RUNGE KUTTA

PURPOSE

Solve first and second order differential equations via Runge Kutta methods. It solves initial value problems.

DESCRIPTION

Differential equations are those which involve a relation between derivatives. For example, the equation

$$y = x^2$$

can be expressed as the differential equation

$$dy/dx = 2x$$

In this case, instead of giving a value of y for each value of x, the differential equation gives the slope at each value of x.

The following should be noted:

1. The term ordinary refers to the fact that only ordinary derivatives are involved (that is, no partial derivatives). DATAPLOT does not currently support any methods for solving partial differential equations.
2. The dependent variable is the variable being differentiated and the independent variable is the variable with respect to which the differentiations are taken. In the example dy/dx = 2x, x is the independent variable and y is the dependent variable.
3. The order is the order of the highest derivative. Currently, the RUNGE KUTTA command is limited to first and second order differential equations. First order differential equations are often expressed as:

   $$dy/dx = F(x,y)$$

   while second order differential equations are often expressed as:

   $$d^2y/dx = G(x, y, dy/dx)$$

   Higher order differential equations can be expressed in a similar manner.
4. The degree is the power to which the highest order derivative is raised. The RUNGE KUTTA command only deals with first degree differential equations.
5. An analytic solution is an equation of the form

   $$y = F(x)$$

   which results in the original differential equation when it is differentiated. Although the RUNGE KUTTA command does not provide an analytic solution (that is, the function F), it can generate the y value for any corresponding x value. The RUNGE KUTTA command returns y values (and dy/dx values for the second order case) for a given set of X values.
6. Differential equations are complicated by the fact that there is an infinite number of solutions (this is due to the fact that the derivative of any constant is zero). However, if the solution is required to pass through a specific point (x0,y0), then the solution is unique. This point is referred to as an initial value. Boundary value problems are those in which more than one initial point is specified. The RUNGE KUTTA command can solve initial value problems, but it does not solve boundary value problems. DATAPLOT does not currently provide any methods for solving boundary value problems.

SYNTAX 1

LET <y> = RUNGE KUTTA <f> <x1> <SUBSET/EXCEPT/FOR qualification>

where <f> is a function or functional expression that defines the differential equation (this function often contains <y>);
<x1> is a variable that defines the desired grid of X values;
<y> is a variable where the computed values for the dependent variable are saved (the first element of this variable must be previously defined);
and where the <SUBSET/EXCEPT/FOR qualification> is optional and rarely used in this context.

This syntax is used for first order differential equations. The initial value (x0, y0) is specified by the first elements of <x1> and <y> respectively. These points must be specified before entering the RUNGE KUTTA command. The solution is the set of points <y> and <x1>.

SYNTAX 2

LET <y> <yd> = RUNGE KUTTA <f> <x1> <SUBSET/EXCEPT/FOR qualification>

where <f> is a function or functional expression that defines the differential equation (this function often contains <y> and <yd>);
<x1> is a variable that defines the desired grid of X values;

<y> is a variable where the computed values for the dependent variable are saved (the first element of this variable must be previously defined);

<yd> is a variable where the computed values for the dy/dx term are saved (the first element of this variable must be previously defined);

and where the <SUBSET/EXCEPT/FOR qualification> is optional and rarely used in this context.

This syntax is used for second order differential equations. The initial value (x0, y0, yd0) is specified by the first elements of <x1>, <y>, and <yd> respectively. These points must be specified before entering the RUNGE KUTTA command. The solution is the set of points <y>, <yd>, and <x1>.

EXAMPLES

LET Y = RUNGE KUTTA 2*X X1
LET Y = RUNGE KUTTA F X1
LET Y YD = RUNGE KUTTA F X1

NOTE

DATAPLOT uses a fourth order Runge-Kutta algorithm. This algorithm is documented in the Numerical Recipes book. It is also covered in most numerical analysis textbooks. Runge-Kutta methods have the advantage of almost always succeeding, but they do not always have as high an accuracy or the computational efficiency as more sophisticated methods such as Bulirsch-Stoer or predictor-corrector methods.

Runge-Kutta methods are not appropriate for stiff differential equations.

DEFAULT

None

SYNONYMS

None

RELATED COMMANDS

DERIVATIVE	=	Compute the derivative of a function.
INTEGRAL	=	Compute the integral of a function.
ROOTS	=	Compute the roots of a function.
INTERPOLATE	=	Interpolate a function.

REFERENCE

“Numerical Recipes: The Art of Scientific Computing (FORTRAN Version),” Press, Flannery, Teukolsky, and Vetterling. Cambridge University Press, 1989 (chapter 14).

APPLICATIONS

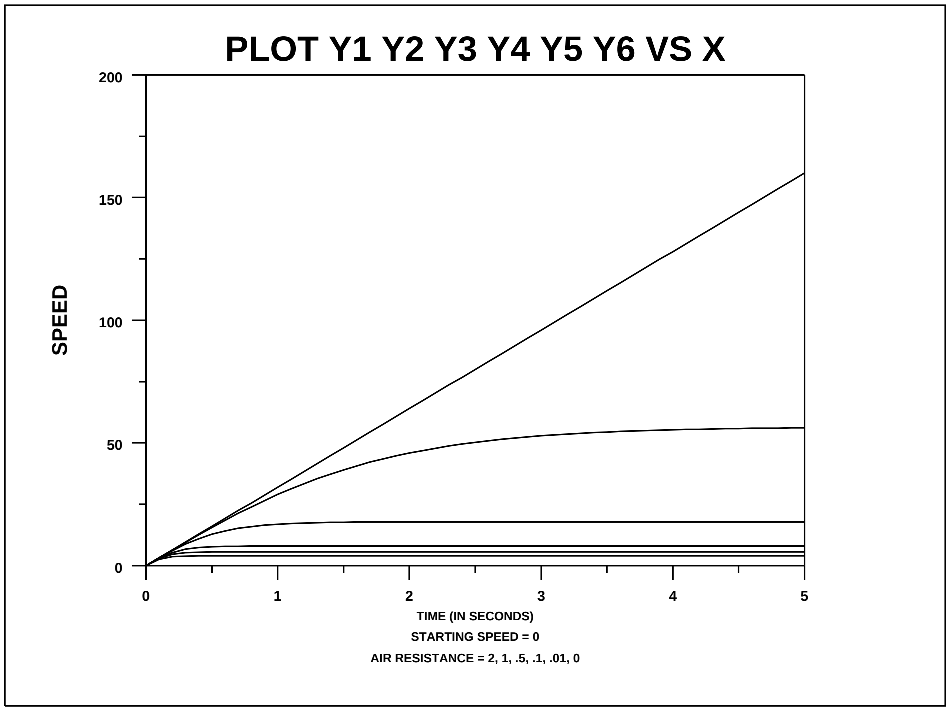
Differential Equations

IMPLEMENTATION DATE

87/10

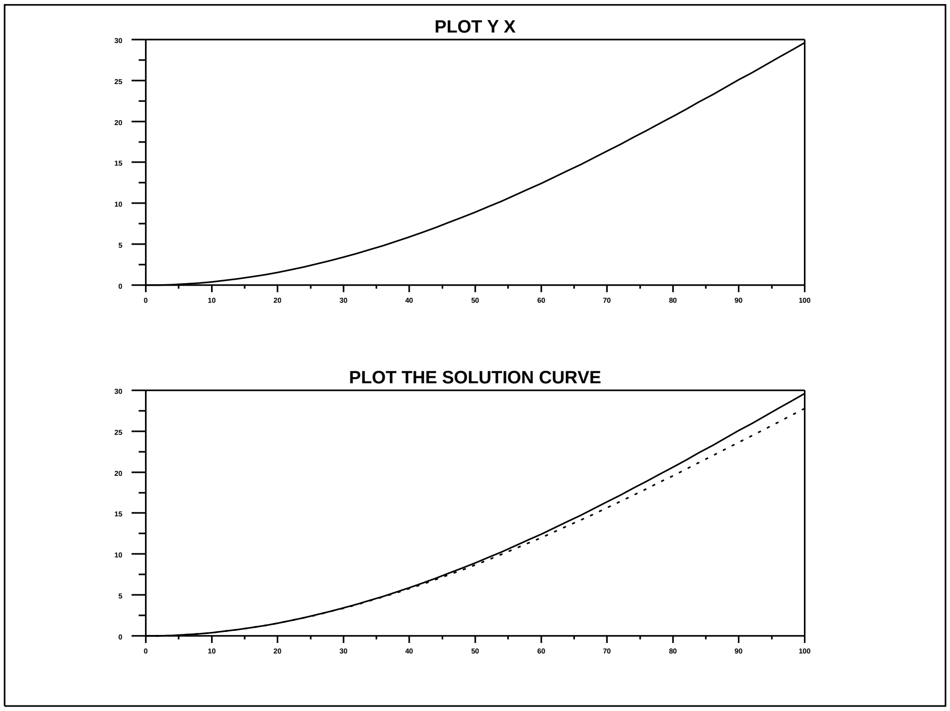
PROGRAM 1 (first order example)

```
. THIS IS THE DATAPLOT PROGRAM FILE RAIN.DP
. PURPOSE--DETERMINE THE EQUATION OF MOTION OF A RAINDROP
. WITH SEVERAL DIFFERENT AIR RESISTANCE VALUES.
. ANALYSIS TECHNIQUE--RUNGE-KUTTA SOLUTION TO DIFFERENTIAL EQUATION.
. NOTE--Y = SPEED AFTER TIME X
. X = TIME (IN SECONDS)
. M = MASS
. C = AIR RESISTANCE
. -----START POINT-----------------------------------
. STEP 1--DEFINE THE RIGHT-HAND SIDE FUNCTION OF Y' = F
. DEFINE THE DESIRED GRID OF X VALUES, INITIAL CONDITION, AND
. A TABLE OF VALUES FOR C
LET FUNCTION F = 32-(C/M)*Y**2
LET X = 0  .1  5
LET Y(1)=0
LET M = 1
LET CTAB = DATA 2  1  .5  .1  .01  0
. STEP 2--SOLVE THE DIFFERENTIAL EQUATION FOR EACH VALUE OF C
LOOP FOR K = 1 1 6
     LET C = CTAB(K)
     LET Y = RUNGE-KUTTA F X
     LET Y^K = Y
END OF LOOP
. STEP 3--PLOT OUT THE 6 TRACES
Y1LABEL SPEED; XLABEL TIME (IN SECONDS)
X2LABEL STARTING SPEED = 0
X3LABEL AIR RESISTANCE = 2, 1, .5, .1, .01, 0
PLOT Y1 Y2 Y3 Y4 Y5 Y6 VS X
```



PROGRAM 2 (second order example)

```
. THIS IS THE DATAPLOT PROGRAM FILE BEAM.DP
. PURPOSE--DETERMINE DEFLECTION VS DISTANCE TRACE OF ELASTIC BEAM
. ANCHORED AT ONE END AND WITH A LOAD AT THE OTHER END
. (CANTILEVER).
. TECHNIQUE--RUNGE-KUTTA SOLUTION OF 2ND ORDER DIFFERENTIAL EQUATION
. APPLICATION--STRUCTURAL ENGINEERING
. SOURCE--DORN, WILLIAM S. AND MCCRACKEN, DANIEL D., NUMERICAL METHODS
. WITH FORTRAN IV CASE STUDIES, JOHN WILEY AND SONS, INC.,
. NEW YORK, 1972, PAGES 391-401.
. TO FIND--DEFLECTION VALUE FOR VARIOUS DISTANCES OUT ONTO THE BEAM
. NOTE--THE ELASTIC DEFLECTION SATISFIES THE DIFFERENTIAL EQUATION--
. Y’’ / (1 + (Y’)**2 )**(3/2) = P*(L-X) / (E*I)
. Y’’ = P*(L-X) / (E*I) * (1 + (Y’)**2 )**(3/2)
. WITH Y(0) = 0 AND Y’(0) = 0
. WHERE
. Y = DEFLECTION FROM HORIZONTAL (IN INCHES)
. X = DISTANCE OUT ONTO BEAM (X = 0, 2, 4, 6, ..., 100 INCHES)
. P = LOAD (IN POUNDS) AT FREE END (HERE = 64)
. L = LENGTH (IN INCHES) OF BEAM (HERE = 100)
. E = YOUNG’S MODULUS (= 6 * 10**6)
. I = MOMENT OF INERTIA OF CROSS SECTION (HERE = 0.128 INCHES**4)
.
. -----START POINT-----------------------------------
.
. STEP 1--DEFINE THE PHYSICAL PARAMETERS OF THE BEAM
LET P = 64
LET L = 100
LET E = 6*10**6
LET I = 0.128
. STEP 2--DEFINE THE RIGHT-SIDE FUNCTION OF Y’’(X) = F
LET FUNCTION F1 = P*(L-X)/(E*I)
LET FUNCTION F2 = (1+YP**2)**(3/2)
LET FUNCTION F = F1*F2
. STEP 3--DEFINE INITIAL CONDITIONS AND DEFINE THE DESIRED SEQUENCE
. OF POINTS AT WHICH TO COMPUTE THE SOLUTION CURVE.
LET Y(1) = 0
LET YP(1) = 0
LET X = SEQUENCE 0 2 100
. STEP 4--SOLVE THE DIFFERENTIAL EQUATION
LET Y YP = RUNGA-KUTTA F X
MULTIPLOT 2 1; MULTIPLOT CORNER COORDINATES 0 0 100 100
TITLE AUTOMATIC
PLOT Y X
PRINT X Y YP
. STEP 5--PLOT OUT THE SOLUTION CURVE
LET YBE = (P/(6*E*I))*(X**2)*(3*L-X)
TITLE PLOT THE SOLUTION CURVE
LINES SOLID DOTTED
PLOT Y YBE VS X
END OF MULTIPLOT
```



PLOT Y X

PLOT THE SOLUTION CURVE