DERIVATIVE

PURPOSE

Compute the derivative of a function.

DESCRIPTION

DATAPLOT computes and prints the analytic form of the derivative function. It can either return the derivative function as a new function or it can return the derivative evaluated at one or more points. In either case, the derivative is first evaluated symbolically.

SYNTAX 1

LET <resp> = DERIVATIVE <function> WRT <var>
where <function> is the name of a previously defined function or a functional expression;
<var> is the name of the variable with respect to which the derivative is taken;
<resp> is a variable of the same length as <var> where the evaluated derivatives are stored.

With this syntax, the derivative variable (<var>) must be defined for one or more points. The derivative is evaluated at each of these points and the resulting value is put in the corresponding element of <resp>. The analytic derivative function is printed but not saved in a function that can be used later.

SYNTAX 2

LET <resp> = DERIVATIVE <function> WRT <var> FOR <var> = <value>
where <function> is the name of a previously defined function or a functional expression;
<var> is the name of the variable with respect to which the derivative is taken;
<value> is a number or parameter at which the derivative is evaluated;
<resp> is a parameter where the evaluated derivatives are stored.

This syntax is similar to SYNTAX 1. However, the FOR clause identifies a single point at which the derivative is to be evaluated.

SYNTAX 3

LET FUNCTION <d1> = DERIVATIVE <function> WRT <var>
where <function> is the name of a previously defined function or a functional expression;
<var> is the name of the variable with respect to which the derivative is taken;
<d1> is the name of a function where the computed derivative function is stored.

With this syntax, the analytic derivative is saved in a new function (<d1>) which can be used later in the same way as any other function. To evaluate the derivative at specific points, simply evaluate the function in the standard way. For example,

```
LET FUNCTION F1 = X**2 + 3*X -5
LET X = SEQUENCE 0 0.1 10
LET Y1 = F1
```

The variable Y1 contains the values of F1 evaluated at the given points of X.

EXAMPLES

```
LET XDERV = DERIVATIVE 3*X**2 -8*X + 4 WRT X
LET FUNCTION D1 = DERIVATIVE F1 WRT X
LET XDERV = DERIVATIVE 3*X**2 - 8*X + 4 WRT X FOR X = 2.3
```

NOTE 1

DATAPLOT can take derivatives of functions containing combinations of the following types of functions:

1. Arithmetic operations (i.e., +, -, *, /, and **). This includes combinations of the these operations (e.g., a polynomial function).
2. All the built-in trigonometric functions (including the inverse trigonometric functions).
3. All the built-in hyperbolic trigonometric functions (including the inverse hyperbolic trigonometric functions).
4. The LOG (LN is not recognized) and LOG10 functions for natural and base 10 logarithms respectively. Logarithms in other bases should be expressed as a function of base 10 logarithms (e.g., LOG (base 2) 30 = LOG (base 10) 30/ LOG (base 10) 2).
5. The SQRT function.
6. The EXP function.

Even if it is limited to the above types of functions, the differentiation may fail if the function gets too complicated.

The other built-in functions are not recognized directly. However, some of them can be handled by defining a user function in terms of functions that DATAPLOT does recognize.

NOTE 2

DATAPLOT only calculates first order derivatives. To calculate higher order derivatives, use the third syntax to return the first derivative as a function. Then take the derivative of this function to get the second order derivative. This can be repeated as many times as needed (although going beyond second order derivatives is rare).

NOTE 3

DATAPLOT only takes derivatives with respect to a single variable. That is, it does not take partial derivatives. The following syntax will NOT work:

```
LET FUNCTION F = 4*X**2*Y**3 + 3*X + 4*Y
LET FUNCTION D = DERIVATIVE F WRT X
```

DEFAULT

None

SYNONYMS

None

RELATED COMMANDS

INTEGRAL	=	Compute the integral of a function.
ROOTS	=	Compute the roots of a function.
RUNGE KUTTA	=	Runge Kutta differential equation solver.
INTERPOLATE	=	Interpolate a function.

REFERENCE

Consult any standard Calculus textbook.

APPLICATIONS

Mathematics

IMPLEMENTATION DATE

Pre-1987

PROGRAM

```
LET FUNCTION F1 = SIN(X)*COS(X)
LET START = -PI/2
LET STOP = PI/2
LET X = SEQUENCE START 0.1 STOP
LET D1 = DERIVATIVE F1 WRT X
LINE SOLID DASH
LET Y1 = F1
TITLE PLOT OF FUNCTION AND ITS DERIVATIVE
PLOT Y1 D1 VS X
```