Chapter 16
Differential Equations

In this Chapter we present examples of solving ordinary differential equations (ODE) using calculator functions. A differential equation is an equation involving derivatives of the independent variable. In most cases, we seek the dependent function that satisfies the differential equation.

Basic operations with differential equations

In this section we present some uses of the calculator for entering, checking and visualizing the solution of ODEs.

Entering differential equations

The key to using differential equations in the calculator is typing in the derivatives in the equation. The easiest way to enter a differential equation is to type it in the equation writer. For example, to type the following ODE: (x-1)·(dy(x)/dx)^2 + 2·x·y(x) = e^x sin x, use:

[↱] EQW [↰] () [ALPHA] [↰] [X] [−] [1] [▶] [▶] [▶] [×] [↱] ∂ [ALPHA] [↰] [X]
[▶] [ALPHA] [↰] [Y] [↰] () [ALPHA] [↰] [X] [▶] [▶] [Y^x] [2] [▶] [▶] [+] [2] [×]
[ALPHA] [↰] [X] [×] [ALPHA] [↰] [Y] [↰] () [ALPHA] [↰] [X] [▶] [▶] [▶] [▶]
[↱] = [↰] e^x [ALPHA] [↰] [X] [▶] [×] [SIN] [ALPHA] [↰] [X] [ENTER]

The derivative dy/dx is represented by `∂x(y(x)) or by d1y(x)`. For solution or calculation purposes, you need to specify y(x) in the expression, i.e., the dependent variable must include its independent variable(s) in any derivative in the equation.

You can also type an equation directly into the stack by using the symbol ∂ in the derivatives. For example, to type the following ODE involving second-order derivatives: d^{2}u(x)/dx^2 + 3u(x)·(du(x)/dx) + u(x)^2 = 1/x, directly into the screen, use:

['] [↱] ∂ [ALPHA] [↰] [X] [↰] () [↱] ∂ [ALPHA] [↰] [X] [↰] () [ALPHA] [↰] [U]
[↰] () [ALPHA] [↰] [X] [▶] [▶] [▶] [+] [3] [×] [ALPHA] [↰] [U] [↰] ()
[ALPHA] [↰] [X] [▶] [×] [↱] ∂ [ALPHA] [↰] [X] [↰] () [ALPHA] [↰] [U] [↰] ()
[ALPHA] [↰] [X] [▶] [▶] [+] [ALPHA] [↰] [U] [↰] () [ALPHA] [↰] [X] [▶] [Y^x] [2]
[↱] = [1] [÷] [ALPHA] [↰] [X] [ENTER]

The result is ‘`∂x(∂x(u(x)))+3*u(x)*∂x(u(x))+u^2=1/x`’. This format shows up in the screen when the _Textbook option in the display setting

(MODE DEF) is not selected. Press ▼ to see the equation in the Equation Writer.

An alternative notation for derivatives typed directly in the stack is to use 'd1' for the derivative with respect to the first independent variable, 'd2' for the derivative with respect to the second independent variable, etc. A second-order derivative, e.g., d^2x/dt^2, where x = x(t), would be written as 'd1d1x(t)', while $(dx/dt)^2$ would be written 'd1x(t)^2'. Thus, the PDE $\partial^2y/\partial t^2 - g(x,y)\cdot(\partial^2y/\partial x^2)^2 = r(x,y)$, would be written, using this notation, as 'd2d2y(x,t)-g(x,y)*d1d1y(x,t)^2=r(x,y)'.

The notation using 'd' and the order of the independent variable is the notation preferred by the calculator when derivatives are involved in a calculation. For example, using function DERIV, in ALG mode, as shown next DERIV('x*f(x,t)+g(t,y) = h(x,y,t)',t), produces the following expression: `'x*d2f(x,t)+d1g(t,y)=d3h(x,y,t)'`. Translated to paper, this expression represents the partial differential equation $x\cdot(\partial f/\partial t) + \partial g/\partial t = \partial h/\partial t$.

Because the order of the variable t is different in f(x,t), g(t,y), and h(x,y,t), derivatives with respect to t have different indices, i.e., d2f(x,t), d1g(t,y), and d3h(x,y,t). All of them, however, represent derivatives with respect to the same variable.

Expressions for derivatives using the order-of-variable index notation do not translate into derivative notation in the equation writer, as you can check by pressing ▼ while the last result is in stack level 1. However, the calculator understands both notations and operates accordingly regarding of the notation used.

Checking solutions in the calculator

To check if a function satisfy a certain equation using the calculator, use function SUBST (see Chapter 5) to replace the solution in the form 'y = f(x)' or 'y = f(x,t)', etc., into the differential equation. You may need to simplify the result by using function EVAL to verify the solution. For example, to check that $u = A \sin \omega_o t$ is a solution of the equation $d^2u/dt^2 + \omega_o^2\cdot u = 0$, use the following:

In ALG mode:

SUBST('∂t(∂t(u(t)))+ ω0^2*u(t) = 0','u(t)=A*SIN (ω0*t)') ENTER

EVAL(ANS(1)) ENTER

In RPN mode:

‘∂t(∂t(u(t)))+ ω0^2*u(t) = 0’ ENTER ‘u(t)=A*SIN (ω0*t)’ ENTER

SUBST EVAL

The result is ‘0=0’.

For this example, you could also use: ‘∂t(∂t(u(t))))+ ω0^2*u(t) = 0’ to enter the differential equation.

Slope field visualization of solutions

Slope field plots, introduced in Chapter 12, are used to visualize the solutions to a differential equation of the form dy/dx = f(x,y). A slope field plot shows a number of segments tangential to the solution curves, y = f(x). The slope of the segments at any point (x,y) is given by dy/dx = f(x,y), evaluated at any point (x,y), represents the slope of the tangent line at point (x,y).

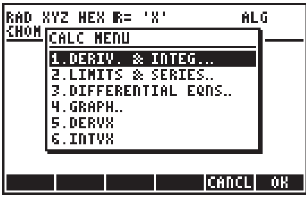
Example 1 -- Trace the solution to the differential equation y’ = f(x,y) = sin x cos y, using a slope field plot. To solve this problem, follow the instructions in Chapter 12 for *slopefield* plots.

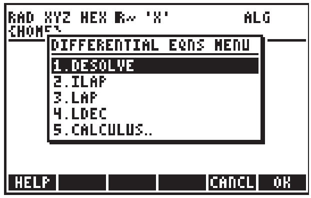
If you could reproduce the slope field plot in paper, you can trace by hand lines that are tangent to the line segments shown in the plot. This lines constitute lines of y(x,y) = constant, for the solution of y’ = f(x,y). Thus, slope fields are useful tools for visualizing particularly difficult equations to solve.

In summary, slope fields are graphical aids to sketch the curves y = g(x) that correspond to solutions of the differential equation dy/dx = f(x,y).

The CALC/DIFF menu

The DIFFERENTIAL EQNS.. sub-menu within the CALC (← CALC) menu provides functions for the solution of differential equations. The menu is listed below with system flag 117 set to CHOOSE boxes:





These functions are briefly described next. They will be described in more detail in later parts of this Chapter.

DESOLVE: Differential Equation SOLVEr, provides a solution if possible

ILAP: Inverse LAPlace transform, $L^{-1}[F(s)] = f(t)$

LAP: LAPlace transform, $L[f(t)]=F(s)$

LDEC: solves Linear Differential Equations with Constant coefficients, including systems of differential equations with constant coefficients

Solution to linear and non-linear equations

An equation in which the dependent variable and all its pertinent derivatives are of the first degree is referred to as a linear differential equation. Otherwise, the equation is said to be non-linear. Examples of linear differential equations are: $d^2x/dt^2 + \beta\cdot(dx/dt) + \omega_o\cdot x = A \sin \omega_f t$, and $\partial C/\partial t + u\cdot(\partial C/\partial x) = D\cdot(\partial^2 C/\partial x^2)$.

An equation whose right-hand side (not involving the function or its derivatives) is equal to zero is called a homogeneous equation. Otherwise, it is called non-homogeneous. The solution to the homogeneous equation is known as a general solution. A particular solution is one that satisfies the non-homogeneous equation.

Function LDEC

The calculator provides function LDEC (Linear Differential Equation Command) to find the general solution to a linear ODE of any order with constant coefficients, whether it is homogeneous or not. This function requires you to provide two pieces of input:

- the right-hand side of the ODE
- the characteristic equation of the ODE

Both of these inputs must be given in terms of the default independent variable for the calculator's CAS (typically 'X'). The output from the function is the general solution of the ODE. The function LDEC is available through in the CALC/DIFF menu. The examples are shown in the RPN mode, however, translating them to the ALG mode is straightforward.

Example 1 – To solve the homogeneous ODE: $d^3y/dx^3-4\cdot(d^2y/dx^2)-11\cdot(dy/dx)+30\cdot y = 0$, enter: `0` ENTER `'X^3-4*X^2-11*X+30'` ENTER `LDEC` EVAL. The solution is:

$$\frac{(120\cdot cC0+16\cdot cC1-8\cdot cC2)\cdot e^{3\cdot X}\cdot e^{2\cdot X}-\left((30\cdot cC0-(5\cdot cC1+5\cdot cC2))\cdot e^{5\cdot X}\cdot e^{3\cdot X}-(30\cdot cC0-(21\cdot cC1-3\cdot cC2))\right)}{120\cdot e^{3\cdot X}}$$

where cC0, cC1, and cC2 are constants of integration. While this result seems very complicated, it can be simplified if we take

K1 = (10*cC0-(7+cC1-cC2))/40, K2 = -(6*cC0-(cC1+cC2))/24,

and

K3 = (15*cC0+(2*cC1-cC2))/15.

Then, the solution is

$$y = K_1\cdot e^{-3x} + K_2\cdot e^{5x} + K_3\cdot e^{2x}.$$

The reason why the result provided by LDEC shows such complicated combination of constants is because, internally, to produce the solution, LDEC utilizes Laplace transforms (to be presented later in this chapter), which transform the solution of an ODE into an algebraic solution. The combination of constants result from factoring out the exponential terms after the Laplace transform solution is obtained.

Example 2 – Using the function LDEC, solve the non-homogeneous ODE:

$$d^3y/dx^3-4\cdot(d^2y/dx^2)-11\cdot(dy/dx)+30\cdot y = x^2.$$

Enter:

`'X^2'` ENTER `'X^3-4*X^2-11*X+30'` ENTER `LDEC` EVAL

The solution, shown partially here in the Equation Writer, is:

$$\frac{(27000\cdot cC0+3600\cdot cC1-(1800\cdot cC2+450))\cdot e^{3\cdot X}\cdot e^{2\cdot X}-\left[\left((6750\cdot cC0-(1125\cdot cC1+1125\cdot cC2+18))\cdot e^{5\cdot X}-\left(900\cdot X^2+660\cdot X+482\right)\right)\cdot e^{3\cdot X}-\left(6750\cdot cC0-(4725\cdot cC1-(675\cdot cC2-50))\right)\right]}{27000\cdot e^{3\cdot X}}$$

Replacing the combination of constants accompanying the exponential terms with simpler values, the expression can be simplified to $y = K_1 \cdot e^{-3x} + K_2 \cdot e^{5x} + K_3 \cdot e^{2x} + (450 \cdot x^2+330 \cdot x+241)/13500$.

We recognize the first three terms as the general solution of the homogeneous equation (see Example 1, above). If y_h represents the solution to the homogeneous equation, i.e., $y_h = K_1 \cdot e^{-3x} + K_2 \cdot e^{5x} + K_3 \cdot e^{2x}$. You can prove that the remaining terms in the solution shown above, i.e., $y_p = (450 \cdot x^2+330 \cdot x+241)/13500$, constitute a particular solution of the ODE.

Note: This result is general for all non-homogeneous linear ODEs, i.e., given the solution of the homogeneous equation, $y_h(x)$, the solution of the corresponding non-homogeneous equation, $y(x)$, can be written as

$$y(x) = y_h(x) + y_p(x),$$

where $y_p(x)$ is a particular solution to the ODE.

To verify that $y_p = (450 \cdot x^2+330 \cdot x+241)/13500$, is indeed a particular solution of the ODE, use the following:

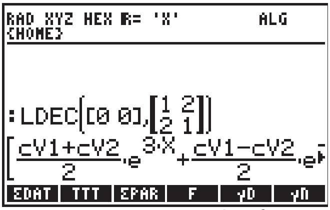
```
'd1d1d1Y(X)-4*d1d1Y(X)-11*d1Y(X)+30*Y(X) = X^2' ENTER
'Y(X)=(450*X^2+330*X+241)/13500' ENTER
SUBST EVAL
```

Allow the calculator about ten seconds to produce the result: ‘X^2 = X^2’.
Example 3 - Solving a system of linear differential equations with constant coefficients.
Consider the system of linear differential equations:

$$x_1'(t) + 2x_2'(t) = 0,$$

$$2x_1'(t) + x_2'(t) = 0.$$

In algebraic form, this is written as: $\mathbf{A}\cdot\mathbf{x}'(t) = 0$, where $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$. The system can be solved by using function LDEC with arguments [0,0] and matrix A, as shown in the following screen using ALG mode:



The solution is given as a vector containing the functions [$x_1(t)$, $x_2(t)$]. Pressing ▼ will trigger the Matrix Writer allowing the user to see the two components of the vector. To see all the details of each component, press the EDIT soft menu key. Verify that the components are:

Function DESOLVE

The calculator provides function DESOLVE (Differential Equation SOLVEr) to solve certain types of differential equations. The function requires as input the differential equation and the unknown function, and returns the solution to the equation if available. You can also provide a vector containing the differential equation and the initial conditions, instead of only a differential equation, as input to DESOLVE. The function DESOLVE is available in the CALC/DIFF menu. Examples of DESOLVE applications are shown below using RPN mode.

Example 1 – Solve the first-order ODE:

$$dy/dx + x^2\cdot y(x) = 5.$$

In the calculator use:

```
'd1y(x)+x^2*y(x)=5' ENTER 'y(x)' ENTER DESOLVE
```

The solution provided is

{'y = (INT(5*EXP(xt^3/3),xt,x)+cC0)*1/EXP(x^3/3)}' }, i.e.,

$$y(x) = \exp(-x^3/3) \cdot \left(\int 5 \cdot \exp(x^3/3) \cdot dx + cC_0 \right)$$

The variable ODETYPE

You will notice in the soft-menu key labels a new variable called ODETY (ODETYPE). This variable is produced with the call to the DESOL function and holds a string showing the type of ODE used as input for DESOLVE. Press ODETY to obtain the string "`1st order linear`".

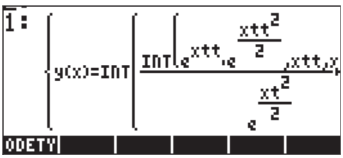
<u>Example 2</u> -- Solve the second-order ODE:

$$d^2y/dx^2 + x\,(dy/dx) = \exp(x).$$

In the calculator use:

`'d1d1y(x)+x*d1y(x) = EXP(x)'` ENTER `'y(x)'` ENTER `DESOLVE`

The result is an expression having two implicit integrations, namely,



For this particular equation, however, we realize that the left-hand side of the equation represents d/dx(x dy/dx), thus, the ODE is now written:

$$d/dx(x\ dy/dx) = \exp x,$$

and

$$x\ dy/dx = \exp x + C.$$

Next, we can write

$$dy/dx = (C + \exp x)/x = C/x + e^x/x.$$

In the calculator, you may try to integrate:

`'d1y(x) = (C + EXP(x))/x'` ENTER `'y(x)'` ENTER `DESOLVE`

The result is { 'y(x) = INT((EXP(xt)+C)/xt,xt,x)+C0' }, i.e.,

$$y(x) = \int \cdot \frac{e^x + C}{x} dx + C_0$$

Performing the integration by hand, we can only get it as far as:

$$y(x) = \int \cdot \frac{e^x}{x} dx + C \cdot \ln x + C_0$$

because the integral of exp(x)/x is not available in closed form.

<u>Example 3</u> – Solving an equation with initial conditions. Solve

$$d^2y/dt^2 + 5y = 2 \cos(t/2),$$

with initial conditions

$$y(0) = 1.2, \; y'(0) = -0.5.$$

In the calculator, use:

['d1d1y(t)+5*y(t) = 2*COS(t/2)' 'y(0) = 6/5' 'd1y(0) = -1/2'] `ENTER`
'y(t)' `ENTER`
DESOLVE

Notice that the initial conditions were changed to their Exact expressions, 'y(0) = 6/5', rather than 'y(0)=1.2', and 'd1y(0) = -1/2', rather than, 'd1y(0) = -0.5'. Changing to these Exact expressions facilitates the solution.

Note: To obtain fractional expressions for decimal values use function →Q (See Chapter 5).

The solution is:

Press `EVAL` `EVAL` to simplify the result to
'y(t) = -((19*√5*SIN(√5*t)-(148*COS(√5*t)+80*COS(t/2)))/190)'.

Press VAR ODETY to get the string "`Linear w/ cst coeff`" for the ODE type in this case.

Laplace Transforms

The Laplace transform of a function f(t) produces a function F(s) in the image domain that can be utilized to find the solution of a linear differential equation involving f(t) through algebraic methods. The steps involved in this application are three:

1. Use of the Laplace transform converts the linear ODE involving f(t) into an algebraic equation.
2. The unknown F(s) is solved for in the image domain through algebraic manipulation.
3. An inverse Laplace transform is used to convert the image function found in step 2 into the solution to the differential equation f(t).

Definitions

The Laplace transform for function f(t) is the function F(s) defined as

$$\mathsf{L}\{f(t)\} = F(s) = \int_0^\infty f(t)\cdot e^{-st}dt.$$

The image variable s can be, and it generally is, a complex number.

Many practical applications of Laplace transforms involve an original function f(t) where t represents time, e.g., control systems in electric or hydraulic circuits. In most cases one is interested in the system response after time t>0, thus, the definition of the Laplace transform, given above, involves an integration for values of t larger than zero.
The inverse Laplace transform maps the function F(s) onto the original function f(t) in the time domain, i.e., $L^{-1}\{F(s)\} = f(t)$.

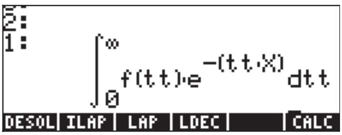
The convolution integral or convolution product of two functions f(t) and g(t), where g is shifted in time, is defined as

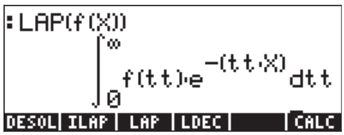
$$(f * g)(t) = \int_0^t f(u)\cdot g(t-u)\cdot du\ .$$

Laplace transform and inverses in the calculator

The calculator provides the functions LAP and ILAP to calculate the Laplace transform and the inverse Laplace transform, respectively, of a function f(VX), where VX is the CAS default independent variable, which you should set to 'X'. Thus, the calculator returns the transform or inverse transform as a function of X. The functions LAP and ILAP are available under the CALC/DIFF menu. The examples are worked out in the RPN mode, but translating them to ALG mode is straightforward. For these examples, set the CAS mode to Real and Exact.

Example 1 – You can get the definition of the Laplace transform use the following: 'f(X)' ENTER LAP in RPN mode, or LAP(f(X))in ALG mode. The calculator returns the result (RPN, left; ALG, right):





Compare these expressions with the one given earlier in the definition of the Laplace transform, i.e.,

$$\mathsf{L}\{f(t)\} = F(s) = \int_0^\infty f(t) \cdot e^{-st} dt,$$

and you will notice that the CAS default variable X in the equation writer screen replaces the variable s in this definition. Therefore, when using the function LAP you get back a function of X, which is the Laplace transform of f(X).

Example 2 – Determine the Laplace transform of f(t) = e^{2t}·sin(t). Use: 'EXP(2*X)*SIN(X)' ENTER LAP The calculator returns the result: 1/(SQ(X-2)+1). Press EVAL to obtain, 1/(X^2-4X+5).

When you translate this result in paper you would write

$$F(s) = \mathsf{L}\{e^{2t} \cdot \sin t\} = \frac{1}{s^2 - 4 \cdot s + 5}$$

Example 3 – Determine the inverse Laplace transform of F(s) = sin(s). Use: 'SIN(X)' ENTER ILAP. The calculator takes a few seconds to return the result: 'ILAP(SIN(X))', meaning that there is no closed-form expression f(t), such that f(t) = $L^{-1}\{\sin(s)\}$.

Example 4 – Determine the inverse Laplace transform of F(s) = $1/s^3$. Use: '1/X^3' ENTER ILAP EVAL. The calculator returns the result: 'X^2/2', which is interpreted as $L^{-1}\{1/s^3\} = t^2/2$.

Example 5 – Determine the Laplace transform of the function f(t) = cos (a·t+b). Use: 'COS(a*X+b)' ENTER LAP . The calculator returns the result:

$$\frac{X}{SQ(X)+SQ(a)}\cdot COS(b)-SIN(b)\cdot\frac{a}{SQ(X)+SQ(a)}$$

Press EVAL to obtain –(a sin(b) – X cos(b))/(X^2+a^2). The transform is interpreted as follows: L {cos(a·t+b)} = (s·cos b – a·sin b)/(s^2+a^2).

Laplace transform theorems

To help you determine the Laplace transform of functions you can use a number of theorems, some of which are listed below. A few examples of the theorem applications are also included.

- Differentiation theorem for the first derivative. Let f_o be the initial condition for f(t), i.e., f(0) = f_o, then

$$L\{df/dt\} = s\cdot F(s) - f_o.$$

Example 1 – The velocity of a moving particle v(t) is defined as v(t) = dr/dt, where r = r(t) is the position of the particle. Let r_o = r(0), and R(s) =L{r(t)}, then, the transform of the velocity can be written as V(s) = L{v(t)}=L{dr/dt}= $s\cdot R(s)-r_o$.

- Differentiation theorem for the second derivative. Let f_o = f(0), and $(df/dt)_o$ = $df/dt|_{t=0}$, then $L\{d^2f/dt^2\} = s^2\cdot F(s) - s\cdot f_o - (df/dt)_o$.

Example 2 – As a follow up to Example 1, the acceleration a(t) is defined as $a(t) = d^2r/dt^2$. If the initial velocity is $v_o = v(0) = dr/dt|_{t=0}$, then the Laplace transform of the acceleration can be written as:

$$A(s) = L\{a(t)\} = L\{d^2r/dt^2\}= s^2\cdot R(s) - s\cdot r_o - v_o.$$

- Differentiation theorem for the n-th derivative.
  Let $f^{(k)}{}_o = d^kf/dx^k|_{t = 0}$, and $f_o = f(0)$, then

$$L\{d^nf/dt^n\} = s^n\cdot F(s) - s^{n-1}\cdot f_o - \ldots - s\cdot f^{(n-2)}{}_o - f^{(n-1)}{}_o.$$

- Linearity theorem. $L\{af(t)+bg(t)\} = a\cdot L\{f(t)\} + b\cdot L\{g(t)\}$.

- Differentiation theorem for the image function. Let $F(s) = L\{f(t)\}$, then $d^nF/ds^n = L\{(-t)^n\cdot f(t)\}$.

Example 3 – Let $f(t) = e^{-at}$, using the calculator with 'EXP(-a*X)' ENTER LAP, you get '1/(X+a)', or $F(s) = 1/(s+a)$. The third derivative of this expression can be calculated by using:

'X' ENTER ↱ ∂ 'X' ENTER ↱ ∂ 'X' ENTER ↱ ∂ EVAL

The result is

'-6/(X^4+4*a*X^3+6*a^2*X^2+4*a^3*X+a^4)', or

$$d^3F/ds^3 = -6/(s^4+4\cdot a\cdot s^3+6\cdot a^2\cdot s^2+4\cdot a^3\cdot s+a^4).$$

Now, use '(-X)^3*EXP(-a*X)' ENTER LAP EVAL. The result is exactly the same.

- Integration theorem. Let $F(s) = L\{f(t)\}$, then

$$L\left\{\int_0^t f(u)du\right\} = \frac{1}{s}\cdot F(s).$$

- Convolution theorem. Let $F(s) = L\{f(t)\}$ and $G(s) = L\{g(t)\}$, then

$$\mathsf{L}\left\{\int_0^t f(u)g(t-u)du\right\} = \mathsf{L}\{(f*g)(t)\} =$$

$$\mathsf{L}\{f(t)\}\cdot\mathsf{L}\{g(t)\} = F(s)\cdot G(s)$$

Example 4 – Using the convolution theorem, find the Laplace transform of (f*g)(t), if f(t) = sin(t), and g(t) = exp(t). To find F(s) = L{f(t)}, and G(s) = L{g(t)}, use: 'SIN(X)' ENTER LAP EVAL. Result, '1/(X^2+1)', i.e., F(s) = 1/(s^2+1). Also, 'EXP(X)' ENTER LAP. Result, '1/(X-1)', i.e., G(s) = 1/(s-1). Thus, L{(f*g)(t)} = F(s)·G(s) = 1/(s^2+1)·1/(s-1) = 1/((s-1)(s^2+1)) = 1/(s^3-s^2+s-1).

- Shift theorem for a shift to the right. Let F(s) = L{f(t)}, then

  $$\mathsf{L}\{f(t-a)\}=e^{-as}\cdot\mathsf{L}\{f(t)\} = e^{-as}\cdot F(s).$$

- Shift theorem for a shift to the left. Let F(s) = L{f(t)}, and a >0, then

  $$\mathsf{L}\{f(t+a)\} = e^{as}\cdot\left(F(s) - \int_0^a f(t)\cdot e^{-st}\cdot dt\right).$$

- Similarity theorem. Let F(s) = L{f(t)}, and a>0, then L{f(a·t)} = (1/a)·F(s/a).
- Damping theorem. Let F(s) = L{f(t)}, then L{e^{-bt}·f(t)} = F(s+b).
- Division theorem. Let F(s) = L{f(t)}, then

  $$\mathsf{L}\left\{\frac{f(t)}{t}\right\} = \int_s^\infty F(u)du.$$

- Laplace transform of a periodic function of period T:

  $$\mathsf{L}\{f(t)\} = \frac{1}{1-e^{-sT}}\cdot\int_0^T f(t)\cdot e^{-st}\cdot dt.$$

- Limit theorem for the initial value: Let F(s) = L{f(t)}, then

  $$f_0 = \lim_{t\to 0} f(t) = \lim_{s\to\infty}[s\cdot F(s)].$$

- Limit theorem for the final value: Let F(s) = L{f(t)}, then

$$f_{\infty} = \lim_{t\to\infty} f(t) = \lim_{s\to 0}[s \cdot F(s)].$$

Dirac's delta function and Heaviside's step function

In the analysis of control systems it is customary to utilize a type of functions that represent certain physical occurrences such as the sudden activation of a switch (Heaviside's step function, H(t)) or a sudden, instantaneous, peak in an input to the system (Dirac's delta function, δ(t)). These belong to a class of functions known as generalized or symbolic functions [e.g., see Friedman, B., 1956, Principles and Techniques of Applied Mathematics, Dover Publications Inc., New York (1990 reprint)].

The formal definition of <u>Dirac's delta function</u>, δ(x), is δ(x) = 0, for $x \neq 0$, and

$$\int_{-\infty}^{\infty} \delta(x)dx = 1.0.$$

Also, if f(x) is a continuous function, then $\int_{-\infty}^{\infty} f(x)\delta(x-x_0)dx = f(x_0).$

An interpretation for the integral above, paraphrased from Friedman (1990), is that the δ-function "picks out" the value of the function f(x) at $x = x_0$. Dirac's delta function is typically represented by an upward arrow at the point x = x0, indicating that the function has a non-zero value only at that particular value of x_0.

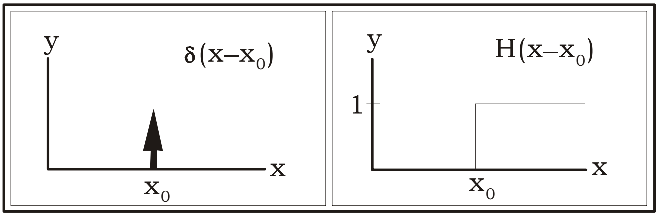
<u>Heaviside's step function</u>, H(x), is defined as

$$H(x) = \begin{cases} 1, & x > 0 \\ 0, & x < 0 \end{cases}$$

Also, for a continuous function f(x),

$$\int_{-\infty}^{\infty} f(x)H(x-x_0)dx = \int_{x_0}^{\infty} f(x)dx.$$

Dirac's delta function and Heaviside's step function are related by dH/dx = δ(x). The two functions are illustrated in the figure below.



You can prove that $L\{H(t)\} = 1/s$,
from which it follows that $L\{U_o \cdot H(t)\} = U_o/s$,
where U_o is a constant. Also, $L^{-1}\{1/s\}=H(t)$,
and $L^{-1}\{U_o/s\}= U_o \cdot H(t)$.

Also, using the shift theorem for a shift to the right, $L\{f(t-a)\}=e^{-as} \cdot L\{f(t)\} = e^{-as} \cdot F(s)$, we can write $L\{H(t-k)\}=e^{-ks} \cdot L\{H(t)\} = e^{-ks} \cdot (1/s) = (1/s) \cdot e^{-ks}$.

Another important result, known as the <u>second shift theorem for a shift to the right</u>, is that $L^{-1}\{e^{-as} \cdot F(s)\}=f(t-a) \cdot H(t-a)$, with $F(s) = L\{f(t)\}$.

In the calculator the Heaviside step function H(t) is simply referred to as '1'. To check the transform in the calculator use: `1` `ENTER` LAP. The result is '1/X', i.e., $L\{1\} = 1/s$. Similarly, 'U0' `ENTER` LAP , produces the result 'U0/X', i.e., $L\{U_0\} = U_0/s$.

You can obtain Dirac's delta function in the calculator by using: `1` `ENTER` ILAP
The result is 'Delta(X)'.

This result is simply symbolic, i.e., you cannot find a numerical value for, say 'Delta(5)'.

This result can be defined the Laplace transform for Dirac's delta function, because from $L^{-1}\{1.0\}= \delta(t)$, it follows that $L\{\delta(t)\} = 1.0$

Also, using the shift theorem for a shift to the right, $L\{f(t-a)\}=e^{-as} \cdot L\{f(t)\} = e^{-as} \cdot F(s)$, we can write $L\{\delta(t-k)\}=e^{-ks} \cdot L\{\delta(t)\} = e^{-ks} \cdot 1.0 = e^{-ks}$.

Applications of Laplace transform in the solution of linear ODEs

At the beginning of the section on Laplace transforms we indicated that you could use these transforms to convert a linear ODE in the time domain into an algebraic equation in the image domain. The resulting equation is then solved for a function F(s) through algebraic methods, and the solution to the ODE is found by using the inverse Laplace transform on F(s).

The theorems on derivatives of a function, i.e.,

$$L\{df/dt\} = s\cdot F(s) - f_o,$$

$$L\{d^2f/dt^2\} = s^2\cdot F(s) - s\cdot f_o - (df/dt)_o,$$

and, in general,

$$L\{d^nf/dt^n\} = s^n\cdot F(s) - s^{n-1}\cdot f_o - \ldots - s\cdot f^{(n-2)}_o - f^{(n-1)}_o,$$

are particularly useful in transforming an ODE into an algebraic equation.

<u>Example</u> 1 – To solve the first order equation,

$$dh/dt + k\cdot h(t) = a\cdot e^{-t},$$

by using Laplace transforms, we can write:

$$L\{dh/dt + k\cdot h(t)\} = L\{a\cdot e^{-t}\},$$

$$L\{dh/dt\} + k\cdot L\{h(t)\} = a\cdot L\{e^{-t}\}.$$

Note: 'EXP(-X)' `ENTER` LAP , produces '1/(X+1)', i.e., $L\{e^{-t}\}=1/(s+1)$.

With $H(s) = L\{h(t)\}$, and $L\{dh/dt\} = s\cdot H(s) - h_o$, where $h_o = h(0)$, the transformed equation is $s\cdot H(s)-h_o+k\cdot H(s) = a/(s+1)$.

Use the calculator to solve for H(s), by writing:

'X*H-h0+k*H=a/(X+1)' `ENTER` 'H' ISOL

The result is 'H=((X+1)*h0+a)/(X^2+(k+1)*X+k)'.

To find the solution to the ODE, h(t), we need to use the inverse Laplace transform, as follows:

OBJ→ ◄ ◄	Isolates right-hand side of last expression
ILAP EVAL	Obtains the inverse Laplace transform

The result is $\frac{a\cdot e^{k\cdot X}+((k-1)\cdot ho-a)\cdot e^{X}}{(k-1)\cdot e^{X}\cdot e^{k\cdot X}}$. Replacing X with t in this expression and simplifying, results in $h(t) = a/(k\text{-}1)\cdot e^{-t} +((k\text{-}1)\cdot h_o\text{-}a)/(k\text{-}1)\cdot e^{-kt}$.

Check what the solution to the ODE would be if you use the function LDEC:

'a*EXP(-X)' ENTER 'X+k' ENTER LDEC EVAL

The result is: $\frac{a\cdot e^{k\cdot X}+((k-1)\cdot cC0-a)\cdot e^{X}}{(k-1)\cdot e^{X}\cdot e^{k\cdot X}}$, i.e.,

$$h(t) = a/(k\text{-}1)\cdot e^{-t} +((k\text{-}1)\cdot cC_o\text{-}a)/(k\text{-}1)\cdot e^{-kt}.$$

Thus, cC0 in the results from LDEC represents the initial condition h(0).

> **Note**: When using the function LDEC to solve a linear ODE of order n in f(X), the result will be given in terms of n constants cC0, cC1, cC2, …, cC(n-1), representing the initial conditions f(0), f'(0), f"(0), …, $f^{(n-1)}$ (0).

Example 2 – Use Laplace transforms to solve the second-order linear equation,

$$d^2y/dt^2+2y = \sin 3t.$$

Using Laplace transforms, we can write:

$$L\{d^2y/dt^2+2y\} = L\{\sin 3t\},$$

$$L\{d^2y/dt^2\} + 2\cdot L\{y(t)\} = L\{\sin 3t\}.$$

Note: 'SIN(3*X)' ENTER LAP EVAL produces '3/(X^2+9)', i.e., $L\{\sin 3t\}=3/(s^2+9)$.

With $Y(s) = L\{y(t)\}$, and $L\{d^2y/dt^2\} = s^2\cdot Y(s) - s\cdot y_o - y_1$, where $y_o = h(0)$ and $y_1 = h'(0)$, the transformed equation is

$$s^2\cdot Y(s) - s\cdot y_o - y_1 + 2\cdot Y(s) = 3/(s^2+9).$$

Use the calculator to solve for Y(s), by writing:

'X^2*Y-X*y0-y1+2*Y=3/(X^2+9)' ENTER 'Y' ISOL

The result is

'Y=((X^2+9)*y1+(y0*X^3+9*y0*X+3))/(X^4+11*X^2+18)'.

To find the solution to the ODE, y(t), we need to use the inverse Laplace transform, as follows:

OBJ→ ◄ ◄ Isolates right-hand side of last expression
ILAP EVAL Obtains the inverse Laplace transform

The result is

$$\frac{\left(7\cdot\sqrt{2}\cdot y1+3\cdot\sqrt{2}\right)\cdot \mathrm{SIN}\left(\sqrt{2}\cdot X\right)+14\cdot y0\cdot \mathrm{COS}\left(\sqrt{2}\cdot X\right)-2\cdot \mathrm{SIN}(3\cdot X)}{14}$$

i.e.,

$$y(t) = -(1/7)\sin 3x + y_o \cos \sqrt{2}x + (\sqrt{2}\,(7y_1+3)/14)\sin \sqrt{2}x.$$

Check what the solution to the ODE would be if you use the function LDEC:

'SIN(3*X)' ENTER 'X^2+2' ENTER LDEC EVAL

The result is:

$$\frac{\left(7\cdot\sqrt{2}\cdot cC1+3\cdot\sqrt{2}\right)\cdot \mathrm{SIN}\left(\sqrt{2}\cdot X\right)+14\cdot cC0\cdot \mathrm{COS}\left(\sqrt{2}\cdot X\right)-2\cdot \mathrm{SIN}(3\cdot X)}{14}$$

i.e., the same as before with cC0 = y0 and cC1 = y1.

Note: Using the two examples shown here, we can confirm what we indicated earlier, i.e., that function ILAP uses Laplace transforms and inverse transforms to solve linear ODEs given the right-hand side of the equation and the characteristic equation of the corresponding homogeneous ODE.

Example 3 – Consider the equation

$$d^2y/dt^2+y = \delta(t\text{-}3),$$

where δ(t) is Dirac's delta function.

Using Laplace transforms, we can write:

$$L\{d^2y/dt^2+y\} = L\{\delta(t\text{-}3)\},$$

$$L\{d^2y/dt^2\} + L\{y(t)\} = L\{\delta(t\text{-}3)\}.$$

With '`Delta(X-3)`' ENTER LAP , the calculator produces EXP(-3*X), i.e., L{δ(t-3)} = e^{-3s}. With Y(s) = L{y(t)}, and $L\{d^2y/dt^2\} = s^2\cdot Y(s) - s\cdot y_o - y_1$, where y_o = h(0) and y_1 = h'(0), the transformed equation is $s^2\cdot Y(s) - s\cdot y_o - y_1 + Y(s) = e^{-3s}$. Use the calculator to solve for Y(s), by writing:

'X^2*Y-X*y0-y1+Y=EXP(-3*X)' ENTER 'Y' ISOL

The result is 'Y=(X*y0+(y1+EXP(-(3*X))))/(X^2+1)'.

To find the solution to the ODE, y(t), we need to use the inverse Laplace transform, as follows:

OBJ→ ◄ ◄	Isolates right-hand side of last expression
ILAP EVAL	Obtains the inverse Laplace transform

The result is 'y1*SIN(X)+y0*COS(X)+SIN(X-3)*Heaviside(X-3)'.

Notes:

[1]. An alternative way to obtain the inverse Laplace transform of the expression '(X*y0+(y1+EXP(-(3*X))))/(X^2+1)' is by separating the expression into partial fractions, i.e.,

'y0*X/(X^2+1) + y1/(X^2+1) + EXP(-3*X)/(X^2+1)',

and use the linearity theorem of the inverse Laplace transform

$$L^{-1}\{a\cdot F(s)+b\cdot G(s)\} = a\cdot L^{-1}\{F(s)\} + b\cdot L^{-1}\{G(s)\},$$

to write,

$$L^{-1}\{y_o\cdot s/(s^2+1)+y_1/(s^2+1)) + e^{-3s}/(s^2+1)) \} =$$

$$y_o\cdot L^{-1}\{s/(s^2+1)\}+ y_1\cdot L^{-1}\{1/(s^2+1)\}+ L^{-1}\{e^{-3s}/(s^2+1))\},$$

Then, we use the calculator to obtain the following:

'X/(X^2+1)' [ENTER] ILAP Result, 'COS(X)', i.e., $L^{-1}\{s/(s^2+1)\}= \cos t$.

'1/(X^2+1)' [ENTER] ILAP Result, 'SIN(X)', i.e., $L^{-1}\{1/(s^2+1)\}= \sin t$.

'EXP(-3*X)/(X^2+1)' [ENTER] ILAP Result, SIN(X-3)*Heaviside(X-3)'.

[2]. The very last result, i.e., the inverse Laplace transform of the expression '(EXP(-3*X)/(X^2+1))', can also be calculated by using the second shifting theorem for a shift to the right

$$L^{-1}\{e^{-as} \cdot F(s)\}=f(t-a)\cdot H(t-a),$$

if we can find an inverse Laplace transform for $1/(s^2+1)$. With the calculator, try '1/(X^2+1)' [ENTER] ILAP. The result is 'SIN(X)'. Thus, $L^{-1}\{e^{-3s}/(s^2+1)\}\} = \sin(t-3)\cdot H(t-3)$,

Check what the solution to the ODE would be if you use the function LDEC:

'Delta(X-3)' [ENTER] 'X^2+1' [ENTER] LDEC [EVAL]

The result is:

'SIN(X-3)*Heaviside(X-3) + cC1*SIN(X) + cC0*COS(X)'.

Please notice that the variable X in this expression actually represents the variable t in the original ODE. Thus, the translation of the solution in paper may be written as:

$$y(t) = Co \cdot \cos t + C_1 \cdot \sin t + \sin(t-3) \cdot H(t-3)$$

When comparing this result with the previous result for y(t), we conclude that $cC_o = y_o$, $cC_1 = y_1$.

Defining and using Heaviside's step function in the calculator

The previous example provided some experience with the use of Dirac's delta function as input to a system (i.e., in the right-hand side of the ODE describing the system). In this example, we want to use Heaviside's step function, H(t). In the calculator we can define this function as:

'H(X) = IFTE(X>0, 1, 0)' ENTER ← DEF

This definition will create the variable H in the calculator's soft menu key.

Example 1 – To see a plot of H(t-2), for example, use a FUNCTION type of plot (see Chapter 12):

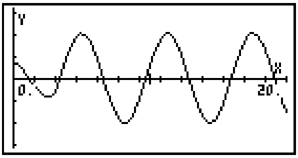
- Press ← 2D/3D , simultaneously in RPN mode, to access to the PLOT SETUP window.
- Change `TYPE` to `FUNCTION`, if needed
- Change EQ to 'H(X-2)'.
- Make sure that `Indep` is set to 'X'.
- Press NXT OK to return to normal calculator display.
- Press ← WIN , simultaneously, to access the PLOT window.
- Change the H-VIEW range to 0 to 20, and the V-VIEW range to -2 to 2.
- Press ERASE DRAW to plot the function .

Use of the function H(X) with LDEC, LAP, or ILAP, is not allowed in the calculator. You have to use the main results provided earlier when dealing with the Heaviside step function, i.e., $\mathcal{L}\{H(t)\} = 1/s$, $\mathcal{L}^{-1}\{1/s\}=H(t)$, $\mathcal{L}\{H(t-k)\}=e^{-ks}\cdot\mathcal{L}\{H(t)\} = e^{-ks}\cdot(1/s) = \cdot(1/s)\cdot e^{-ks}$ and $\mathcal{L}^{-1}\{e^{-as}\cdot F(s)\}=f(t-a)\cdot H(t-a)$.

Example 2 – The function $H(t-t_o)$ when multiplied to a function f(t), i.e., $H(t-t_o)f(t)$, has the effect of switching on the function f(t) at $t = t_o$. For example, the solution obtained in Example 3, above, was $y(t) = y_o \cos t + y_1 \sin t + \sin(t-3)\cdot H(t-3)$. Suppose we use the initial conditions $y_o = 0.5$, and $y_1 = -0.25$. Let's plot this function to see what it looks like:

- Press [←] 2D/3D , simultaneously if in RPN mode, to access to the PLOT SETUP window.
- Change `TYPE` to `FUNCTION`, if needed
- Change EQ to '0.5*COS(X)-0.25*SIN(X)+SIN(X-3)*H(X-3)'.
- Make sure that `Indep` is set to 'X'.
- H-VIEW: 0 20, V-VIEW: -3 2.
- Press ERASE DRAW to plot the function.
- Press EDIT (NXT) LABEL to see the plot.

The resulting graph will look like this:



Notice that the signal starts with a relatively small amplitude, but suddenly, at t=3, it switches to an oscillatory signal with a larger amplitude. The difference between the behavior of the signal before and after t = 3 is the "switching on" of the particular solution $y_p(t) = \sin(t-3)\cdot H(t-3)$. The behavior of the signal before t = 3 represents the contribution of the homogeneous solution, $y_h(t) = y_o \cos t + y_1 \sin t$.

The solution of an equation with a driving signal given by a Heaviside step function is shown below.

<u>Example 3</u> – Determine the solution to the equation, $d^2y/dt^2+y = H(t-3)$,

where H(t) is Heaviside's step function. Using Laplace transforms, we can write: L{d^2y/dt^2+y} = L{H(t-3)}, L{d^2y/dt^2} + L{y(t)} = L{H(t-3)}. The last term in this expression is: L{H(t-3)} = $(1/s)\cdot e^{-3s}$. With Y(s) = L{y(t)}, and L{d^2y/dt^2} = $s^2\cdot Y(s) - s\cdot y_o - y_1$, where y_o = h(0) and y_1 = h'(0), the transformed equation is $s^2\cdot Y(s) - s\cdot y_o - y_1 + Y(s) = (1/s)\cdot e^{-3s}$. Change CAS mode to Exact, if necessary. Use the calculator to solve for Y(s), by writing:

'X^2*Y-X*y0-y1+Y=(1/X)*EXP(-3*X)' [ENTER] 'Y' ISOL

The result is 'Y=(X^2*y0+X*y1+EXP(-3*X))/(X^3+X)'.

To find the solution to the ODE, y(t), we need to use the inverse Laplace transform, as follows:

OBJ→ [◄] [◄]	Isolates right-hand side of last expression
ILAP	Obtains the inverse Laplace transform

The result is 'y1*SIN(X-1)+y0*COS(X-1)-(COS(X-3)-1)*Heaviside(X-3)'.

Thus, we write as the solution: $y(t) = y_o \cos t + y_1 \sin t + H(t-3)\cdot(1+\sin(t-3))$.

Check what the solution to the ODE would be if you use the function LDEC:

'H(X-3)' [ENTER] [ENTER] 'X^2+1' [ENTER] LDEC

The result is:

$$\text{SIN(X)}\cdot\int_0^{+\infty}\frac{\text{IFTE(ttt-3>0,1,0)}}{e^{\text{X}\cdot\text{ttt}}}\,d\text{ttt} + \text{cC1}\cdot\text{SIN(X)}+\text{cC0}\cdot\text{COS(X)}$$

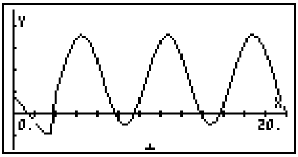
Please notice that the variable X in this expression actually represents the variable t in the original ODE, and that the variable *ttt* in this expression is a dummy variable. Thus, the translation of the solution in paper may be written as:

$$y(t) = Co\cdot\cos t + C_1\cdot\sin t + \sin t\cdot\int_0^\infty H(u-3)\cdot e^{-ut}\cdot du.$$

Example 4 – Plot the solution to Example 3 using the same values of y_o and y_1 used in the plot of Example 1, above. We now plot the function

$$y(t) = 0.5 \cos t - 0.25 \sin t + (1+\sin(t-3))\cdot H(t-3).$$

In the range 0 < t < 20, and changing the vertical range to (-1,3), the graph should look like this:



Again, there is a new component to the motion switched at t=3, namely, the particular solution $y_p(t) = [1+\sin(t-3)]\cdot H(t-3)$, which changes the nature of the solution for t>3.

The Heaviside step function can be combined with a constant function and with linear functions to generate square, triangular, and saw tooth finite pulses, as follows:

- Square pulse of size U_o in the interval a < t < b:

$$f(t) = Uo[H(t-a)-H(t-b)].$$

- Triangular pulse with a maximum value Uo, increasing from a < t < b, decreasing from b < t < c:

$$f(t) = U_o\cdot ((t-a)/(b-a)\cdot[H(t-a)-H(t-b)]+(1-(t-b)/(b-c))[H(t-b)-H(t-c)]).$$

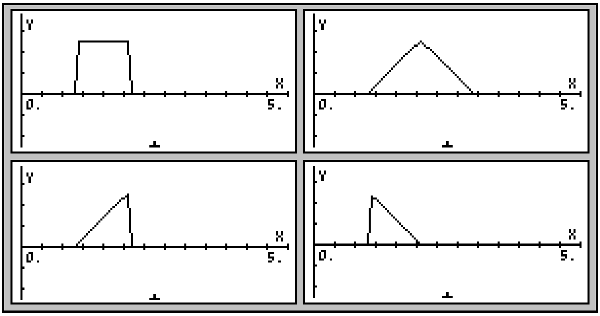
- Saw tooth pulse increasing to a maximum value Uo for a < t < b, dropping suddenly down to zero at t = b:

$$f(t) = U_o\cdot (t-a)/(b-a)\cdot[H(t-a)-H(t-b)].$$

- Saw tooth pulse increasing suddenly to a maximum of Uo at t = a, then decreasing linearly to zero for a < t < b:

$f(t) = U_o \cdot [1-(t-a)/(b-1)] \cdot [H(t-a)-H(t-b)]$.

Examples of the plots generated by these functions, for Uo = 1, a = 2, b = 3, c = 4, horizontal range = (0,5), and vertical range = (-1, 1.5), are shown in the figures below:



Fourier series

Fourier series are series involving sine and cosine functions typically used to expand periodic functions. A function f(x) is said to be *periodic*, of period T, if f(x+T) = f(t). For example, because sin(x+2π) = sin x, and cos(x+2π) = cos x, the functions *sin* and *cos* are 2π-periodic functions. If two functions f(x) and g(x) are periodic of period T, then their linear combination h(x) = a·f(x) + b·g(x), is also periodic of period T. A T-periodic function f(t) can be expanded into a series of sine and cosine functions known as a Fourier series given by

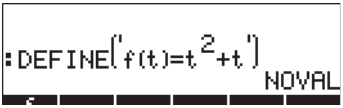
$$f(t) = a_0 + \sum_{n=1}^{\infty}\left(a_n \cdot \cos\frac{2n\pi}{T}t + b_n \cdot \sin\frac{2n\pi}{T}t\right)$$

where the coefficients a_n and b_n are given by

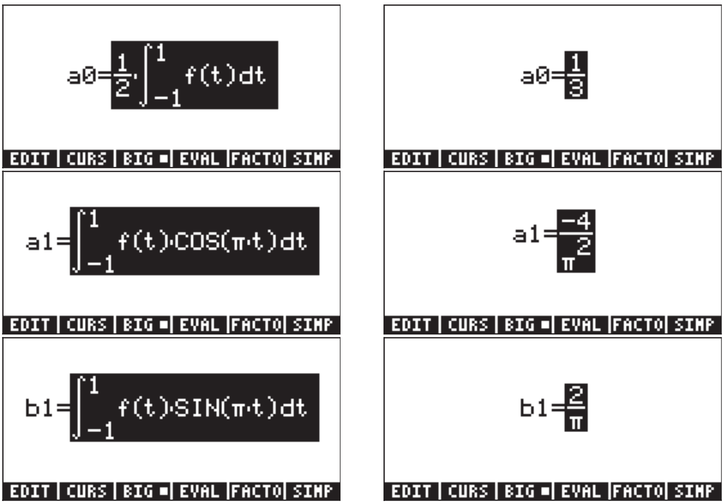
$$a_0 = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\cdot dt, \quad a_n = \frac{2}{T}\int_{-T/2}^{T/2} f(t)\cdot\cos\frac{2n\pi}{T}t\cdot dt,$$

$$b_n = \int_{-T/2}^{T/2} f(t)\cdot\sin\frac{2n\pi}{T}t\cdot dt.$$

The following exercises are in ALG mode, with CAS mode set to Exact. (When you produce a graph, the CAS mode will be reset to Approx. Make sure to set it back to Exact after producing the graph.) Suppose, for example, that the function $f(t) = t^2+t$ is periodic with period $T = 2$. To determine the coefficients a_0, a_1, and b_1 for the corresponding Fourier series, we proceed as follows: First, define function $f(t) = t^2+t$:



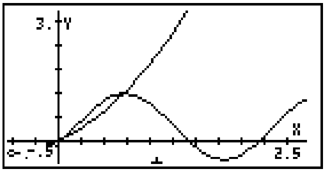
Next, we'll use the Equation Writer to calculate the coefficients:



Thus, the first three terms of the function are:

$$f(t) \approx 1/3 - (4/\pi^2)\cdot\cos(\pi\cdot t)+(2/\pi)\cdot\sin(\pi\cdot t).$$

A graphical comparison of the original function with the Fourier expansion using these three terms shows that the fitting is acceptable for $t < 1$, or thereabouts. But, then, again, we stipulated that $T/2 = 1$. Therefore, the fitting is valid only between $-1 < t < 1$.



Function FOURIER

An alternative way to define a Fourier series is by using complex numbers as follows:

$$f(t) = \sum_{n=-\infty}^{+\infty} c_n \cdot \exp(\frac{2in\pi t}{T}),$$

where

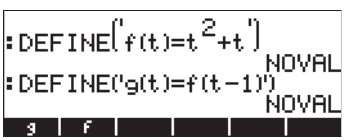
$$c_n = \frac{1}{T}\int_0^T f(t)\cdot \exp(\frac{2\cdot i\cdot n\cdot \pi}{T}\cdot t)\cdot dt, \quad n = -\infty,...,-2,-1,0,1,2,...\infty.$$

Function FOURIER provides the coefficient c_n of the complex-form of the Fourier series given the function f(t) and the value of n. The function FOURIER requires you to store the value of the period (T) of a T-periodic function into the CAS variable PERIOD before calling the function. The function FOURIER is available in the DERIV sub-menu within the CALC menu (← CALC).

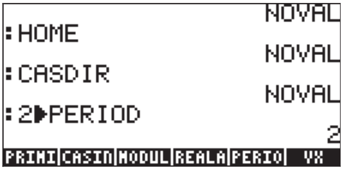
Fourier series for a quadratic function

Determine the coefficients c_0, c_1, and c_2 for the function $f(t) = t^2+t$, with period T = 2. (Note: Because the integral used by function FOURIER is calculated in the interval [0,T], while the one defined earlier was calculated in the interval [-T/2,T/2], we need to shift the function in the t-axis, by subtracting T/2 from t, i.e., we will use $g(t) = f(t-1) = (t-1)^2+(t-1)$.)

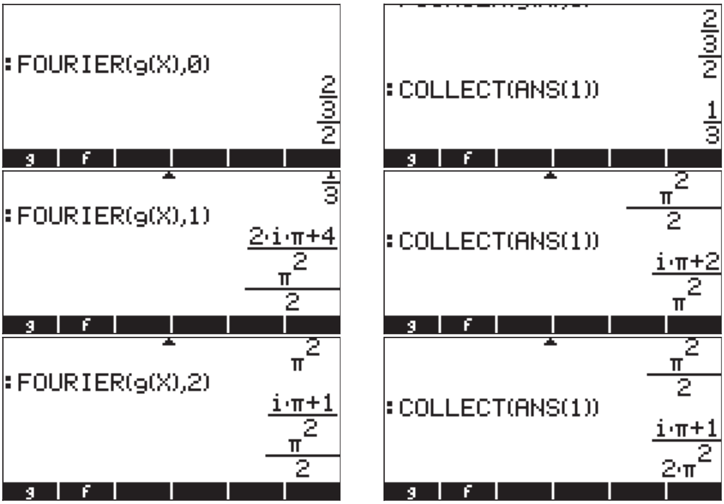
Using the calculator in ALG mode, first we define functions f(t) and g(t):



Next, we move to the CASDIR sub-directory under HOME to change the value of variable PERIOD, e.g., ← (hold) UPDIR ENTER VAR CASDI ENTER 2 STO▶ PERIO ENTER



Return to the sub-directory where you defined functions f and g, and calculate the coefficients (Accept change to Complex mode when requested):

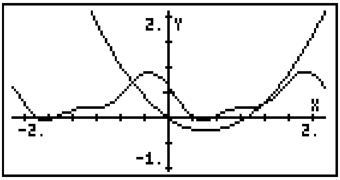


Thus, $c_0 = 1/3$, $c_1 = (\pi\cdot i+2)/\pi^2$, $c_2 = (\pi\cdot i+1)/(2\pi^2)$.

The Fourier series with three elements will be written as

$$g(t) \approx \mathrm{Re}[(1/3) + (\pi\cdot i+2)/\pi^2\cdot \exp(i\cdot\pi\cdot t)+ (\pi\cdot i+1)/(2\pi^2)\cdot \exp(2\cdot i\cdot\pi\cdot t)].$$

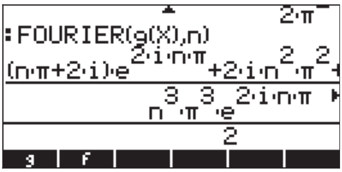
A plot of the shifted function g(t) and the Fourier series fitting follows:

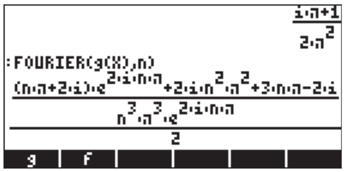


The fitting is somewhat acceptable for 0<t<2, although not as good as in the previous example.

A general expression for c_n

The function FOURIER can provide a general expression for the coefficient c_n of the complex Fourier series expansion. For example, using the same function g(t) as before, the general term c_n is given by (figures show normal font and small font displays):





The general expression turns out to be, after simplifying the previous result,

$$c_n = \frac{(n\pi + 2i)\cdot e^{2in\pi} + 2i^2 n^2 \pi^2 + 3n\pi - 2i}{2n^3\pi^3 \cdot e^{2in\pi}}$$

We can simplify this expression even further by using Euler's formula for complex numbers, namely, $e^{2in\pi} = \cos(2n\pi) + i\cdot\sin(2n\pi) = 1 + i\cdot 0 = 1$, since $\cos(2n\pi) = 1$, and $\sin(2n\pi) = 0$, for n integer.

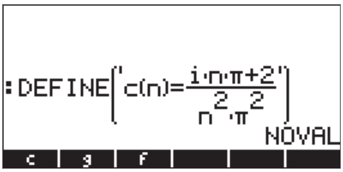
Using the calculator you can simplify the expression in the equation writer (EQW) by replacing $e^{2in\pi} = 1$. The figure shows the expression after simplification:

The result is $c_n = (i \cdot n \cdot \pi + 2)/(n^2 \cdot \pi^2)$.

Putting together the complex Fourier series

Having determined the general expression for c_n, we can put together a finite complex Fourier series by using the summation function (Σ) in the calculator as follows:

- First, define a function c(n) representing the general term c_n in the complex Fourier series.



- Next, define the finite complex Fourier series, F(X,k), where X is the independent variable and k determines the number of terms to be used. Ideally we would like to write this finite complex Fourier series as

$$F(X,k) = \sum_{n=-k}^{k} c(n) \cdot \exp(\frac{2 \cdot i \cdot \pi \cdot n}{T} \cdot X)$$

However, because the function c(n) is not defined for n = 0, we will be better advised to re-write the expression as

$$F(X,k,c0) = c0 +$$

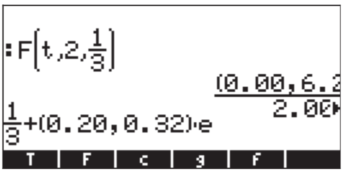
$$\sum_{n=1}^{k} [c(n) \cdot \exp(\frac{2 \cdot i \cdot \pi \cdot n}{T} \cdot X) + c(-n) \cdot \exp(-\frac{2 \cdot i \cdot \pi \cdot n}{T} \cdot X)],$$

Or, in the calculator entry line as:

DEFINE('F(X,k,c0) = c0+Σ(n=1,k,c(n)*EXP(2*i*π*n*X/T)+
c(-n)*EXP(-(2*i*π*n*X/T))'),

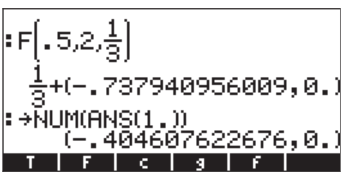
where T is the period, T = 2. The following screen shots show the definition of function F and the storing of T = 2:

The function F can be used to generate the expression for the complex Fourier series for a finite value of k. For example, for k = 2, c_0 = 1/3,and using t as the independent variable, we can evaluate F(t,2,1/3) to get:



This result shows only the first term (c0) and part of the first exponential term in the series. The decimal display format was changed to Fix with 2 decimals to be able to show some of the coefficients in the expansion and in the exponent. As expected, the coefficients are complex numbers.

The function F, thus defined, is fine for obtaining values of the finite Fourier series. For example, a single value of the series, e.g., F(0.5,2,1/3), can be obtained by using (CAS modes set to Exact, step-by-step, and Complex):



Accept change to `Approx` mode if requested. The result is the value –0.40467…. The actual value of the function g(0.5) is g(0.5) = -0.25. The following calculations show how well the Fourier series approximates this value as the number of components in the series, given by k, increases:

F (0.5, 1, 1/3) = (-0.303286439037,0.)

F (0.5, 2, 1/3) = (-0.404607622676,0.)

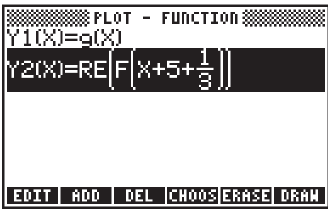
F (0.5, 3, 1/3) = (-0.192401031886,0.)

F (0.5, 4, 1/3) = (-0.167070735979,0.)

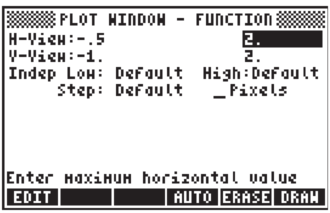
F (0.5, 5, 1/3) = (-0.294394690453,0.)

F (0.5, 6, 1/3) = (-0.305652599743,0.)

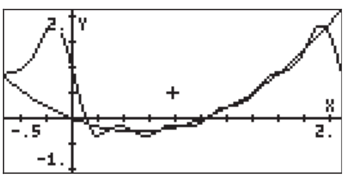
To compare the results from the series with those of the original function, load these functions into the PLOT – FUNCTION input form (← Y= , simultaneously if using RPN mode):



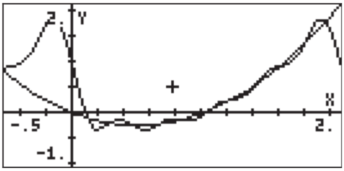
Change the limits of the Plot Window (← WIN) as follows:



Press the soft-menu keys ERASE DRAW to produce the plot:



Notice that the series, with 5 terms, "hugs" the graph of the function very closely in the interval 0 to 2 (i.e., through the period T = 2). You can also notice a periodicity in the graph of the series. This periodicity is easy to visualize by expanding the horizontal range of the plot to (-0.5,4):



Fourier series for a triangular wave

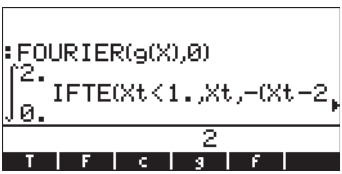
Consider the function

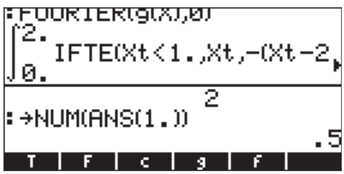
$$g(x) = \begin{cases} x, & if\ 0 < x < 1 \\ 2 - x, & if\ 1 < x < 2 \end{cases}$$

which we assume to be periodic with period T = 2. This function can be defined in the calculator, in ALG mode, by the expression

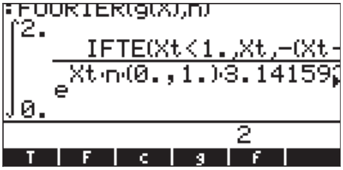
DEFINE('g(X) = IFTE(X<1,X,2-X)')

If you started this example after finishing example 1 you already have a value of 2 stored in CAS variable PERIOD. If you are not sure, check the value of this variable, and store a 2 in it if needed. The coefficient c_0 for the Fourier series is calculated as follows:





The calculator will request a change to Approx mode because of the integration of the function IFTE() included in the integrand. Accepting, the change to Approx produces $c_0 = 0.5$. If we now want to obtain a generic expression for the coefficient c_n use:



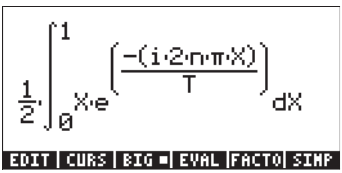
The calculator returns an integral that cannot be evaluated numerically because it depends on the parameter n. The coefficient can still be calculated by typing its definition in the calculator, i.e.,

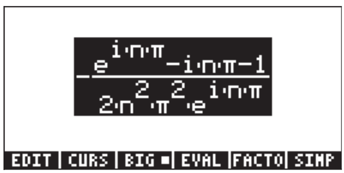
$$\frac{1}{2}\cdot\int_0^1 X\cdot EXP\left(-\frac{i\cdot 2\cdot n\cdot\pi\cdot X}{T}\right)\cdot dX+$$

$$\frac{1}{2}\cdot\int_1^2 (2-X)\cdot EXP\left(-\frac{i\cdot 2\cdot n\cdot\pi\cdot X}{T}\right)\cdot dX$$

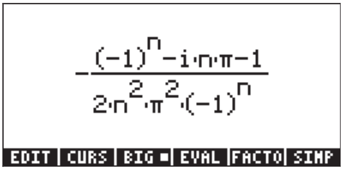
where T = 2 is the period. The value of T can be stored using:

Typing the first integral above in the Equation Writer, selecting the entire expression, and using EVAL, will produce the following:

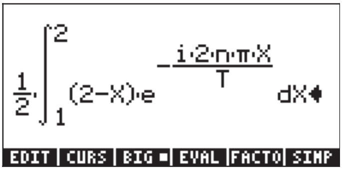


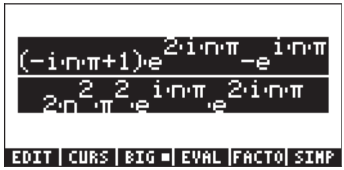


Recall the $e^{in\pi} = \cos(n\pi) + i\cdot\sin(n\pi) = (-1)^n$. Performing this substitution in the result above we have:

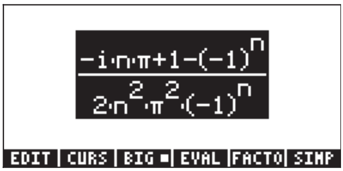


Press `ENTER` `ENTER` to copy this result to the screen. Then, reactivate the Equation Writer to calculate the second integral defining the coefficient c_n, namely,

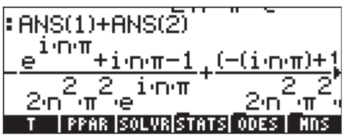




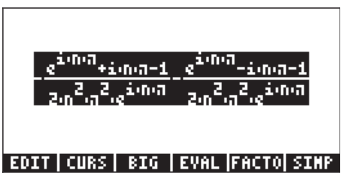
Once again, replacing $e^{in\pi} = (-1)^n$, and using $e^{2in\pi} = 1$, we get:

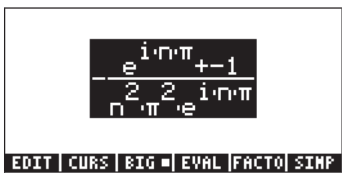


Press `ENTER` `ENTER` to copy this second result to the screen. Now, add ANS(1) and ANS(2) to get the full expression for c_n:



Pressing `▼` will place this result in the Equation Writer, where we can simplify (`SIMP`) it to read:



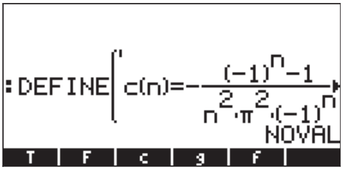


Once again, replacing $e^{in\pi} = (-1)^n$, results in

This result is used to define the function c(n) as follows:

DEFINE('c(n) = - (((-1)^n-1)/(n^2*π^2*(-1)^n)')

i.e.,

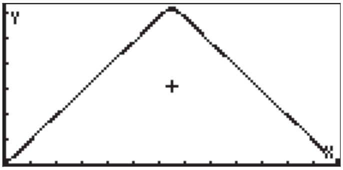


Next, we define function F(X,k,c0) to calculate the Fourier series (if you completed example 1, you already have this function stored):

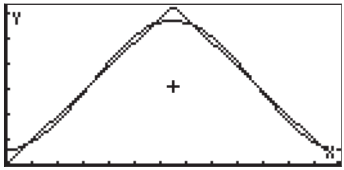
DEFINE('F(X,k,c0) = c0+Σ(n=1,k,c(n)*EXP(2*i*π*n*X/T)+
c(-n)*EXP(-(2*i*π*n*X/T))'),

To compare the original function and the Fourier series we can produce the simultaneous plot of both functions. The details are similar to those of example 1, except that here we use a horizontal range of 0 to 2 and a vertical range from 0 to 1, and adjust the equations to plot as shown here:

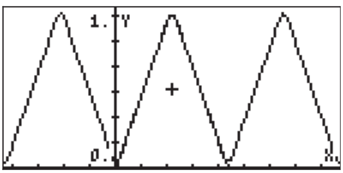
The resulting graph is shown below for k = 5 (the number of elements in the series is 2k+1, i.e., 11, in this case):



From the plot it is very difficult to distinguish the original function from the Fourier series approximation. Using k = 2, or 5 terms in the series, shows not so good a fitting:



The Fourier series can be used to generate a periodic triangular wave (or saw tooth wave) by changing the horizontal axis range, for example, from –2 to 4. The graph shown below uses k = 5:



Fourier series for a square wave

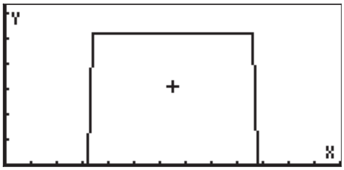
A square wave can be generated by using the function

$$g(x) = \begin{cases} 0, & if\ 0 < x < 1 \\ 1, & if\ 1 < x < 3 \\ 0, & if\ 3 < x < 4 \end{cases}$$

In this case, the period T, is 4. Make sure to change the value of variable to 4 (use: 4 STO▶ ENTER). Function g(X) can be defined in the calculator by using

DEFINE('g(X) = IFTE((X>1) AND (X<3),1,0)')

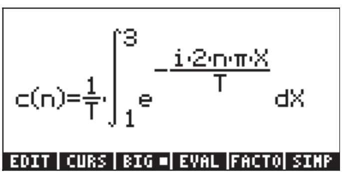
The function plotted as follows (horizontal range: 0 to 4, vertical range:0 to 1.2):

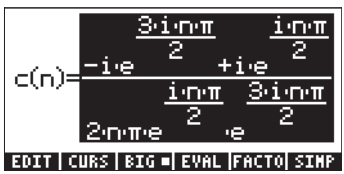


Using a procedure similar to that of the triangular shape in example 2, above, you can find that

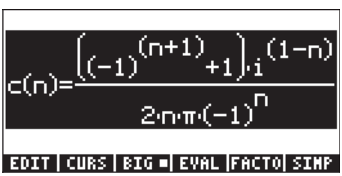
$$c_0 = \frac{1}{T} \cdot \left(\int_1^3 1 \cdot dX \right) = 0.5,$$

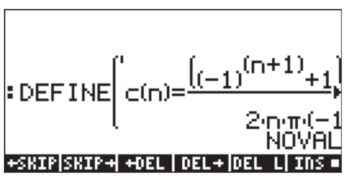
and



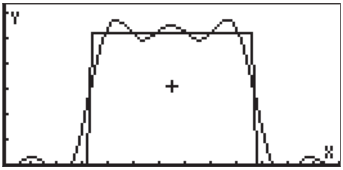


We can simplify this expression by using $e^{in\pi/2} = i^n$ and $e^{3in\pi/2} = (-i)^n$ to get:

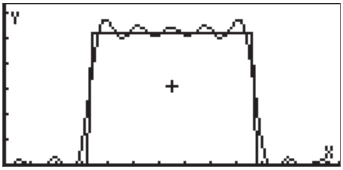




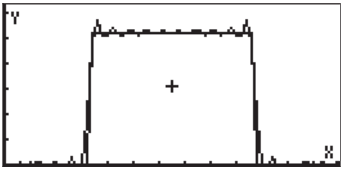
The simplification of the right-hand side of c(n), above, is easier done on paper (i.e., by hand). Then, retype the expression for c(n) as shown in the figure to the left above, to define function c(n). The Fourier series is calculated with F(X,k,c0), as in examples 1 and 2 above, with c0 = 0.5. For example, for k = 5, i.e., with 11 components, the approximation is shown below:



A better approximation is obtained by using k = 10, i.e.,



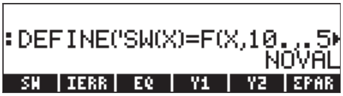
For k = 20, the fitting is even better, but it takes longer to produce the graph:



Fourier series applications in differential equations

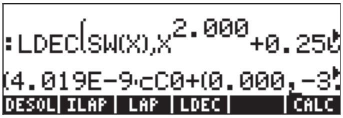
Suppose we want to use the periodic square wave defined in the previous example as the excitation of an undamped spring-mass system whose homogeneous equation is: $d^2y/dX^2 + 0.25y = 0$.

We can generate the excitation force by obtaining an approximation with k =10 out of the Fourier series by using SW(X) = F(X,10,0.5):

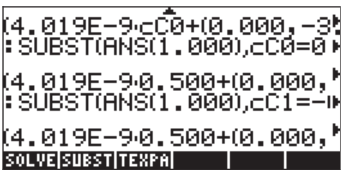


We can use this result as the first input to the function LDEC when used to obtain a solution to the system $d^2y/dX^2 + 0.25y = SW(X)$, where SW(X) stands for Square Wave function of X. The second input item will be the characteristic equation corresponding to the homogeneous ODE shown above, i.e., 'X^2+0.25' .

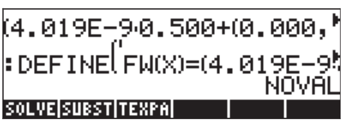
With these two inputs, function LDEC produces the following result (decimal format changed to Fix with 3 decimals).



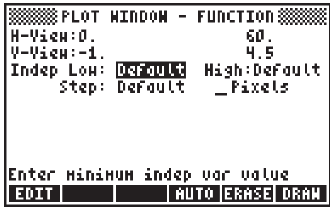
Pressing ▽ allows you to see the entire expression in the Equation writer. Exploring the equation in the Equation Writer reveals the existence of two constants of integration, cC0 and cC1. These values would be calculated using initial conditions. Suppose that we use the values cC0 = 0.5 and cC1 = -0.5, we can replace those values in the solution above by using function SUBST (see Chapter 5). For this case, use SUBST(ANS(1),cC0=0.5) ENTER, followed by SUBST(ANS(1),cC1=-0.5) ENTER. Back into normal calculator display we can use:



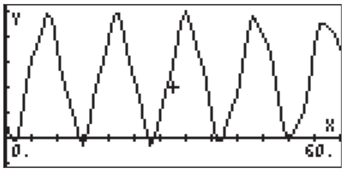
The latter result can be defined as a function, FW(X), as follows (cutting and pasting the last result into the command):



We can now plot the real part of this function. Change the decimal mode to Standard, and use the following:



The solution is shown below:



Fourier Transforms

Before presenting the concept of Fourier transforms, we'll discuss the general definition of an integral transform. In general, an integral transform is a transformation that relates a function f(t) to a new function F(s) by an integration of the form $F(s) = \int_a^b \kappa(s,t) \cdot f(t) \cdot dt.$ The function κ(s,t) is known as the kernel of the transformation.

The use of an integral transform allows us to resolve a function into a given spectrum of components. To understand the concept of a spectrum, consider the Fourier series

$$f(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cdot \cos \omega_n x + b_n \cdot \sin \omega_n x\right),$$

representing a periodic function with a period T. This Fourier series can be re-written as $f(x) = a_0 + \sum_{n=1}^{\infty} A_n \cdot \cos(\varpi_n x + \phi_n),$ where

$$A_n = \sqrt{a_n^2 + b_n^2}, \quad \phi_n = \tan^{-1}\left(\frac{b_n}{a_n}\right),$$

for n =1,2, …

The amplitudes A_n will be referred to as the spectrum of the function and will be a measure of the magnitude of the component of f(x) with frequency $f_n = n/T$. The basic or fundamental frequency in the Fourier series is $f_0 = 1/T$, thus, all other frequencies are multiples of this basic frequency, i.e., $f_n = n \cdot f_0$. Also, we can define an angular frequency, $\omega_n = 2n\pi/T = 2\pi \cdot f_n = 2\pi \cdot n \cdot f_0 = n \cdot \omega_0$, where ω_0 is the basic or fundamental angular frequency of the Fourier series.

Using the angular frequency notation, the Fourier series expansion is written as

$$f(x) = a_0 + \sum_{n=1}^{\infty} A_n \cdot \cos(\omega_n x + \phi_n).$$

$$= a_0 + \sum_{n=1}^{\infty} \left(a_n \cdot \cos \omega_n x + b_n \cdot \sin \omega_n x\right)$$

A plot of the values A_n vs. ω_n is the typical representation of a discrete spectrum for a function. The discrete spectrum will show that the function has components at angular frequencies ω_n which are integer multiples of the fundamental angular frequency ω_0.

Suppose that we are faced with the need to expand a non-periodic function into sine and cosine components. A non-periodic function can be thought of as having an infinitely large period. Thus, for a very large value of T, the fundamental angular frequency, $\omega_0 = 2\pi/T$, becomes a very small quantity, say $\Delta\omega$. Also, the angular frequencies corresponding to $\omega_n = n \cdot \omega_0 = n \cdot \Delta\omega$, $(n = 1, 2, \ldots, \infty)$, now take values closer and closer to each other, suggesting the need for a continuous spectrum of values.

The non-periodic function can be written, therefore, as

$$f(x) = \int_0^{\infty} [C(\omega) \cdot \cos(\omega \cdot x) + S(\omega) \cdot \sin(\omega \cdot x)] d\omega,$$

where

$$C(\omega) = \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} f(x) \cdot \cos(\omega \cdot x) \cdot dx,$$

and

$$S(\omega) = \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} f(x) \cdot \sin(\omega \cdot x) \cdot dx.$$

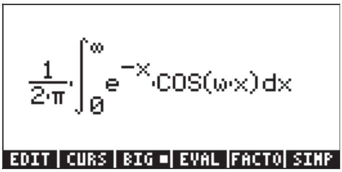
The continuous spectrum is given by

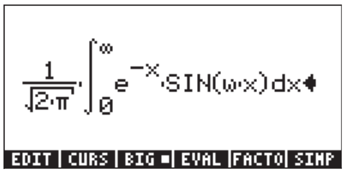
$$A(\omega) = \sqrt{[C(\omega)]^2 + [S(\omega)]^2}$$

The functions C(ω), S(ω), and A(ω) are continuous functions of a variable ω, which becomes the transform variable for the Fourier transforms defined below.

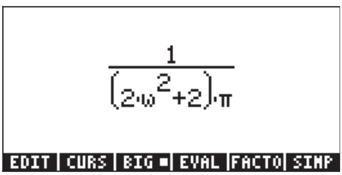
<u>Example 1</u> – Determine the coefficients C(ω), S(ω), and the continuous spectrum A(ω), for the function f(x) = exp(-x), for x > 0, and f(x) = 0, x < 0.

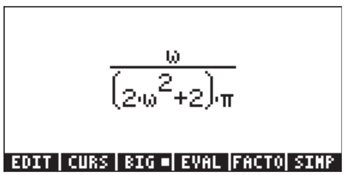
In the calculator, set up and evaluate the following integrals to calculate C(ω) and S(ω), respectively. CAS modes are set to Exact and Real.



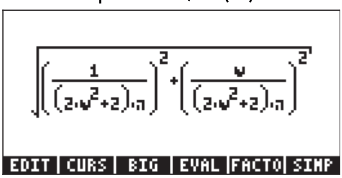


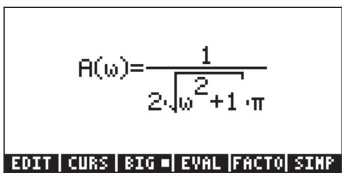
Their results are, respectively:



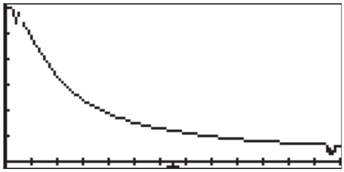


The continuous spectrum, A(ω) is calculated as:





Define this expression as a function by using function DEFINE (← DEF). Then, plot the continuous spectrum, in the range 0 < ω < 10, as:



Definition of Fourier transforms

Different types of Fourier transforms can be defined. The following are the definitions of the sine, cosine, and full Fourier transforms and their inverses used in this Chapter.

Fourier sine transform

$$\mathrm{F}s\{f(t)\} = F(\omega) = \frac{2}{\pi} \cdot \int_0^{\infty} f(t) \cdot \sin(\omega \cdot t) \cdot dt$$

Inverse sine transform

$$\mathrm{F}_s^{-1}\{F(\omega)\} = f(t) = \int_0^{\infty} F(\omega) \cdot \sin(\omega \cdot t) \cdot dt$$

Fourier cosine transform

$$\mathrm{F}c\{f(t)\} = F(\omega) = \frac{2}{\pi} \cdot \int_0^{\infty} f(t) \cdot \cos(\omega \cdot t) \cdot dt$$

Inverse cosine transform

$$\mathrm{F}_c^{-1}\{F(\omega)\} = f(t) = \int_0^{\infty} F(\omega) \cdot \cos(\omega \cdot t) \cdot dt$$

Fourier transform (proper)

$$\mathrm{F}\{f(t)\} = F(\omega) = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\infty} f(t) \cdot e^{-i\omega t} \cdot dt$$

Inverse Fourier transform (proper)

$$\mathrm{F}^{-1}\{F(\omega)\} = f(t) = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\infty} F(\omega) \cdot e^{-i\omega t} \cdot dt$$

Example 1 – Determine the Fourier transform of the function f(t) = exp(- t), for t >0, and f(t) = 0, for t<0.

The continuous spectrum, F(ω), is calculated with the integral:

$$\frac{1}{\sqrt{2\pi}}\int_0^\infty e^{-(1+i\omega)t}dt = \lim_{\varepsilon\to\infty}\frac{1}{\sqrt{2\pi}}\int_0^\varepsilon e^{-(1+i\omega)t}dt$$

$$= \lim_{\varepsilon\to\infty}\frac{1}{\sqrt{2\pi}}\left[\frac{1-\exp(-(1+i\omega)t)}{1+i\omega}\right] = \frac{1}{\sqrt{2\pi}}\cdot\frac{1}{1+i\omega}.$$

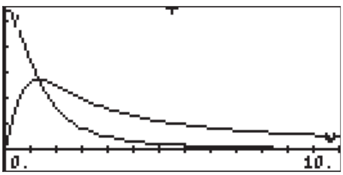
This result can be rationalized by multiplying numerator and denominator by the conjugate of the denominator, namely, 1-iω. The result is now:

$$F(\omega) = \frac{1}{2\pi}\cdot\frac{1}{1+i\omega} = \frac{1}{2\pi}\cdot\left(\frac{1}{1+i\omega}\right)\cdot\left(\frac{1-i\omega}{1-i\omega}\right)$$

$$= \frac{1}{2\pi}\left(\frac{1}{1+\omega^2} - i\cdot\frac{\omega}{1+\omega^2}\right)$$

which is a complex function.

The absolute value of the real and imaginary parts of the function can be plotted as shown below



Notes:
The magnitude, or absolute value, of the Fourier transform, |F(ω)|, is the frequency spectrum of the original function f(t). For the example shown above, $|F(\omega)| = 1/[2\pi(1+\omega^2)]^{1/2}$. The plot of |F(ω)| vs. ω was shown earlier.

Some functions, such as constant values, sin x, exp(x), x^2, etc., do not have Fourier transform. Functions that go to zero sufficiently fast as x goes to infinity do have Fourier transforms.

Properties of the Fourier transform

Linearity: If a and b are constants, and f and g functions, then F{a·f + b·g} = a F{f }+ b F{g}.

Transformation of partial derivatives. Let u = u(x,t). If the Fourier transform transforms the variable x, then

$$F\{\partial u/\partial x\} = i\omega\, F\{u\},\quad F\{\partial^2 u/\partial x^2\} = -\omega^2 F\{u\},$$
$$F\{\partial u/\partial t\} = \partial F\{u\}/\partial t,\ F\{\partial^2 u/\partial t^2\} = \partial^2 F\{u\}/\partial t^2$$

Convolution: For Fourier transform applications, the operation of convolution is defined as

$$(f * g)(x) = \frac{1}{\sqrt{2\pi}} \cdot \int f(x-\xi)\cdot g(\xi)\cdot d\xi.$$

The following property holds for convolution:

$$F\{f*g\} = F\{f\}\cdot F\{g\}.$$

Fast Fourier Transform (FFT)

The Fast Fourier Transform is a computer algorithm by which one can calculate very efficiently a discrete Fourier transform (DFT). This algorithm has applications in the analysis of different types of time-dependent signals, from turbulence measurements to communication signals.

The discrete Fourier transform of a sequence of data values $\{x_j\}$, j = 0, 1, 2, …, n-1, is a new finite sequence $\{X_k\}$, defined as

$$X_k = \frac{1}{n}\sum_{j=0}^{n-1} x_j \cdot \exp(-i\cdot 2\pi kj/n), \qquad k = 0,1,2,\ldots,n-1$$

The direct calculation of the sequence X_k involves n^2 products, which would involve enormous amounts of computer (or calculator) time particularly for large values of n. The Fast Fourier Transform reduces the number of operations to the order of $n\cdot\log_2 n$. For example, for n = 100, the FFT requires about 664 operations, while the direct calculation would require 10,000 operations. Thus, the number of operations using the FFT is reduced by a factor of 10000/664 ≈ 15.

The FFT operates on the sequence $\{x_j\}$ by partitioning it into a number of shorter sequences. The DFT's of the shorter sequences are calculated and later combined together in a highly efficient manner. For details on the algorithm refer, for example, to Chapter 12 in Newland, D.E., 1993, "An Introduction to Random Vibrations, Spectral & Wavelet Analysis – Third Edition," Longman Scientific and Technical, New York.

The only requirement for the application of the FFT is that the number n be a power of 2, i.e., select your data so that it contains 2, 4, 8, 16, 32, 62, etc., points.

Examples of FFT applications

FFT applications usually involve data discretized from a time-dependent signal. The calculator can be fed that data, say from a computer or a data logger, for processing. Or, you can generate your own data by programming a function and adding a few random numbers to it.

Example 1 – Define the function f(x) = 2 sin (3x) + 5 cos(5x) + 0.5*RAND, where RAND is the uniform random number generator provided by the calculator. Generate 128 data points by using values of x in the interval (0,12.8). Store those values in an array, and perform a FFT on the array.

First, we define the function f(x) as a RPN program:

<< → x '2*SIN(3*x) + 5*COS(5*x)' EVAL RAND 5 * + →NUM >>

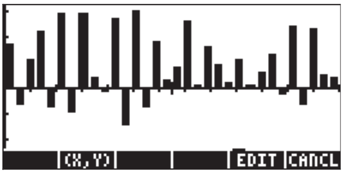
and store this program in variable f. Next, type the following program to generate 2^m data values between a and b. The program will take the values of m, a, and b:

<< → m a b << '2^m' EVAL → n << '(b-a)/(n+1)' EVAL → Dx << 1 n FOR j 'a+(j-1)*Dx' EVAL f NEXT n →ARRY >> >> >> >>

Store this program under the name GDATA (Generate DATA). Then, run the program for the values, m = 5, a = 0, b = 100. In RPN mode, use:

5 SPC 0 SPC 1 0 0 GDATA

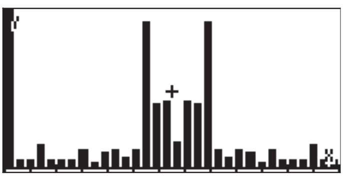
The figure below is a box plot of the data produced. To obtain the graph, first copy the array just created, then transform it into a column vector by using: OBJ→ [/] [+] →ARRY (Functions OBJ→ and →ARRY are available in the command catalog, [→] _CAT). Store the array into variable ΣDAT by using function STOΣ (also available through [→] _CAT). Select Bar in the TYPE for graphs, change the view window to H-VIEW: 0 32, V-VIEW: -10 10, and BarWidth to 1. Press CANCL [ON] to return to normal calculator display.



To perform the FFT on the array in stack level 1 use function FFT available in the MTH/FFT menu on array ΣDAT: FFT FFT. The FFT returns an array of complex numbers that are the arrays of coefficients X_k of the DFT. The magnitude of the coefficients X_k represents a frequency spectrum of the original data. To obtain the magnitude of the coefficients you could transform the array into a list, and then apply function ABS to the list. This is accomplished by using: OBJ→ [EVAL] [◄] →LIST [←] ABS

Finally, you can convert the list back to a column vector to be stored in ΣDAT, as follows: OBJ→ [1] [ENTER] [2] →LIST →ARRY STOΣ

To plot the spectrum, follow the instructions for producing a bar plot given earlier. The vertical range needs to be changed to –1 to 80. The spectrum of frequencies is the following:



The spectrum shows two large components for two frequencies (these are the sinusoidal components, sin (3x) and cos(5x)), and a number of smaller components for other frequencies.

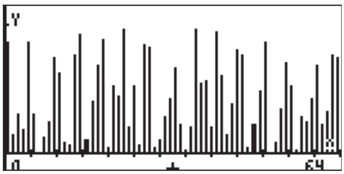
Example 2 – To produce the signal given the spectrum, we modify the program GDATA to include an absolute value, so that it reads:

<< → m a b << '2^m' EVAL → n << '(b-a)/(n+1)' EVAL → Dx << 1 n FOR j 'a+(j-1)*Dx' EVAL f ABS NEXT n →ARRY >> >> >> >>

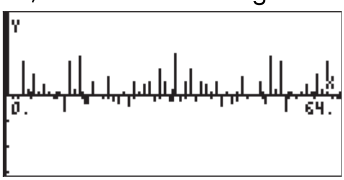
Store this version of the program under GSPEC (Generate SPECtrum). Run the program with m = 6, a = 0, b = 100. In RPN mode, use:

6 SPC 0 SPC 1 0 0 GSPEC

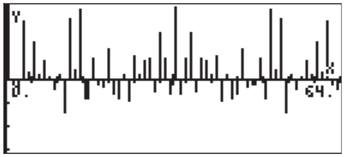
Press ENTER when done, to keep an additional copy of the spectrum array. Convert this row vector into a column vector and store it into ΣDAT. Following the procedure for generating a bar plot, the spectrum generated for this example looks as shown below. The horizontal range in this case is 0 to 64, while the vertical range is –1 to 10:



To reproduce the signal whose spectrum is shown , use function IFFT. Since we left a copy of the spectrum in the stack (a row vector), all you need to do if find function IFFT in the MTH/FFT menu or through the command catalog, ↱ CAT . As an alternative, you could simply type the function name, i.e., type ALPHA ALPHA I F F T ENTER . The signal is shown as an array (row vector) with complex numbers. We are interested only in the real part of the elements. To extract the real part of the complex numbers, use function RE from the CMPLX menu (see Chapter 4), e.g., type ALPHA ALPHA R E ENTER . What results is another row vector. Convert it into a column vector, store it into ΣDAT, and plot a bar plot to show the signal. The signal for this example is shown below, using a horizontal range of 0 to 64, and a vertical range of –1 to 1:



Except for a large peak at t = 0, the signal is mostly noise. A smaller vertical scale (-0.5 to 0.5) shows the signal as follows:



Solution to specific second-order differential equations

In this section we present and solve specific types of ordinary differential equations whose solutions are defined in terms of some classical functions, e.g., Bessel's functions, Hermite polynomials, etc. Examples are presented in RPN mode.

The Cauchy or Euler equation

An equation of the form $x^2 \cdot (d^2y/dx^2) + a \cdot x \cdot (dy/dx) + b \cdot y = 0$, where a and b are real constants, is known as the Cauchy or Euler equation. A solution to the Cauchy equation can be found by assuming that $y(x) = x^n$.
Type the equation as: 'x^2*d1d1y(x)+a*x*d1y(x)+b*y(x)=0' ENTER
Then, type and substitute the suggested solution: 'y(x) = x^n' ENTER SUBST

The result is: 'x^2*(n*(x^(n-1-1)*(n-1)))+a*x*(n*x^(n-1))+b*x^n =0, which simplifies to 'n*(n-1)*x^n+a*n*x^n+b*x^n = 0'. Dividing by x^n, results in an auxiliary algebraic equation: 'n*(n-1)+a*n+b = 0', or.

$$n^2 + (a-1) \cdot n + b = 0 .$$

- If the equation has two different roots, say n_1 and n_2, then the general solution of this equation is $y(x) = K_1 \cdot x^{n_1} + K_2 \cdot x^{n_2}$.
- If $b = (1-a)^2/4$, then the equation has a double root $n_1 = n_2 = n = (1-a)/2$, and the solution turns out to be $y(x) = (K_1 + K_2 \cdot \ln x)x^n$.

Legendre's equation

An equation of the form $(1-x^2) \cdot (d^2y/dx^2) - 2 \cdot x \cdot (dy/dx) + n \cdot (n+1) \cdot y = 0$, where n is a real number, is known as the Legendre's differential equation. Any solution for this equation is known as a Legendre's function. When n is a nonnegative integer, the solutions are called Legendre's polynomials. Legendre's polynomial of order n is given by

$$P_n(x) = \sum_{m=0}^{M} (-1)^m \cdot \frac{(2n-2m)!}{2^n \cdot m! \cdot (n-m)! \cdot (n-2m)!} \cdot x^{n-2m}$$

$$= \frac{(2n)!}{2^n \cdot (n!)^2} \cdot x^n - \frac{(2n-2)!}{2^n \cdot 1! \cdot (n-1)!(n-2)!} \cdot x^{n-2} + \ldots - ..$$

where M = n/2 or (n-1)/2, whichever is an integer.

Legendre's polynomials are pre-programmed in the calculator and can be recalled by using the function LEGENDRE given the order of the polynomial, n. The function LEGENDRE can be obtained from the command catalog (⮣ _CAT) or through the menu ARITHMETIC/POLYNOMIAL menu (see Chapter 5). In RPN mode, the first six Legendre polynomials are obtained as follows:

0 LEGENDRE, result: 1, i.e., $P_0(x) = 1.0$.
1 LEGENDRE, result: 'X', i.e., $P_1(x) = x$.
2 LEGENDRE, result: '(3*X^2-1)/2', i.e., $P_2(x) = (3x^2-1)/2$.
3 LEGENDRE, result: '(5*X^3-3*X)/2', i.e., $P_3(x) =(5x^3-3x)/2$.
4 LEGENDRE, result: '(35*X^4-30*X^2+3)/8', i.e.,
$P_4(x) =(35x^4-30x^2+3)/8$.
5 LEGENDRE, result: '(63*X^5-70*X^3+15*X)/8', i.e.,
$P_5(x) =(63x^5-70x^3+15x)/8$.

The ODE $(1-x^2)\cdot(d^2y/dx^2)-2\cdot x\cdot (dy/dx)+[n\cdot (n+1)-m^2/(1-x^2)] \cdot y = 0$, has for solution the function $y(x) = P_n^m(x)= (1-x^2)^{m/2}\cdot(d^mPn/dx^m)$. This function is referred to as an associated Legendre function.

Bessel's equation

The ordinary differential equation $x^2\cdot(d^2y/dx^2) + x\cdot (dy/dx)+ (x^2-\nu^2) \cdot y = 0$, where the parameter ν is a nonnegative real number, is known as Bessel's differential equation. Solutions to Bessel's equation are given in terms of Bessel functions of the first kind of order ν:

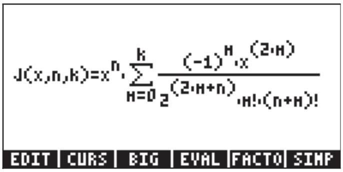
$$J_\nu(x) = x^\nu \cdot \sum_{m=0}^{\infty} \frac{(-1)^m \cdot x^{2m}}{2^{2m+\nu} \cdot m! \cdot \Gamma(\nu+m+1)},$$

where ν is not an integer, and the function Gamma Γ(α) is defined in Chapter 3.

If ν = n, an integer, the Bessel functions of the first kind for n = integer are defined by

$$J_n(x) = x^n \cdot \sum_{m=0}^{\infty} \frac{(-1)^m \cdot x^{2m}}{2^{2m+n} \cdot m! \cdot (n+m)!}.$$

Regardless of whether we use ν (non-integer) or n (integer) in the calculator, we can define the Bessel functions of the first kind by using the following finite series:



Thus, we have control over the function's order, n, and of the number of elements in the series, k. Once you have typed this function, you can use function DEFINE to define function J(x,n,k). This will create the variable [J] in the soft-menu keys. For example, to evaluate $J_3(0.1)$ using 5 terms in the series, calculate J(0.1,3,5), i.e., in RPN mode: [.] [1] [SPC] [3] [SPC] [5] [J] The result is 2.08203157E-5.

If you want to obtain an expression for $J_0(x)$ with, say, 5 terms in the series, use J(x,0,5). The result is

'1-0.25*x^2+0.015625*x^4-4.3403777E-4*x^6+6.782168E-6*x^8-6.78168*x^10'.

For non-integer values ν, the solution to the Bessel equation is given by

$$y(x) = K_1 \cdot J_\nu(x) + K_2 \cdot J_{-\nu}(x).$$

For integer values, the functions Jn(x) and J-n(x) are linearly dependent, since

$$J_n(x) = (-1)^n \cdot J_{-n}(x),$$

therefore, we cannot use them to obtain a general function to the equation. Instead, we introduce the Bessel functions of the second kind defined as

$$Y_\nu(x) = [J_\nu(x) \cos \nu\pi - J_{-\nu}(x)]/\sin \nu\pi,$$

for non-integer ν, and for n integer, with n > 0, by

$$Y_n(x) = \frac{2}{\pi}\cdot J_n(x)\cdot(\ln\frac{x}{2}+\gamma)+\frac{x^n}{\pi}\cdot\sum_{m=0}^{\infty}\frac{(-1)^{m-1}\cdot(h_m+h_{m+n})}{2^{2m+n}\cdot m!\cdot(m+n)!}\cdot x^{2m}$$

$$-\frac{x^{-n}}{\pi}\cdot\sum_{m=0}^{n-1}\frac{(n-m-1)!}{2^{2m-n}\cdot m!}\cdot x^{2m}$$

where γ is the Euler constant, defined by

$$\gamma = \lim_{r\to\infty}[1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{r}-\ln r] \approx 0.57721566490\ldots,$$

and h_m represents the harmonic series

$$h_m = 1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{m}$$

For the case n = 0, the Bessel function of the second kind is defined as

$$Y_0(x) = \frac{2}{\pi}\cdot\left[J_0(x)\cdot(\ln\frac{x}{2}+\gamma)+\sum_{m=0}^{\infty}\frac{(-1)^{m-1}\cdot h_m}{2^{2m}\cdot(m!)^2}\cdot x^{2m}\right].$$

With these definitions, a general solution of Bessel's equation for all values of ν is given by $y(x) = K_1\cdot J_\nu(x)+K_2\cdot Y_\nu(x)$.

In some instances, it is necessary to provide complex solutions to Bessel's equations by defining the Bessel functions of the third kind of order ν as

$$H_n^{(1)}(x) = J_\nu(x)+i\cdot Y_\nu(x), \text{ and } H_n^{(2)}(x) = J_\nu(x)-i\cdot Y_\nu(x),$$

These functions are also known as the first and second Hankel functions of order ν.

In some applications you may also have to utilize the so-called modified Bessel functions of the first kind of order ν defined as $I_\nu(x)= i^{-\nu}\cdot J_\nu(i\cdot x)$, where i is the unit imaginary number. These functions are solutions to the differential equation $x^2\cdot(d^2y/dx^2) + x\cdot (dy/dx)- (x^2+\nu^2) \cdot y = 0$.

The modified Bessel functions of the second kind,

$$K_\nu(x) = (\pi/2)\cdot[I_{-\nu}(x) - I_\nu(x)]/\sin \nu\pi,$$

are also solutions of this ODE.

You can implement functions representing Bessel's functions in the calculator in a similar manner to that used to define Bessel's functions of the first kind, but keeping in mind that the infinite series in the calculator need to be translated into a finite series.

Chebyshev or Tchebycheff polynomials

The functions $T_n(x) = \cos(n\cdot\cos^{-1} x)$, and $U_n(x) = \sin[(n+1)\cos^{-1} x]/(1-x^2)^{1/2}$, n = 0, 1, … are called <u>Chebyshev or Tchebycheff polynomials of the first and second kind</u>, respectively. The polynomials Tn(x) are solutions of the differential equation $(1-x^2)\cdot(d^2y/dx^2) - x\cdot(dy/dx) + n^2\cdot y = 0$.

In the calculator the function TCHEBYCHEFF generates the Chebyshev or Tchebycheff polynomial of the first kind of order n, given a value of n > 0. If the integer n is negative (n < 0), the function TCHEBYCHEFF generates a Tchebycheff polynomial of the second kind of order n whose definition is

$$U_n(x) = \sin(n\cdot\arccos(x))/\sin(\arccos(x)).$$

You can access the function TCHEBYCHEFF through the command catalog (→ CAT).

The first four Chebyshev or Tchebycheff polynomials of the first and second kind are obtained as follows:

0 TCHEBYCHEFF, result: 1,	i.e.,	$T_0(x) = 1.0$.
-0 TCHEBYCHEFF, result: 1,	i.e.,	$U_0(x) = 1.0$.
1 TCHEBYCHEFF, result: 'X',	i.e.,	$T_1(x) = x$.
-1 TCHEBYCHEFF, result: 1,	i.e.,	$U_1(x) = 1.0$.
2 TCHEBYCHEFF, result: '2*X^2-1,	i.e.,	$T_2(x) = 2x^2-1$.
-2 TCHEBYCHEFF, result: '2*X',	i.e.,	$U_2(x) = 2x$.
3 TCHEBYCHEFF, result: '4*X^3-3*X',	i.e.,	$T_3(x) = 4x^3-3x$.
-3 TCHEBYCHEFF, result: '4*X^2-1',	i.e.,	$U_3(x) = 4x^2-1$.

Laguerre's equation

Laguerre's equation is the second-order, linear ODE of the form $x\cdot(d^2y/dx^2) + (1-x)\cdot (dy/dx) + n\cdot y = 0$. Laguerre polynomials, defined as

$$L_0(x) = 1, \quad L_n(x) = \frac{e^x}{n!}\cdot\frac{d^n(x^n\cdot e^{-x})}{dx^n}, n = 1,2,...$$

,

are solutions to Laguerre's equation. Laguerre's polynomials can also be calculated with:

$$L_n(x) = \sum_{m=0}^{n}\frac{(-1)^m}{m!}\cdot\binom{n}{m}\cdot x^m.$$

$$= 1 - n\cdot x + \frac{n(n-1)}{4}\cdot x^2 - ... + ... + \frac{(-1)^n}{n!}\cdot x^n$$

The term

$$\binom{n}{m} = \frac{n!}{m!(n-m)!} = C(n,m)$$

is the m-th coefficient of the binomial expansion $(x+y)^n$. It also represents the number of combinations of n elements taken m at a time. This function is available in the calculator as function COMB in the MTH/PROB menu (see also Chapter 17).

You can define the following function to calculate Laguerre's polynomials:

$$L(x,n) = \sum_{m=0}^{n}\frac{(-1)^m}{m!}\cdot COMB(n,m)\cdot x^m$$

When done typing it in the equation writer press use function DEFINE to create the function L(x,n) into variable L .

To generate the first four Laguerre polynomials use, L(x,0), L(x,1), L(x,2), L(x,3). The results are:

$L_0(x) =$.

$L_1(x) = 1-x$.

$L_2(x) = 1\text{-}2x+ 0.5x^2$
$L_3(x) = 1\text{-}3x+1.5x^2\text{-}0.16666\ldots x^3$.

Weber's equation and Hermite polynomials

Weber's equation is defined as $d^2y/dx^2+(n+1/2\text{-}x^2/4)y = 0$, for n = 0, 1, 2, … A particular solution of this equation is given by the function , $y(x) = \exp(\text{-}x^2/4)H^*(x/\sqrt{2})$, where the function $H^*(x)$ is the Hermite polynomial:

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

In the calculator, the function HERMITE, available through the menu ARITHMETIC/POLYNOMIAL. Function HERMITE takes as argument an integer number, n, and returns the Hermite polynomial of n-th degree. For example, the first four Hermite polynomials are obtained by using:

0 HERMITE, result: 1,	i.e., $H_0^* = 1$.
1 HERMITE, result: '2*X',	i.e., $H_1^* = 2x$.
2 HERMITE, result: '4*X^2-2',	i.e., $H_2^* = 4x^2\text{-}2$.
3 HERMITE, result: '8*X^3-12*X',	i.e., $H_3^* = 8x^3\text{-}12x$.

Numerical and graphical solutions to ODEs

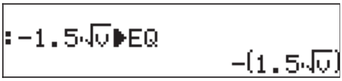
Differential equations that cannot be solved analytically can be solved numerically or graphically as illustrated below.

Numerical solution of first-order ODE

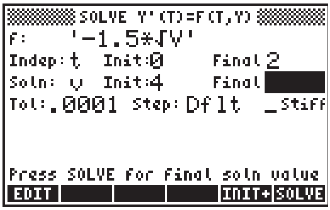
Through the use of the numerical solver (→ NUM.SLV), you can access an input form that lets you solve first-order, linear ordinary differential equations. The use of this feature is presented using the following example. The method used in the solution is a fourth-order Runge-Kutta algorithm preprogrammed in the calculator.

Example 1 -- Suppose we want to solve the differential equation, $dv/dt = \text{-}1.5\ v^{1/2}$, with v = 4 at t = 0. We are asked to find v for t = 2.

First, create the expression defining the derivative and store it into variable EQ. The figure to the left shows the ALG mode command, while the right-hand side figure shows the RPN stack before pressing STO▶.



Then, enter the NUMERICAL SOLVER environment and select the differential equation solver: ↱ NUM.SLV ▼ OK . Enter the following parameters:



To solve, press: SOLVE (wait) EDIT. The result is 0.2499 ≈ 0.25. Press OK.

Solution presented as a table of values

Suppose we wanted to produce a table of values of v, for t = 0.00, 0.25, …, 2.00, we will proceed as follows:

First, prepare a table to write down your results. Write down in your table the step-by-step results:

t	v
0.00	0.00
0.25	
…	…
2.00	

Next, within the SOLVE environment, change the final value of the independent variable to 0.25, use :

▲ .25 OK ▶ ▶ SOLVE (wait) EDIT

(Solves for v at t = 0.25, v = 3.285 ….)

OK INIT+ ▲ . 5 OK ▶ ▶ SOLVE (wait) EDIT

(Changes initial value of t to 0.25, and final value of t to 0.5, solve for v(0.5) = 2.640…)

OK INIT+ ▲ .75 OK ▶ ▶ SOLVE (wait) EDIT

(Changes initial value of t to 0.5, and final value of t to 0.75, solve for v(0.75) = 2.066…)

OK INIT+ ▲ 1 OK ▶ ▶ SOLVE (wait) EDIT
(Changes initial value of t to 0.75, and final value of t to 1, solve for v(1) = 1.562…)
Repeat for t = 1.25, 1.50, 1.75, 2.00. Press OK after viewing the last result in EDIT. To return to normal calculator display, press ON or NXT OK. The different solutions will be shown in the stack, with the latest result in level 1.

The final results look as follows (rounded to the third decimal):

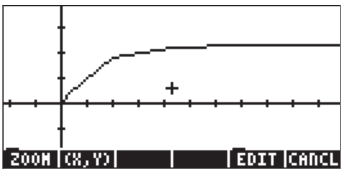
t	v
0.00	4.000
0.25	3.285
0.50	2.640
0.75	2.066
1.00	1.562
1.25	1.129
1.50	0.766
1.75	0.473
2.00	0.250

Graphical solution of first-order ODE

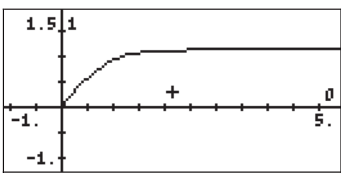
When we can not obtain a closed-form solution for the integral, we can always plot the integral by selecting `Diff Eq` in the `TYPE` field of the PLOT environment as follows: suppose that we want to plot the position x(t) for a velocity function v(t) = $\exp(-t^2)$, with x = 0 at t = 0. We know there is no closed-form expression for the integral, however, we know that the definition of v(t) is dx/dt = $\exp(-t^2)$.

The calculator allows for the plotting of the solution of differential equations of the form Y'(T) = F(T,Y). For our case, we let Y = x and T = t, therefore, F(T,Y) = f(t, x) = $\exp(-t^2)$. Let's plot the solution, x(t), for t = 0 to 5, by using the following keystroke sequence:

- ← 2D/3D (simultaneously, if in RPN mode) to enter PLOT environment
- Highlight the field in front of `TYPE`, using the ▲▼ keys. Then, press CHOOS, and highlight `Diff Eq`, using the ▲▼ keys. Press OK.
- Change field F: to 'EXP(- t^2)'
- Make sure that the following parameters are set to: `H-VAR: 0, V-VAR: 1`
- Change the independent variable to t .
- Accept changes to PLOT SETUP: NXT OK
- ← WIN (simultaneously, if in RPN mode). To enter PLOT WINDOW environment
- Change the horizontal and vertical view window to the following settings: `H-VIEW: -1    5;  V-VIEW: -1     1.5`
- Also, use the following values for the remaining parameters: Init: 0, Final: 5, Step: Default, Tol: 0.0001, Init-Soln: 0
- To plot the graph use: ERASE DRAW



When you observe the graph being plotted, you'll notice that the graph is not very smooth. That is because the plotter is using a time step that may be a bit large for a smooth graph. To refine the graph and make it smoother, use a step of 0.1. Press CANCL and change the *Step* : value to 0.1, then use ERASE DRAW once more to repeat the graph. The plot will take longer to be completed, but the shape is definitely smoother than before. Try the following: EDIT NXT LABEL MENU to see axes labels and range.



Notice that the labels for the axes are shown as 0 (horizontal, for t) and 1 (vertical, for x). These are the definitions for the axes as given in the PLOT SETUP window (← 2D/3D) i.e., H-VAR: 0, and V-VAR: 1. To see the graphical solution in detail use the following:

`NXT` `NXT` `PICT` To recover menu and return to PICT environment.

`(X,Y)` To determine coordinates of any point on the graph.

Use the `◄` `►` keys to move the cursor around the plot area. At the bottom of the screen you will see the coordinates of the cursor as (X,Y), i.e., the calculator uses X and Y as the default names for the horizontal and vertical axes, respectively. Press `NXT` `CANCL` to recover the menu and return to the PLOT WINDOW environment. Finally, press `ON` to return to normal display.

Numerical solution of second-order ODE

Integration of second-order ODEs can be accomplished by defining the solution as a vector. As an example, suppose that a spring-mass system is subject to a damping force proportional to its speed, so that the resulting differential

equation is: $\frac{d^2x}{dt^2} = -18.75 \cdot x - 1.962 \cdot \frac{dx}{dt}$

or, $x'' = -18.75\,x - 1.962\,x'$,

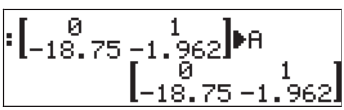
subject to the initial conditions, v = x' = 6, x = 0, at t = 0. We want to find x, x' at t = 2.

Re-write the ODE as: **w**' = **Aw**, where **w** = [x x']^T, and **A** is the 2 x 2 matrix shown below.

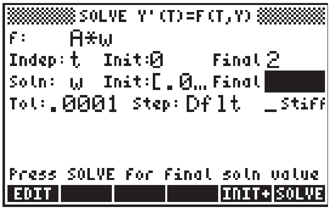
$$\begin{bmatrix} x \\ x' \end{bmatrix}' = \begin{bmatrix} 0 & 1 \\ -18.75 & -1.962 \end{bmatrix} \cdot \begin{bmatrix} x \\ x' \end{bmatrix}$$

The initial conditions are now written as **w** = [0 6]^T, for t = 0. (Note: The symbol [ ]^T means the transpose of the vector or matrix).

To solve this problem, first, create and store the matrix **A**, e.g., in ALG mode:



Then, activate the numerical differential equation solver by using: `→` NUM.SLV `▼` `OK`. To solve the differential equation with starting time t = 0 and final time t = 2, the input form for the differential equation solver should look as follows (notice that the Init: value for the Soln: is a vector [0, 6]):



Press SOLVE (wait) EDIT to solve for w(t=2). The solution reads [.16716… -.6271…], i.e., x(2) = 0.16716, and x'(2) = v(2) = -0.6271. Press CANCL to return to SOLVE environment.

Solution presented as a table of values

In the previous example we were interested only in finding the values of the position and velocity at a given time t. If we wanted to produce a table of values of x and x', for t = 0.00, 0.25, …, 2.00, we will proceed as follows: First, prepare a table to write down your results:

t	x	x'
0.00	0.00	6.00
0.25		
…	…	…
2.00		

Next, within the SOLVE environment, change the final value of the independent variable to 0.25, use:

▲ .25 OK ▶ ▶ SOLVE (wait) EDIT

(Solves for w at t = 0.25, w = [0.968 1.368].)

OK INIT+ ▲ . 5 OK ▶ ▶ SOLVE (wait) EDIT

(Changes initial value of t to 0.25, and final value of t to 0.5, solve again for w(0.5) = [0.748 -2.616])

OK INIT+ ▲ .75 OK ▶ ▶ SOLVE (wait) EDIT

(Changes initial value of t to 0.5, and final value of t to 0.75, solve again for w(0.75) = [0.0147 -2.859])

OK INIT+ ▲ 1 OK ▶ ▶ SOLVE (wait) EDIT

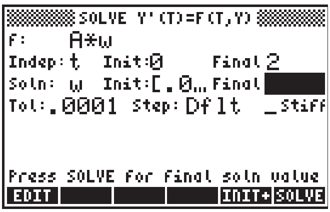
(Changes initial value of t to 0.75, and final value of t to 1, solve again for w(1) = [-0.469 -0.607])
Repeat for t = 1.25, 1.50, 1.75, 2.00. Press OK after viewing the last result in EDIT. To return to normal calculator display, press ON or NXT OK. The different solutions will be shown in the stack, with the latest result in level 1.

The final results look as follows:

t	x	x'		t	x	x'
0.00	0.000	6.000		1.25	-0.354	1.281
0.25	0.968	1.368		1.50	0.141	1.362
0.50	0.748	-2.616		1.75	0.227	0.268
0.75	-0.015	-2.859		2.00	0.167	-0.627
1.00	-0.469	-0.607				

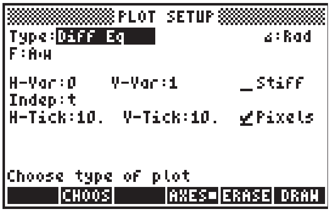
Graphical solution for a second-order ODE

Start by activating the differential equation numerical solver, ↱ NUM.SLV ▼ OK . The SOLVE screen should look like this:



Notice that the initial condition for the solution (Soln: w Init:[0., …) includes the vector [0, 6]. Press NXT OK.

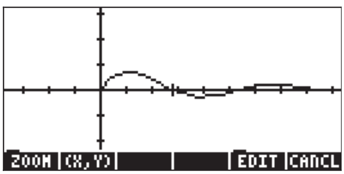
Next, press ↰ 2D/3D (simultaneously, if in RPN mode) to enter the PLOT environment. Highlight the field in front of `TYPE`, using the ▲▼ keys. Then, press CHOOS, and highlight `Diff Eq`, using the ▲▼ keys. Press OK. Modify the rest of the PLOT SETUP screen to look like this:



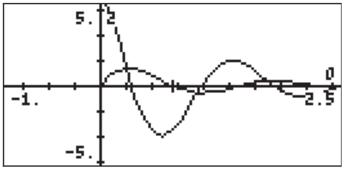
Notice that the option V-Var: is set to 1, indicating that the first element in the vector solution, namely, x', is to be plotted against the independent variable t. Accept changes to PLOT SETUP by pressing NXT OK.

Press ← WIN (simultaneously, if in RPN mode) to enter the PLOT WINDOW environment. Modify this input form to look like this:

To plot the x' vs. t graph use: ERASE DRAW . The plot of x' vs. t looks like this:



To plot the second curve we need to use the PLOT SETUP input form once, more. To reach this form from the graph above use: CANCL NXT OK ← 2D/3D (simultaneously, if in RPN mode) . Change the value of the V-Var: field to 2, and press DRAW (do not press ERASE or you would loose the graph produced above). Use: EDIT NXT LABEL MENU to see axes labels and range. Notice that the x axis label is the number 0 (indicating the independent variable), while the y-axis label is the number 2 (indicating the second variable, i.e., the last variable plotted). The combined graph looks like this:



Press NXT NXT PICT CANCL ON to return to normal calculator display.

Numerical solution for stiff first-order ODE

Consider the ODE: dy/dt = -100y+100t+101, subject to the initial condition y(0) = 1.

Exact solution

This equation can be written as dy/dt + 100 y = 100 t + 101, and solved using an integrating factor, IF(t) = exp(100t), as follows (RPN mode, with CAS set to Exact mode):

‘(100*t+101)*EXP(100*t)’ ENTER ‘t’ ENTER RISCH

The result is ‘(t+1)*EXP(100*t)’.

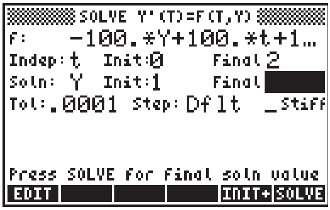
Next, we add an integration constant, by using: ‘C’ ENTER +

Then, we divide by FI(x), by using: ‘EXP(100*t)’ ENTER ÷ .

The result is: ‘((t+1)*EXP(100*t)+C)/EXP(100*t)’, i.e., y(t) = 1+ t +C·e^{100t}. Use of the initial condition y(0) = 1, results in 1 = 1 + 0 + C·e^0, or C = 0, the particular solution being y(t) = 1+t.

Numerical solution

If we attempt a direct numerical solution of the original equation dy/dt = -100y+100t+101, using the calculator’s own numerical solver, we find that the calculator takes longer to produce a solution that in the previous first-order example. To check this out, set your differential equation numerical solver (→ NUM.SLV ▼ OK) to:

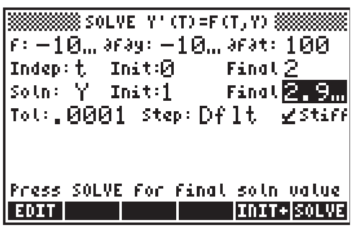


Here we are trying to obtain the value of y(2) given y(0) = 1. With the `Soln: Final` field highlighted, press SOLVE. You can check that a solution takes about 6 seconds, while in the previous first-order example the solution was almost instantaneous. Press ON to cancel the calculation.

This is an example of a stiff ordinary differential equation. A stiff ODE is one whose general solution contains components that vary at widely different rates under the same increment in the independent variable. In this particular case, the general solution, $y(t) = 1+ t +C\cdot e^{100t}$, contains the components 't' and '$C\cdot e^{100t}$', which vary at very different rates, except for the cases C=0 or C≈0 (e.g., for C = 1, t =0.1, $C\cdot e^{100t}$ =22026).

The calculator's ODE numerical solver allows for the solution of stiff ODEs by selecting the option `_Stiff` in the `SOLVE Y'(T) = F(T,Y)` screen. With this option selected you need to provide the values of ∂f/∂y and ∂f/∂t. For the case under consideration ∂f/∂y = -100 and ∂f/∂t = 100.

Enter those values in the corresponding fields of the `SOLVE Y'(T) = F(T,Y)` screen:



When done, move the cursor to the `Soln:Final` field and press SOLVE. This time, the solution in produced in about 1 second. Press EDIT to see the solution: 2.9999999999, i.e., 3.0.

Note: The option `Stiff` is also available for graphical solutions of differential equations.

Numerical solution to ODEs with the SOLVE/DIFF menu

The SOLVE soft menu is activated by using 74 MENU in RPN mode. This menu is presented in detail in Chapter 6. One of the sub-menus, DIFF, contains functions for the numerical solution of ordinary differential equations for use in programming. These functions are described next using RPN mode and system flag 117 set to SOFT menus. (See note in page 16-75).

The functions provided by the SOLVE/DIFF menu are the following:

Function RKF

This function is used to compute the solution to an initial value problem for a first-order differential equation using the Runge-Kutta-Fehlbert 4^{th} -5^{th} order solution scheme. Suppose that the differential equation to be solved is given by dy/dx = f(x,y), with y = 0 at x = 0, and that you will allow a convergence criteria ε for the solution. You can also specify an increment in the independent variable, Δx, to be used by the function. To run this function you will prepare your stack as follows:

3: {'x', 'y', 'f(x,y)'}
2: { ε Δx }
1: x_{final}

The value in the first stack level is the value of the independent variable where you want to find your solution, i.e., you want to find, $y_{final} = f_s(x_{final})$, where $f_s(x)$ represents the solution to the differential equation. The second stack level may contain only the value of ε, and the step Δx will be taken as a small default value. After running function RKF, the stack will show the lines:

2: {'x', 'y', 'f(x,y)'}
1: ε

The value of the solution, y_{final}, will be available in variable . This function is appropriate for programming since it leaves the differential equation specifications and the tolerance in the stack ready for a new solution. Notice that the solution uses the initial conditions x = 0 at y = 0. If, your actual initial solutions are $x = x_{init}$ at $y = y_{init}$, you can always add these values to the solution provided by RKF, keeping in mind the following relationship:

RKF solution		Actual solution	
x	y	x	y
0	0	x_{init}	y_{init}
x_{final}	y_{final}	$x_{init} + x_{final}$	$y_{init} + y_{final}$

The following screens show the RPN stack before and after applying function RKF for the differential equation dy/dx = x+y, ε = 0.001, Δx = 0.1.

After applying function RKF, variable contains the value 4.3880...

Function RRK

This function is similar to the RKF function, except that RRK (Rosenbrock and Runge-Kutta methods) requires as the list in stack level 3 for input not only the names of the independent and dependent variables and the function defining the differential equation, but also the expressions for the first and second derivatives of the expression. Thus, the input stack for this function will look as follows:

3: {'x', 'y', 'f(x,y)' '∂f/∂x' '∂f/∂y' }
2: { ε Δx }
1: x_{final}

The value in the first stack level is the value of the independent variable where you want to find your solution, i.e., you want to find, $y_{final} = f_s(x_{final})$, where $f_s(x)$ represents the solution to the differential equation. The second stack level may contain only the value of ε, and the step Δx will be taken as a small default value. After running function RRK, the stack will show the lines:

2: {'x', 'y', 'f(x,y)' '∂f/∂x' '∂f/vy' }

1: { ε Δx }

The value of the solution, y_{final}, will be available in variable y.

This function can be used to solve so-called "stiff" differential equations.

The following screen shots show the RPN stack before and after application of function RRK:

The value stored in variable y is 3.00000000004.

Function RKFSTEP

This function uses an input list similar to that of function RKF, as well as the tolerance for the solution, and a possible step Δx, and returns the same input list, followed by the tolerance, and an estimate of the next step in the independent variable. The function returns the input list, the tolerance, and the next step in the independent variable that satisfies that tolerance. Thus, the input stack looks as follows:

3: {'x', 'y', 'f(x,y)'}

2: ε

1: Δx

After running this function, the stack will show the lines:

3: {'x', 'y', 'f(x,y)'}

2: ε

1: $(\Delta x)_{next}$

Thus, this function is used to determine the appropriate size of a time step to satisfy the required tolerance.

The following screen shots show the RPN stack before and after application of function RKFSTEP:

These results indicate that $(\Delta x)_{next}$ = 0.34049…

Function RRKSTEP

This function uses an input list similar to that of function RRK, as well as the tolerance for the solution, a possible step Δx, and a number (LAST) specifying the last method used in the solution (1, if RKF was used, or 2, if RRK was used). Function RRKSTEP returns the same input list, followed by the tolerance, an estimate of the next step in the independent variable, and the current method (CURRENT) used to arrive at the next step. Thus, the input stack looks as follows:

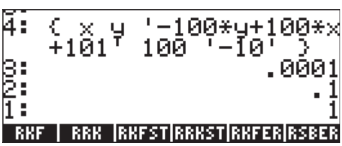
4: {'x', 'y', 'f(x,y)'}
3: ε
2: Δx
1: LAST

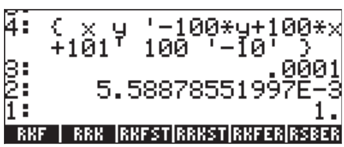
After running this function, the stack will show the lines:

4: {'x', 'y', 'f(x,y)'}
3: ε
2: $(\Delta x)_{next}$
1: CURRENT

Thus, this function is used to determine the appropriate size of a time step ($(\Delta x)_{next}$) to satisfy the required tolerance, and the method used to arrive at that result (CURRENT).

The following screen shots show the RPN stack before and after application of function RRKSTEP:





These results indicate that $(\Delta x)_{next}$ = 0.00558… and that the RKF method (CURRENT = 1) should be used.

Function RKFERR

This function returns the absolute error estimate for a given step when solving a problem as that described for function RKF. The input stack looks as follows:

2: {'x', 'y', 'f(x,y)'}
1: Δx

After running this function, the stack will show the lines:

4: {'x', 'y', 'f(x,y)'}
3: ε
2: Δy
1: error

Thus, this function is used to determine the increment in the solution, Δy, as well as the absolute error (error).

The following screen shots show the RPN stack before and after application of function RKFERR:

These result show that Δy = 0.827… and error = $-1.89\ldots\times 10^{-6}$.

Function RSBERR

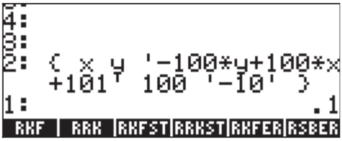
This function performs similarly to RKERR but with the input elements listed for function RRK. Thus, the input stack for this function will look as follows:

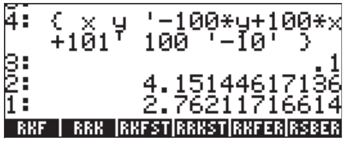
2: {'x', 'y', 'f(x,y)' '∂f/∂x' '∂f/vy' }
1: Δx

After running the function, the stack will show the lines:

4: {'x', 'y', 'f(x,y)' '∂f/∂x' '∂f/vy' }:
3: ε
2: Δy
1: error

The following screen shots show the RPN stack before and after application of function RSBERR:





These results indicate that Δy = 4.1514… and error = 2.762..., for Dx = 0.1. Check that, if Dx is reduced to 0.01, Δy = -0.00307… and error = 0.000547.

Note: As you execute the commands in the DIFF menu values of x and y will be produced and stored as variables in your calculator. The results provided by the functions in this section will depend on the current values of x and y. Therefore, some of the results illustrated above may differ from what you get in your calculator.