with ASIN(X/√ (X^2+1)). Examples:

‘ATAN(SIN(π/2))’ [ATAN2S]	Results in ‘ASIN(1/√2)’. With [↦][EVAL] we get π/4.
‘ATAN(ABS(X))’ [ATAN2S]	If in complex mode, results in ‘ASIN(√ (RE(X)^2+IM(X)^2)/ √ (RE(X)^2+IM(X)^2+1))’ If in real mode, results in ‘ASIN(ABS(X)/ √ (X^2+1))’
‘ATAN(X+Y)’ [ATAN2S]	Results in ‘ASIN((X+Y)/ √ (X^2+2*Y*X+(Y^2-1)))’

The function [HALFT] (HALFTAN) replaces occurrences of SIN(X), COS(X), and TAN(X), with expressions involving TAN(X/2).

‘SIN(X)’ [HALFT]	Results in ‘2*TAN(X/2)/(SQ(TAN(X/2))+1)’
‘COS(X)’ [HALFT]	Results in ‘1-SQ(TAN(X/2)))/(SQ(TAN(X/2)+1)’
‘TAN(X)’ [HALFT]	Results in ‘2*TAN(X/2)/(1-SQ(TAN(X/2)))’
‘SIN(X)+COS(X)’ [HALFT]	Results in

‘2*TAN(X/2)/(SQ(TAN(X/2))+1) + (1-SQ(TAN(X/2))/(SQ(TAN(X/2))+1)’

The function [SINCO] (SINCOS) converts natural logarithm and exponential expressions with complex arguments into expressions involving the functions sine and cosine. Set your calculator to complex mode before trying these exercises:

‘EXP((2,3))’ [SINCO]	Results in ‘EXP(2)*(COS(3)+i*SIN(3))’
‘EXP(Z)’[SINCO]	Results in ‘EXP(RE(Z))*(COS(IM(Z))+ i*SIN(IM(Z)))’
‘EXP(i*π)’ [SINCO]	Results in ‘-1’
‘LN(X+I*Y)’ [SINCO]	Results in ‘(LN(X^2+Y^2)/2, ATAN(Y/X)+0*(π/2))’

There are two [TAN2S] functions listed in Menu 2. They represent the functions TAN2SC and TAN2SC2, in that order. The function TAN2SC replaces occurrences of TAN(X) with ‘SIN(X)/COS(X)’. Examples:

‘TAN(Y/2)’ [TAN2S] (TAN2SC)	Results in ‘SIN(Y/2)/COS(Y/2)’
‘TAN(X)^2’[TAN2S] (TAN2SC)	Results in ‘(SIN(X)/COS(X))^2’

The operation of TAN2SC2 depends on whether flag –116 is set [Prefer Sin( )], in which case TAN(X) gets replaced by ‘(1-COS(2*X))/SIN(2*X))’. If flag –116 is clear [Prefer Cos( )], TAN(X) gets replaced by ‘SIN(2*X)/(1+COS(2*X))’. Examples:

Set flag –116, as follows: [1][1][6][+/-][ENTER][ALPHA][S][ALPHA][F][ENTER].

‘TAN(X)’ [TAN2S](TAN2SC2)	Results in ‘(1-COS(2*X))/SIN(2*X)’
‘TAN(π/6)’ [TAN2S](TAN2SC2)	Results in ‘(1-1/2)/ (√3/2)’

‘TAN(X/2)’ [TAN2S](TAN2SC2) Results in ‘(1-COS(X))/SIN(X)’

Clear flag –116, as follows: [1][1][6][+/-][ENTER][ALPHA][C][ALPHA][F][ENTER].

‘TAN(X)’ [TAN2S](TAN2SC2) Results in ‘SIN(2*X)/(1+COS(2*X))’.
‘TAN(π/6)’ [TAN2S](TAN2SC2) Results in ‘√3/2/(1+1/2)’
‘TAN(X/2)’ [TAN2S](TAN2SC2) Results in ‘SIN(X)/(1+COS(X))’

- The function [TCOLL] (TCOLLECT) collects terms involving sine and cosine of the same arguments. Examples:

‘SIN(X)*COS(Y)’ [TCOLL] Results in ‘1/2*SIN(X-Y)+1/2*SIN(X+Y)’
‘SIN(X)^2+2*SIN(X)*COS(Y)+COS(Y)^2’ [TCOLLECT] Results in
‘SIN(2*X)+ -
1/2*COS(2*X)+1+1/2*COS(2*Y)’
‘SIN(X)^2’ [TCOLL] Results in ‘-1/2*COS(2*X)+1/2’

- The function [TLIN] linearizes trigonometric function expressions without collecting those with the same argument. Examples:

‘SIN(X)^2+2*SIN(X)*COS(Y)+COS(Y)^2’ [TLIN] Results in
‘SIN(X-Y)+SIN(X+Y) - 1/2*COS(2*X)+1+1/2*COS(2*Y)’
‘SIN(X)^2’ [TLIN] Results in ‘-1/2*COS(2*X)+1/2’ (Same result as with [TCOLL]
‘SIN(X)*COS(X)^2’ [TLIN] Results in ‘1/4*SIN(3*X)+1/4*SIN(X)’

- The function [TRIG] replaces exponential and logarithmic functions with complex arguments with equivalent trigonometric functions. Examples:

‘LN(X+i*Y)’ [TRIG] Results in ‘(LN(X^2+Y^2)+2*I*ATAN(Y/X))/2’
‘EXP(X+i*Y)’ [TRIG] Results in
‘EXP(RE(X)-IM(Y))*COS(RE(Y)+IM(X))+I*EXP(RE(X)-IM(Y))*SIN(RE(Y)+IM(X))’

‘EXP(Z)’ [TRIG] Results in ‘EXP(RE(Z))*COS(IM(Z))+I*EXP(RE(Z))*SIN(IM(Z))’

- The function [TRIGC] (TRIGCOS) uses the identity ‘SIN(X)^2+COS(X)^2=1’ to simplify expressions while returning cosine terms only if possible. Examples:

‘SIN(X)^2-COS(X)^2’ [TRIGC] Results in ‘-(2*COS(X)^2-1)’
‘(SIN(X)-TAN(X))^2’ [TRIGC] Results in ‘-(COS(X)^2-(TAN(X)^2+1)+2*TAN(X)*SIN(X))’
‘(SIN(X)+COS(X))^2’ [TRIGC] Results in ‘1+2*COS(X)*SIN(X)’

- The function [TRIGS] (TRIGSIN) uses the identity ‘SIN(X)^2+COS(X)^2=1’ to simplify expressions while returning sine terms only if possible.

‘SIN(X)^2-COS(X)^2’ [TRIGS] Results in ‘2*SIN(X)^2-1’
‘(SIN(X)-TAN(X))^2’ [TRIGC] Results in ‘SIN(X)^2-2*TAN(X)*SIN(X)+TAN(X)^2’
‘(SIN(X)-COS(X))^2’ [TRIGC] Results in ‘1-2*COS(X)*SIN(X)’

- The function [TRIGT] (TRIGTAN) replaces terms involving SIN(X) and COS(X) with expressions involving TAN(X).

‘SIN(X)^2-COS(X)^2’ [TRIGC] Results in ‘(TAN(X)^2-1)/(TAN(X)^2+1)’
‘(SIN(X)-TAN(X))^2’ [TRIGC] Results in
‘(TAN(X)^4+2*TAN(X)^2-2*TAN(X)^2*(COS(X)*(SQ(TAN(X))+1)))/(TAN(X)^2+1)’

‘(SIN(X)+COS(X))^2’ [TRIGC] Results in ‘(TAN(X)^2+2*TAN(X)+1)/(TAN(X)^2+1)’

Fraction Manipulation

Fractions can be expanded and factored by using [↱][ALG][EXPAND] and [FACTOR]. For example:

‘(1+X)^3/((X-1)(X+3))’ [EXPAN] Results in ‘(X^3+3*X^2+3*X+1)/(X^2+2*X-3)’
‘X^2*(X+Y)/(2*X-X^2)^2’ [EXPAN] Results in '(X+Y)/(X^2-4*X+4)’
‘X+(X+Y)/(X^2-1)’ [EXPAN] Results in ‘(X^3+Y)/(X^2-1)’
‘4+2*(X-1)+3/((X-2)*(X+3))-5/X^2’ [EXPAN] Results in
‘(2*X^5+4*X^4-10*X^3-14*X^2-5*X)/(X^4+X^3-6*X^2)’
‘(3*X^3-2*X^2)/(X^2-5*X+6)’ [FACTO] Results in ‘X^2*(3*X-2)/((X-2)*(X-3))’
‘(X^3-9*X)/(X^2-5*X+6)’ [FACTO] Results in ‘X*(X+3)/(X-2)’
‘(X^2-1)/(X^3*Y-Y)’ [FACTO] Results in ‘(X+1)/((X^2+X+1)*Y)’

SIMP2

There are a couple of functions in the menu [↰][ARITH] (the key for [1]) [NXT] that apply to fractions: [SIMP2] and [PROPF]. The function [SIMP2] takes as arguments two numbers or polynomials, in stack levels 2 and 1, representing the numerator and denominator, respectively, of a rational fraction, and returns the simplified numerator and denominator in stack levels 2 and 1, respectively. For example:

‘X^3-1’[ENTER]’X^2-4*X+3’[ENTER][↰][ARITH][NXT][SIMP2] Results in:

```
2:        ‘X^2+X+1‘
1:            ‘X-3‘
```

The operation of [SIMP2] is not different from [FACTO] if you write the full fraction, i.e.,

‘(X^3-1)/(X^2-4*X+3)’ [FACTO], to get ‘(X^2+X+1)/(X-3)’

PROPFAC

The function PROPFAC ([↰][ARITH][NXT][PROPF]) 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:

‘5/4’ [↰][ARITH][NXT][PROPF] Results in ‘1+1/4’
‘(x^2+1)/x^2’ [PROPF] Results in ‘1+1/x^2’

PARTFRAC

The function PARTFRAC ([↰][ARITH][POLY][NXT][NXT][PARTF]) decomposes a rational fraction into the partial fractions that produce the original fraction. For example:

‘(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)’
[↰][ARITH][POLY][NXT][NXT][PARTF] Results in ‘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))’

FCOEF

The function FCOEF is used to obtain rational fraction, given the roots and poles of the fraction. This function is accessible through: [↰][ARITH][POLY][FCOEF].

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. The input is:

[2 1 0 3 –5 2 1 -2 -3 -5][FCOEF]. The result is: '(X—5)^2*X^3*(X-2)/9X—3)^5*(X-1)^2'

If you press [↱][EVAL], 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*X62-81*X+243)'

FROOTS

The function FROOTS ([FROOT]) obtains the roots and poles of a fraction. This function is accessible through [↰][ARITH][POLY][NXT][FROOT]. As an example, if you kept the latest result in your stack, press [FROOT] to obtain: [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: '(X^2-5*X+6)/(X^5-X^2)' [FROOT] Results in [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].

Modular arithmetic

Modular arithmetic is used in some operations with polynomials. Therefore, we deemed necessary to present the subject in this chapter.

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, we can refer to arithmetic operations in such system as finite arithmetic operations or, simply, *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⋅3 = 21 = 12 + 9, (or, 7⋅3/12 = 21/12 = 1 + 9/12, i.e., the integer reminder of 21/12 is 9). We can now write 7⋅3 ≡ 9 (mod 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, $r/k \equiv j$ (mod n), if, $j \cdot k \equiv r$ (mod n). This means that r must be the remainder of $j \cdot k/n$. For example, 9/7 ≡ 3 (mod 12), because 7⋅3 ≡ 9 (mod 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*6 (mod 12)	0
6*1 (mod 12)	6	6*7 (mod 12)	6
6*2 (mod 12)	0	6*8 (mod 12)	0
6*3 (mod 12)	6	6*9 (mod 12)	6
6*4 (mod 12)	0	6*10 (mod 12)	0
6*5 (mod 12)	6	6*11 (mod 12)	6

Formal definition of a finite arithmetic ring

The expression

$$a \equiv b \ (mod\ n)$$

is interpreted as "<u>a is congruent to b, modulo n</u>," and holds if $(b-a)$ is a multiple of n. With this definition the rules of arithmetic simplify to the following:

If

$$a \equiv b \ (mod\ n) \text{ and } c \equiv d \ (mod\ n),$$

then

$$a+c \equiv b+d \ (mod\ n),$$
$$a-c \equiv b - d \ (mod\ n),$$
$$a \times c \equiv b \times d \ (mod\ n).$$

For division, follow the rules presented earlier.

For example, 17 ≡ 5 (mod 6), and 21 ≡ 3 (mod 6). Using these rules, we can write:

$$17 + 21 \equiv 5 + 3 \ (mod\ 6) \quad \Rightarrow \quad 38 \equiv 8 \ (mod\ 6) \quad \Rightarrow 38 \equiv 2 \ (mod\ 6)$$
$$17 - 21 \equiv 5 - 3 \ (mod\ 6) \quad \Rightarrow \quad -4 \equiv 2 \ (mod\ 6)$$
$$17 \times 21 \equiv 5 \times 3 \ (mod\ 6) \quad \Rightarrow \quad 357 \equiv 15 \ (mod\ 6) \quad \Rightarrow \ 357 \equiv 3 \ (mod\ 6)$$

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 HP 49 G

All along we have defined our finite arithmetic operation so that the results are always positive. The modular arithmetic system in the HP 49 G 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 HP 49 G includes the numbers:

(-3,-2,-1,0,1,3,4),

while for n = 7 (odd), the corresponding HP 49 G finite arithmetic ring is given by

(-3,-2,-1,0,1,2,3).

Modular arithmetic in the HP 49 G

To launch the modular arithmetic menu in the HP 49 G use [↰][ARITH][MODUL] (ARITH is the left-shift key corresponding to the key for number [1]). The available menu includes:

[ADDTM][DIVMO][DIV2M][EXPAN][FACTO][GCDMO]

Press [NXT] to get the second modular arithmetic menu, which shows the functions:

[INVMO][MOD][MODST][MULTM][POWMO][SUBTM]

The use of this functions is presented next.

<u>Setting the modulus (or MODULO)</u>

The calculator creates a variable called MODULO that is placed in the HOME directory and will store the magnitude of the modulus to be used in modular arithmetic. To get the calculator to create this variable, if it does not already exists in the HOME directory, launch the modular arithmetic menu, by using [↰][ARITH][MODUL]. Then, enter a number, say [8], and press the soft menu key corresponding to [EXPAN]. If the MODULO variable you should get the result –1. To check out your MODULO variable, press [VAR] and enter [↱][MODUL]. The default value of 3 should be placed in the stack indicating that such value currently occupies the variable MODULO.

To change the value of MODULO, you can either store the new value directly in the variable MODULO by placing the value in the stack and then using [↰][MODUL]. If you have already activated the modular arithmetic menu, move to the second menu by using [NXT], place the new value in the stack, and press the soft key labeled [MODST] (MODSTO) to store the new modulus value.

For example, to perform arithmetic operations of modulus 12, use:

[1][2][ENTER] [↰][ARITH][MODUL][NXT][MODST]

Modular arithmetic operations with numbers

To add, subtract, multiply, divide, and raise to a power using modular arithmetic you will use the functions ADDTM, SUBTM, MULTM, DIV3M and DIVMO (these two for division), and POWMO. 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, having stored 12 as our modulus, try the following operations:

[6][SPC][5][ADDTM]	Result: -1, i.e., 6+5 ≡ -1 (mod 12)
[6][SPC][6][ADDTM]	Result: 0, i.e., 6+6 ≡ 0 (mod 12)
[6][SPC][7][ADDTM]	Result: 1, i.e., 6+7 ≡ 1 (mod 12)
[1][1][SPC][5][ADDTM]	Result: 4, i.e., 11+5 ≡ 4 (mod 12)
[8][SPC][1][0][ADDTM]	Result: -6, i.e., 8+10 ≡ -6 (mod 12)

[5][SPC][7][SUBTM]	Result: -2,i.e.,5 - 7 ≡ -2 (mod 12)
[8][SPC][4][SUBTM]	Result: 4, i.e.,8 – 4 ≡ 4 (mod 12)
[5][SPC][1][0]][SUBTM]	Result: -5, i.e., 5 –10 ≡ -5 (mod 12)
[1][1][SPC][8][SUBTM]	Result: 3, i.e., 11 – 8 ≡ 3 (mod 12)
[8][SPC][1][2][SUBTM]	Result: -4, i.e., ≡ 8 - 12 (mod 12)

[6][SPC][8][MULTM]	Result: 0, i.e., 6·8 ≡ 0 (mod 12)
[9][SPC][8][MULTM]	Result: 0, i.e., 9·8 ≡ 0 (mod 12)
[3][SPC][2][MULTM]	Result: 6, i.e., 3·2 ≡ 6 (mod 12)
[5][SPC][6][MULTM]	Result: 6, i.e., 5·6 ≡ 6 (mod 12)
[1][1][SPC][3][MULTM]	Result: -3, i.e., 11·3 ≡ -3 (mod 12)

[1][2][SPC][3][DIVMO]	Result: 4, i.e., 12/3 ≡ 4 (mod 12)
[1][2][SPC][8][DIVMO]	Result: No solution in ring, i.e., 12/8 (mod 12) does not exist
[2][5][SPC][5][DIVMO]	Result: 5, i.e., 25/5 ≡ 5 (mod 12)
[6][4][SPC][1][3][DIVMO]	Result: 4 , i.e., 64/13 ≡ 4 (mod 12)
[6][6][SPC][6][DIVMO]	Result: -1, i.e., 66/6 ≡ -1 (mod 12)

[1][2][SPC][5][DIV2M]	Result: , 0 and 0 , i.e., 2/3(mod 12) ≡ 0 (mod 12) with remainder = 0
[2][6][SPC][1][2][DIV2M]	Result: No solution in ring, i.e., 26/12 (mod 12) does not exist
[1][2][5][SPC][1][7][DIV2M]	Result: 1 and 0, i.e., 125/17 (mod 12) ≡ 1 with remainder = 0
[6][8][SPC][7][DIV2M]	Result: -4 and 0, i.e., 68/7 ≡ 0 (mod 12) with remainder = 0
[7][SPC][5][DIV2M]	Result: -1 and 0, i.e., 7/5 ≡ -1 (mod 12) with remainder = 0

Note: DIVMO provides the quotient of the modular division j/k (mod n), while DIMV2M provides no only the quotient but also the remainder of the modular division j/k (mod n).

[2][SPC][3][POWMO]	Result:-4 , i.e., 2^3≡ -4 (mod 12)
[3][SPC][5][POWMO]	Result: 3, i.e., 3^5≡ 3 (mod 12)
[5][SPC][1][0][POWMO]	Result: 1, i.e., 5^{10}≡ 1 (mod 12)
[1][1][SPC][8][POWMO]	Result: 1, i.e., 11^8 ≡ 1 (mod 12)
[6][SPC][2][POWMO]	Result: 0, i.e., 6^2 ≡ 0 (mod 12)
[9][SPC][9][POWMO]	Result: -3, i.e., 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 [EXPAND] (full name is EXPANDMOD). For example,

[1][2][5][EXPAND]	Result: 5, or 125 ≡ 5 (mod 12)

[1][7][EXPAND]	Result: 5, or 17 ≡ 5 (mod 12)
[6][6][EXPAND]	Result: 6, or 66 ≡ 6 (mod 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 second menu resulting from [↰][ARITH][MODUL].

For example, in modulus 12 arithmetic:

[6][INVMOD]	Result: No solution in ring, i.e., 1/6 (mod 12) does not exist.
[5][INVMOD]	Result: 5, i.e, 1/5 ≡ 5 (mod 12)
[7][INVMOD]	Result: -5, i.e., 1/7 ≡ -5 (mod 12)
[3][INVMOD]	Result: No solution in ring, i.e, 1/3 (mod 12) does not exist.
[1][1][INVMOD]	Result: -1, i.e., 1/11 ≡ -1 (mod 12)

The MOD function

The MOD function is used to obtain the ring number of a given modulus corresponding to a given integer number. To operate this function in RPN mode, enter the integer number to be converted to a ring number followed by the modulus, and separated by an [ENTER] or a [SPC]. Then press [MOD]. On paper this operation is written as

$$m \bmod n = p$$

and is read as "*m* modulo *n* is equal to *p*".

For example, to calculate 15 mod 8, enter

[1][5][SPC][8][MOD].

The result is 7, i.e., 15 mod 8 = 7.

Try the following exercises:

[1][8][SPC][1][1][MOD]	Result: 7, i.e., 18 mod 11 = 7
[2][3][SPC][2][MOD]	Result: 1, i.e., 23 mod 2 = 1
[4][0][SPC][1][3][MOD]	Result: 1, i.e., 40 mod 13 = 1
[2][3][SPC][1][7][MOD]	Result: 6, i.e., 23 mod 17 = 6
[3][4][SPC][6][MOD]	Result: 4, i.e., 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.

Other modular arithmetic functions

Other functions included in the modular arithmetic menu are [GCDMO] (GCDMOD), or greatest common divisor under modular arithmetic, and [FACTO] (FACTORMOD), or factorization under modular arithmetic. These functions are used on polynomial expressions to produce the GCM of two polynomials, or to factor a polynomial. Examples of such applications are presented in the next section.

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). The HP 49 G calculator includes a large number of functions for manipulating polynomials, besides the EXPAND and FACTOR functions, already introduced.

Polynomial manipulation functions are available in the calculator menu: [↰][ARITH][POLY]:

Menu 1: [ABCUV][CHINR][DIV2][EGCD][FACTO][FCOEF]

Press [NXT]

Menu 2: [PROOT][GCD][HERMI][HORNE][LAGRA][LCM]

Press [NXT]

Menu 3: [LEGEN][PARTF][PCOEF][PROOT][PTAYL][QUOT]

Press [NXT]

Menu 4: [REMAI][ ][ ][ ][ ][ARITH]

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 $P(X) = X \ (mod\ X^2)$, or another polynomial $Q(X) = X + 1\ (mod\ X\text{-}2)$. Some polynomial functions apply to such algebraic rings, as presented below.

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:

$$P(X) \equiv Q(X)\ (mod\ M(X)).$$

The later expression is interpreted as “*P(X) is congruent to Q(X), modulo M(X)*”.

ABCUV

Given polynomials A(X), B(X), and C(X), in stack levels 3, 2, and 1, respectively, [ABCUV] returns two polynomials, U(X) and V(X), in stack levels 2 and 1, respectively, so that:

C(X) = U(X)*A(X) + V(X)*B(X).

For example, with A(X) = X^2-1, B(X) = X^2+1, C(X) = X^3+2*X^2-7, use:

‘X^2-1’ [ENTER] ‘X^2+1’ [ENTER] ‘X^3+2*X^2-7’ [ENTER][ABCUV] gives

```
2:   ‘(X^3+2*X-7)/-2‘
1:   ‘-(X^3+2*X-7)/-2‘
```


CHINREM

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 numbers. The input consists of two vectors *[expression_1, modulo_1] and [expression_2, modulo_2]*, in stack levels 2 and 1, respectively. The output is a vector *containing [expression_3, modulo_3]*, where *modulo_3* is related to the product *(modulo_1)·(modulo_2)*. Example:

[‘X+1’, ‘X^2-1’][ENTER][‘X+1’,’X^2’][ENTER] [CHINREM]   Results in [‘’X+1’,-(X^4-X^2)]

DIV2

Given polynomials P(X) in stack level 2, and Q(x) in stack level 1, the function [DIV2] returns the quotient and residual of P(X)/Q(X) in stack levels 2 and 1, respectively. For example:

‘X^3-1’[ENTER]’X-5’[ENTER][DIV2] Results in:

```
2:      ‘X^2+5*X+25‘
1:              124
```

EGCD

EGCD stands for Extended Greatest Common Divisor. Given two polynomials, A(X) and B(X), in stack levels 2 and 1, respectively, it produces the polynomials C(X), U(X), and V(X), in stack levels 3, 2, and 1, respectively, 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, use:

‘X^2+1’[ENTER] ‘X^2-1’ [ENTER] [EGCD] Results in

```
3:                2
2:                1
1:               -1
```

i.e., 2 = 1*(X^2+1’)-1*(X^2-1).

Try another example:

‘X’ [ENTER] ‘X^3-2*X+5’ [ENTER] [EGCD] Results in

```
3:                5
2:       ’-(X^2-2)’
1:                1
```

i.e., 5 = – (X^2-2)*X + 1*(X^3-2*X+5)

FACTOR

The function FACTOR [FACTO] has been presented earlier in the chapter.

FCOEF

The function FCOEF was presented in an earlier section.

FROOTS

The function FROOTS was presented in an earlier section.

GCD

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 follow:

'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}

HERMITE

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

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

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.

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

[3][HERMI] which produces '8*X^3-12*X' ,

and

[5][HERMI] which gives '32*x^5-160*X^3+120*X'

Note: To check which of the two definitions of Hermite polynomials given above is the one that the calculator uses, enter the following:

[EQW][↰][( )][1][+/-][▲][▲][▲][y^x][3][▲][▲][×] [↰][e^x][ALPHA][X][y^x][2][▶][▶][▶][×]
[↱][∂][ALPHA][X] [▶][↱][∂][ALPHA][X] [▶][↱][∂][ALPHA][X] [▶]
[↰][e^x][ALPHA][X][+/-][y^x][2] [ENTER]

Resulting in '(-1)^3*EXP(x^2)*∂x(∂x(∂x(EXP(-x^2))))'. Press [ENTER] to keep an extra copy of the expression in the stack. Then, enter [↱][EVAL]. The result is '(8*X^3-12*X)*EXP(x^2)*EXP(-x^2)'. Pressing [↱][EVAL], once more, we get the result '8*X^3-12*X', which is the same produced with [5][HERMI]. Therefore, we verified, at least for n = 3, that the second definition, i.e., that of H*n(x), is the one used for the calculator to generate Hermite's polynomials.

From this exercise we learn that the keystroke combination [↱][∂] generates derivative symbols in the equation writer.

HORNER

The function HORNER [HORNE] 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), in stack level 2, and the number *a*, in stack level 1. 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 stack levels 3, 2, and 1, respectively. In other words, P(X) = Q(X)(X-a)+P(a). For example,

'X^3+2*X^2-3*X+1' [ENTER] 2 [ENTER] [HORNE] produces the result:

```
3:      'X^2+4*X+5'
2:                2
1:               11
```

We could, therefore, write $X^3+2X^2-3X+1 = (X^2+4X+5)(X-2)+11$.

A second example, would be:

'X^6-1'[ENTER] [5][+/-] [ENTER] [HORNE], which results in :

```
3:'X^5-5*X^4+25*X^3-
125*X^2+625*X-3125'
2:               -5
1:            15624
```

i.e., $X^6-1 = (X^5-5*X^4+25X^3-125X^2+625X-3125)(X+5)+15624$.

The variable VX, or "Why do you use only X in your examples?"

The first time you activate any of the CAS functions, i.e., algebraic, calculus, etc., a variable called VX is created in your HOME directory that takes, by default, the value of 'X'. This is the name of the preferred independent variable for algebraic and calculus applications. For that reason, I have used X as the unknown variable in all the examples in this section. If you use other independent variable names, for example, with HORNER, the CAS will not work properly.

The variable VX is a permanent inhabitant of the HOME directory. There are other CAS variables in the HOME directory that you may or may not have currently in your calculator. Some of these variables are REALASSUME ([REALA]), MODULO ([MODUL]), CASINFO ([CASIN]), etc. You can purge all of them, except VX. Try it.

You can change the value of VX by storing a new algebraic name in it, e.g., 'x', 'y', 'm', etc. In the rest of this section, and in the rest of the book, I will assume that 'X' will be the standard value of VX and I will continue using X as the preferred independent variable for CAS applications.

Also, 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.

LAGRANGE

The function LAGRANGE [LAGRA] requires as input a matrix having two rows and *n* columns. The matrix stores data points of the form [[x_1 x_2 … x_n] [y_1 y_2 … 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:

[['x1' 'x2']['y1' 'y2']][ENTER][LAGRA]. The result is: '((y1-y2)*X+(y2*x1-y1*x2))/(x1-x2)'.

Other examples:

[[1 2 3][2 8 15]][ENTER][LAGRA] produces '(X^2+9*X-6)/2'

[[0.5 1.5 2.5 3.5 4.5][12.2 13.5 19.2 27.3 32.5]] [ENTER] [LAGRA] produces
'-(.1375*X^4+ -.7666666666667*X^3+ - .74375*X^2 = 1.991666666667*X-12.92265625)'.

Entering matrices directly in the stack

Matrices are introduced in a latter chapter. They are basically arrays of numbers, or symbols, in rows and columns. The matrix corresponding to the latter problem's input would be written on paper as

$$\begin{bmatrix} 0.5 & 1.5 & 2.5 & 3.5 & 4.5 \\ 12.2 & 13.5 & 19.2 & 27.3 & 32.5 \end{bmatrix}$$

In the calculator stack, you can enter the matrix by opening a pair of square brackets and using sets of square brackets to represent the rows of the matrix. For the example under consideration, this is:

[[0.5 1.5 2.5 3.5 4.5][12.2 13.5 19.2 27.3 32.5]].

LCM

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

‘2*X^2+4*X+2’ [ENTER] ‘X^2-1’ [ENTER] [LCM] produces ‘(2*X^2+4*X+2)*(X-1)’.

‘X^3-1’[ENTER] ‘X^2+2*X’ [ENTER] [LCM] produces ‘(X^3-1)*(X^2+2*X)’

LEGENDRE

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, enter the order of the polynomial, then press [LEGEN] (LEGENDRE). Examples:

[3][LEGEN] produces: ‘(5*X^3-3*X)/2’
[5][LEGEN] produces: ‘(63*X ^5-70*X^3+15*X)/8’

Checking the solution to Legendre’s equation

To verify that these polynomials indeed satisfy Legendre’s equation, enter the general expression of the equation in a variable to be called LEGEQ, as follows:

[EQW][↰][()] [1][-][ALPHA][X][y^x][2][▶][▶][▶][▶] [×][↱][∂][ALPHA][X] [▶] [↱][∂][ALPHA][X][▶] [ALPHA][Y][↰][()][ALPHA][X][▶][▶][▶][▶][-][2][×][ALPHA][X][×][↱][∂][ALPHA][X][▶] [ALPHA][Y][↰][()][ALPHA][X][▶] [▶][▶][▶][▶][▶][+][ALPHA][↰][N] [×] [↰][()] [ALPHA][↰][N] [+] [1] [▶][▶][▶][×] [ALPHA][Y] [↰][()][ALPHA][X][▶] [▶][▶][▶][▶][▶][↱][=][0] [ENTER]

[‘][ALPHA][ALPHA][L][E][G][E][Q][ENTER][STO▶]

Next, bring the equation back to the stack, by pressing [VAR][LEGEQ]. Replace the value of n with 3, i.e., [↰][{}][ALPHA][↰][N][SPC][3][ENTER][↱][|][ENTER]. The result is:
‘-((Xˆ2-1)*d1d1Y(X)+(2*X*d1Y(X)-(3^2+3)*Y(X)))=0’

Note: d1Y(X) now represents dY/dX, and d1d1Y(X) represents d^2Y/dX^2.

Use [↱][ALG][EXPAN] to get: ‘-((Xˆ2-1)*d1d1Y(X)+(2*X*d1Y(X)-12*Y)=0’

Next, we generate the third-order Legendre polynomial:

[3][↰][ARITH][POLY][NXT][NXT][LEGEN] to get ‘(5*X^3-3*X)/2’

Next, enter ‘Y(X)’ by using: [↱][‘][ALPHA][Y] [↰][()][ALPHA][X][ENTER]

Press [▶] to exchange stack levels 1 and 2, and then [↱][=] to get: ‘Y(X) =‘(5*X^3-3*X)/2’

Press [↱][ALG][SUBST] to replace Y(X) with the Lagrange polynomial, and then press [↱][EVAL]. The result is: ‘ 0 = 0’.

PARTFRAC

The function PARTFRAC was presented in an earlier section.

PCOEF

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:

[-2 –1 0 1 1 2] [PCOEF] produces: [1. –1. –5. 5. 4. –4. 0.], which represents the polynomial X^6-X^5-5X^4+5X^3+4X^2-4X.

PROOT

Given and array containing the coefficients of a polynomial, in decreasing order, the function PROOT provides the roots of the polynomial. As PCOEF, this is a numerical solution originally developed for the HP 48 G/G+/GX series calculator. Example:

From X^2+5X-6 =0, we enter [1 –5 6][ENTER][PROOT]. The function provides the result: [2. 3.].

Direct access to polynomials numerical solution using NUM.SLV

- The function PCOEF corresponds to the numerical solution for polynomial coefficients given its roots (first developed for the HP 48 G/G+/GX series calculator). This solution is also accessible using an input form by using the keystroke sequence: [↦][NUM.SLV] (NUMerical SoLVer). This will generate a dropdown menu. Use the down-arrow key, [▼], twice to highlight `3. Solve poly`…. Press [OK]. The input form will show the field corresponding to the polynomial coefficients highlighted. Press [▼] to move to the `Roots:` field. In this field, enter the array or vector of roots, i.e.,

  [↤][[ ]][2][+/-][SPC][1][+/-][SPC][0][SPC][1][SPC][1][SPC][2][ENTER] (or [OK])

  This will send the cursor back to the `Coefficients` field. Now, press [SOLVE]. The result is the vector [1. –1. –5. 5. 4. … in the `Coefficients` field. To see the entire array, press [EDIT]. This triggers the matrix writer (a sort of spreadsheet used to enter array data) showing a matrix with one column. To see the elements of the array use the right-arrow or left-arrow keys to move about the array. Verify that the array is [1. –1. –5. 5. 4. –4. 0.]. Press [ENTER] when done. Press [ENTER] to return to normal calculator display. You will notice that the polynomial coefficient array has been copied to the stack.

- The NUM.SLV menu can be used to produce the polynomial and place it into the stack by using a procedure similar to that used above, but ending by pressing the key [SYMB] after using [SOLVE]. Repeat the exercise above an try this option. Press [ENTER], when done, to return to normal calculator display. You should get the expression: ‘X^6+-1*X^5+-5*X^4+5*X^3+4*X^2+-4*X’.

- Start the polynomial solution from the NUM.SLV once more. This time we will provide the polynomial coefficients in the proper field, say [1 –5 6], corresponding to X^2-5X+6 = 0, and find the roots by pressing [SOLVE] when the `Roots:` field is highlighted. The solution should be [2. 3.].

PTAYL

Given a polynomial P(X) and a number *a* (stack levels 2 and 1, respectively), 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,'X^3-2*X+2'[ENTER] [2] [ENTER] [PTAYL] produces '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' [↦][ALG][SUBST] [EXPAN]. We recover the original polynomial, but in terms of lower-case x rather than upper-case x.

QUOTIENT and REMAINDER

The functions QUOTIENT [QUOT] and REMAINDER [REMAI] 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,

'X^3-2*X+2'[ENTER] 'X-1' [ENTER] [QUOT] results in 'X^2+X-1'
'X^3-2*X+2'[ENTER] 'X-1' [ENTER] [REMAI] results in 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 PARTFRAC:
'(X^3-2*X+2)/(X-1)' [ENTER] [PARTF] results in 'X^2+X-1 + 1/(X-1)'.

The following functions are not accessible through the POLY menu:

EPSX0 and the CAS variable EPS

The variable ε (epsilon) is typically used in mathematical textbooks to represent a very small number. The HP 49 G CAS will create 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 in stack level 1, will replace all coefficients whose absolute value is less than EPS with a zero. Examples:

'X^3-1.2E-12*X^2+1.2E-6*X+6.2E-11'[CAT][ALPHA][E]. Use the up and down arrow keys to find EPSX0, then press [OK]. The result is 'X^3-0*X^2+.0000012*X+0'. Use [↦][ALG][EXPAN] to obtain 'X^3+.0000012*X+0'.

PEVAL

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} … a_2 a_1 a_0] and a value of x_0, placed in stack levels 2 and 1, respectively. The result is the evaluation $p(x_0)$. Examples:

[1. 5. 6. 1][ENTER][5][ENTER][CAT][ALPHA][P]. Use up and down arrows to find PEVAL, press [OK]. The result is 281.

TCHEBYCHEFF

Given an integer number, n>0, in stack level 1, the function TCHEBYCHEFF 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 generates the Tchebycheff polynomial of the second kind, order n, defined as

$$T_n(X) = sin(n\cdot arccos(X))/sin(arccos(X)).$$

This function is not available in the POLY menu. To invoke it you have the use the catalog, i.e., [CAT][ALPHA][T], then use the down-arrow key [▼] nine times until the command TCHEBYCHEFF is highlighted. Press [OK] to activate the command. Examples:

[5][CAT][ALPHA][T] [▼][▼][▼][▼][▼][▼][▼][▼][▼] [OK] produces '16*X^5-20*x^3+5*X'

At this point the TCHEBYCHEFF command is readily available in the catalog. So, for the next example, use:

REFERENCES (for all HP49 documents at InfoClearinghouse.com)

Devlin, Keith, 1998, “The Language of Mathematics,” W.H. Freeman and Company, New York.

Farlow, Stanley J., 1982, “Partial Differential Equations for Scientists and Engineers,” Dover Publications Inc., New York.

Friedman, B., 1956, “Principles and Techniques of Applied Mathematics,” (reissued 1990), Dover Publications Inc., New York.

Gullberg, J., 1997, “Mathematics – From the Birth of Numbers,” W. W. Norton & Company, New York.

Harris, J.W., and H. Stocker, 1998, “Handbook of Mathematics and Computational Science,” Springer, New York.

Heath, M. T., 1997, “Scientific Computing: An Introductory Survey,” WCB McGraw-Hill, Boston, Mass.

Hewlett Packard Co., 1999, HP 49 G GRAPHING CALCULATOR USER’S GUIDE.

Hewlett Packard Co., 2000, HP 49 G GRAPHING CALCULATOR ADVANCED USER’S GUIDE

Kottegoda, N. T., and R. Rosso, 1997, “Probability, Statistics, and Reliability for Civil and Environmental Engineers,” The Mc-Graw Hill Companies, Inc., New York.

Kreysig, E., 1983, “Advanced Engineering Mathematics – Fifth Edition,” John Wiley & Sons, New York.

Newland, D.E., 1993, “An Introduction to Random Vibrations, Spectral & Wavelet Analysis – Third Edition,” Longman Scientific and Technical, New York.

Tinker, M. and R. Lambourne, 2000, “Further Mathematics for the Physical Sciences,” John Wiley & Sons, LTD., Chichester, U.K.