‘Mates8’:

Calculator of Polynomials

User’s Guide

Version 8.3.62

Contents

1 HOW TO INPUT DATA

1.1 COMMON INPUTS

By common inputs we refer to a mathematical expression as, for example: 2+2, (1+1/10)^10, (x-100)*(x+100) or sin(PI/2) where numbers are expressed in decimal base and trigonometric function's argument in radians. If the expression can be evaluated into a real or complex number, this will be the output.

The calculator will insert implicit operators in an expression such as 2x3-x2-1 converting to 2*x^3-x^2-1; (x-1)2 will be converted to (x-1)^2; and (x-1)(x+1) into (x1)*(x+1).

Numbers may be entered using a comma (,) as thousand separator and a dot (.) as decimal mark, for example: 1,234,567.89

1.2 COMMON ERRORS

Easily, common errors are not detected by the parser, because they match possible valid inputs, and the output will be unpredictable. In this case, just the user expertise gets to the right answer. A common error is to write a function definition as f(x)= x^2-2x+1 (also applies for y= x^2-2x+1) in the wrong box, this is, in the box for expressions instead of the box "variables, functions and constants". For the calculator, 'f(x)= x^2-2x+1' is an equation and the left side 'f(x)' represents f*(x) = f*x, i.e. the product of 2 variables, and the right side is a polynomial 'x^2-2x+1'. So, there is 1 equation and 2 variables: 'f' and 'x'. This problem is solved entering the definition 'f(x)= x^2-2x+1' in the 'vars. & const.' box and the equation to resolve, for example, 'f(x)=0' or y=0, in the 'expressions' box.

Another common error is to employ the letter 'e' as a variable name. Letter 'e' is reserved for Napier's constant (also called Euler's number) 2.718… So, for the parser, the expression (a+e+f)^2 is the same as (a+2.718…+f)^2, were 'e' represents a number and not a variable. A solution would be to use another letter, or prefix with an underscore: (a+_e+f)^2.

1.3 SPACES

Version 8.3.16 or later ignore spaces but not carriage return unless otherwise checked the option 'ignore CR'

1.4 DECIMAL, HEXADECIMAL, OCTAL AND BINARY INPUT

Hexadecimal, octal and/or binary numbers with the dot (.) can be entered by prefixing &h, &o or &b, respectively. When one of the following &h, &o, &b or &d followed by a number, respectively, in hexadecimal, octal, binary or decimal base is found in an expression, the prefix overrides the previous base: 10+&hf+10 gives 41 at the output (decimal: 10+15+16=41) as result (not 35). To restore initial default decimal base &d should be prefixed: 10 + &hf + &d10 = 35. So in brief, the initial default base is decimal until a new base is found and, this last one prevails until a new base prefix overrides. Entering &h0.8 will produce the decimal result 0.5 (= ½)

1.5 Functions, variables and constants

Predefined functions (sin(), cos(), sinh(), exp() …) and constants (e and pi) are already present in the calculator.

A variable, custom function or constant name can be one alphabetic letter, two letters or a letter and one number. Names of more characters must be prefixed by an underscore '_':

Vf=3, Vi=2, _pi2=PI^2, …

To allow two character name, or one character and a number, the required options should be checked.

1.5.1 Predefined functions

Trigonometric (sin, cos, …) functions' argument are expected to be expressed in radians. The default base is radians and may be changed prefixing &deg (for degrees) or &grad (for gradians) to the angles. To restore again radians, prefix &rad to the angle. Example:

For an input "sin(&deg 90) – cos(90) + sin(&grad 100) – cos(100) + sin(&rad PI/2) – cos(pi/2)" the result will be 3 (= 1 – 0 + 1 – 0 + 1 – 0)

Squared sinus function may be written as sin()^2 and also as sin2(); for the math parser both forms are equivalent. This stands for all predefined functions: for example: exp2(2) - exp(2)^2 will give a zero output. So log2(10) is the same as log(10)^2, ln2(10) or ln(10)^2. For logarithm in base 2, there is the function logtwo() and in base 10, logten(). By this means, it is possible to power any function up to power 9: exp3(pi) is equivalent to exp(pi)^3 and logtwo8(2) is equal to logtwo(2)^8.

In versions v8.3.55 and later, 'cos pi' is understood as cos(pi). There is no need to say that cos pi x will become cos(pi)*x and never cos(pi*x). As in previous versions cos2(...) is equivalent to cos(...)^2, but 'cos3 x' equals cos(3)*x.

1.5.2 Custom functions

User may define custom functions by entering the function in the box for this purpose. A custom function may be P(x)=x^2-x+1 or _multiFn(x,y,z)=x^2+y^2+z^2 (one or several arguments, respectively) using a comma ',' or a column separator character, between the arguments.

1.6 Variables values and constants

Variables values should be defined in the same box disposed for custom functions.

Variables values are x=2, _halfPI=pi/2

It is possible to define as many constants as needed in the same box just as variables values. Definitions may be separated one from another by a carriage return character ('CR' or 'enter') or by a pipe ('|' or vertical bar) character.

1.7 Equations and System of Equations

The calculator tries to solve equations: x2-x+1=0, sin2(x)-0.5=0.2 or tan(x)=-2 are equations.

System of Equations should be entered separating one equation from another by a row delimiter, i.e., a carriage return character ('CR' or 'enter') or by a pipe ('|' or vertical bar) character. For example: entering the system x+y=2|x-y=1 will give x=3/2, y=1/2 as output.

1.8 Interpolation

1.8.1 Entering the points.

Given the points (x_j,f(x_j)), viz, (-2,1), (0,-1) and (2,1) the input could be

lagrangianinterpolation(-2,1|0,-1|2,1) or

```
lagrangianinterpolation(
-2,1
0,-1
2,1)
```

i.e., separating the coordinates by column delimiters and separating the points by row delimiters.

1.8.2 Entering an interval limits, the polynomail degree and the function to approximate.

Given an interval [a,b], the desired polynomial degree 'n', and the function f(x) to approximate, substituting a,b,n and f(x) in lagrangianinterpolation(a,b,n,f(x)) by concrete values, for example:

lagrangianinterpolation(-1,1,10,abs(x))

will result, at the output, in a polynomial of degree equal to 10, approximating the function abs(x), absolute value of x, at the interval [-1,1]. For this same example, a set of 11 points x_i, solutions of the Chebyshev Polinomial with degree 11, conform 10 subintervals at [-1,1]; then, abs(xi), 0<i<10, is calculated and finally the Lagragian polynomial. If the 'detail' option is checked, the output will show the intermediate steps.

More info at, for example:

http://svn.assembla.com/svn/mna/tps/tp4ej5Paulina.pdf (in Spanish)

1.9 Derivatives

Derivatives follow the Euler's notation: entering D2x(x^2*y) will produce an output equal to 2.

1.10 Integration

Integrals can be obtained writing the function name as

integral(sin(x))dx - or - ∫(sin(x))dx The Unicode value

for '∫' is 8747.

Limits for definite integrals should be separated by a column delimiter, so for example:

∫(0,pi,sin(x))dx will integrate the sinus function in the interval [0,pi]

1.11 Matrices

1.11.1 Column delimiters

Characters comma, semicolon or tabulation {",",";",Ascii 9)} separate matrices' columns.

1.11.2 Row delimiters

Carriage return character ('CR' or 'enter') or pipe ('|' or vertical bar) character separate rows.

1.11.3 Example

For instance, the input:

```
1,234 , -1
5,432 , 4
+
-6, 7
8, 9
```

will give as result:

```
1228 6
5440 13
```

Tip: when delimiting columns inside a matrix, use " , ", this is, a comma surrounded by spaces, otherwise use semicolons as in:

```
1,234;-1
5,432;4
+
-6;7
8;9
```

Also, the same but replacing line feeds by pipe character: 1,234;-1|5,432;4|+|-6;7 |8;9

As expected, returns:

```
1228 6 5440 13
```

2 FORMATTING THE OUTPUT

2.1 ROUNDING.

When rounding option is selected, numeric values will be rounded, on the output, up to 3 decimals. Examples are 3.123e-6, 3.142 or 2.718.

2.2 ENGINEERING VERSUS GENERAL NOTATION.

Engineering notation, 'Eng', outputs using exponents multiples of 3, i.e., 1e3, 1e-3, 1e6, or 1e-6 come up with a maximum of 3 decimals when rounding, whilst general notation does not. For an input of exp(1)*1e6, being 'Eng' notation selected, the result will be 2.718e6; in case 'Eng' is not selected, the output will be 2718281.828. If no rounding was selected the output would be 2.71828182845905e6 and 2718281.82845904, respectively.

2.3 FRACTIONS.

Fractions option will try to produce fractions in the output, although sometimes is not convinient due to the numbers' nature (numerator and/or denominator too large or small, or precision loss).

2.4 DECIMAL, HEXADECIMAL, OCTAL AND BINARY OUTPUT

Decimal output is always carried out. Selecting options for other bases brings output in the selected bases. Outputs in hexadecimal, octal and binary base are prefixed by &h, &o and &b, respectively.

2.5 DETAILED INFO

The 'detail' option, if available, will append intermediate steps to the output.