LaPlace89

LaPlace + Differential equation solver package version 1.2.4 to TI-89

This package contains functions for LaPlace transformation + solving single or multiple differential equations with constant coefficients. Differential equations can be of any order and complexity. The functions have also the ability to find the solutions of most integral equations or combinations of differential and integral equations (integro-differential equations). Method used: LaPlace transformation.

If you already have LaPlace92 it can be replaced by this package.

Keep the functions together in a separate folder with the name "LAPLACE" and do not create any variable in it.

Functions:

SolveD	solve single differential/integral equations
SimultD	solve multiple simultaneous differential/integral equations
LaPlace	transforms from time to LaPlace domain.
ILaPlace	transforms from LaPlace to time domain.

Programs:

Help	documentation in very short form.
Menu	custom menu

Before using functions, set TI-89 MODE

Complex Format to RECTANGULAR
Angle to RADIAN
Exact/Approx to AUTO

You have to do these settings yourself because; the programs cannot change the mode setting on the calculator.

Help

This program will give online information about and demonstrate the use of functions in this package. When you do not need this program any longer, just delete it.

Syntax: Help()

SolveD

Solving single differential/integral equation. The principle in this function is, first it will transform the equation in to the LaPlace domain and second it solves the equation as a linear equations, third it transforms the solution back to the time-domain (see LaPlace/iLaPlace for further information about LaPlace transformation).

In Principle SolveD can solve differential/integral equations of any order. The only limitation is the size of the calculator's memory (if it is a very complex solution, it may run out of memory).

Equations/initial conditions may contain constants of any kind, but the letter 's' may not be used in any connection.

Heaviside/Dirac delta functions can be used in equation (see LaPlace for further information).

Syntax: *SolvD(equation,{function ,initial conditions})*

Equation	differential/integral equation. A derivative of a function is written: d(f(x),x,n) where "d()" is the normal differentiation function on the calculator and 'n' is the order. Integrals of a function is written: ∫(f(x),x) or d(f(x),x,-n). Where ∫() is the calculators normal integral function.
function	function to solve for f(x).
initial conditions	f(0),f'(0),f''(0),..

Example 1:

Solving second order differential equation:
Equation:

$$\frac{dx(t)^2}{d^2t}+2*\frac{dx(t)}{dt}+5=\sin(2t)\wedge t\geq 0$$

Initial conditions: x'=3 and x=1 at t=0

Solving equation using SolveD:

*SolveD(d(x(t),t,2)+2*d(x(t),t)+5=e^(-t),{x(t),1,3})*

d() is the normal differentiation function on the calculator.

$$x(t)=10-t*e^{-t}-9*e^{-t}-5*t+10$$

Result on the home screen:

Example 2:

Obtain the solution x(t) for t≥0, of the differential equation:

$$\frac{d^2x(t)}{dt^2}+5*\frac{dx}{dt}+6=f(t)$$

Where f(t) is the pulse function

$$F(t)=\begin{cases}3 \text{ for } 0\leq \mathrm{t}<6\\ 0 \text{ else}\end{cases}$$

Initial conditions x(0)=0 and x'(0)=2

First rewriting f(t) to Heaviside functions

f(t) = 3(u(t) - u(t-6))*

Now the equation can be solved with SolvD

Calculator:

*SolveD(d(x(t),t,2)+5*d(x(t),t)+6=3*(u(t)-u(t-6)),{x(t),0,2})*

$$x(t)=\left(\frac{13}{25}-\frac{3*t}{5}-\frac{13*e^{-5t}}{25}\right)*u(t)+\left(\frac{93}{25}-\frac{3*t}{5}-\frac{3*e^{-5t+30}}{25}\right)*u(t-6)$$

Result on the home screen:

Example 3:

Obtain the solution x(t), t>=0, of the integral equation

$$\int x(t)dt+x=\sin(5t)$$

Calculator:

$$SolveD(\int(x(t),t)+x=\sin(5t),\{x(t)\})$$

Result on the home screen:

$$x(t)=\frac{5*\cos(5t)}{26}+\frac{25*\sin(5t)}{26}-\frac{5*e^{-t}}{26}$$

Example 4:

Obtain the solution x(t), t>=0, of the mixed differential/integral equation

$$\int x(t)dt+\frac{dx}{dt}=\cos(t)$$

Initial conditions x(0): unknown

Solving equation on the calculator:

$$SolveD(\int x(t),t)+d(x(t),t)=\cos(t),\{x(t)\})$$

Solution returned on the home screen

$$x(t)=\frac{t*\cos(t)}{t}+x0*\cos(t)+\frac{\sin(t)}{2}$$

Here "x0"= the unknown initial condition

SimultD

Solving systems of simultaneous differential/integral equations. The principle in this function is, first it will transform the equations in to the LaPlace domain and second it solves the equations as a system of linear equations, third it transforms the solutions back to the time-domain (see LaPlace/iLaPlace for further information about LaPlace-transformation).

There are very few rules to obey when using SimultD. First, there has to be an equal number of equations and unknown variables. Second, the variable has to be a function of the type f(var).

Equations do not need to be of same order. In Principle SimultD can solve any number of simultaneous differential/integral equations of any order or mixture of different orders, if there are a sufficient number of equations. The only limitation is the size of the calculator's memory (if it is a very complex solution, it can run out of memory).

Equations/initial conditions may contain constants of any kind, but the letters 's' and 'μ' may not be used in any connection.

Heaviside/Dirac delta functions may be used in equations (see LaPlace for further information).

Syntax:
SimultD([equation;equation;...],
[f1(var),f1(0),f1'(0),..;f2(var),f2(0),f2'(0),..;..])

[equation; equation;..]	differential/integral equations separated by ';'. A derivative of a function is written: d(f(x),x,n) where "d()" is the normal differentiation function on the calculator and 'n' is the order. Integrals of a function is written: ∫(f(x),x) or d(f(x),x,-n). Where ∫() is the calculators normal integral function.
[f1(var),f1(0),..; f2(var),..]	functions and belonging initial conditions separated by ';'.

Example 1.

Solve for t>=0 the system of first-order simultaneous differential equation

$$\begin{cases} \dfrac{dx(t)}{dt} + \dfrac{dy(t)}{dt} + x(t) + 3 * y(t) = e^{-t} \\ 2 * \dfrac{dx(t)}{dt} + \dfrac{dy(t)}{dt} + 5 * x(t) + y(t) = 3 \end{cases}$$

Initial conditions: x=2 and y=1 at t=0

Solving system on the calculator:

Saving the equations to a variable:

*[d(x(t),t)+d(y(t),t)+5*x+3*y=e^(-t); 2*d(x(t),t)+d(y(t),t)+x+y=3]->matx1*

Solving system using SimultD:

SimultD(matx1, [x(t),2;y(t),1])

Result on the home screen

$$\begin{bmatrix} x(t)=\frac{25*e^{t}}{3}-11*e^{-2t}-\frac{9}{2} \\ y(t)=\frac{-25*e^{t}}{2}+\frac{e^{-t}}{2}+\frac{11*e^{-2t}}{2}+\frac{15}{2} \end{bmatrix}$$

ATTENTION when solving equations containing integrals. There are some situations where LaPlace-transformation gives a wrong answer.

1. When an answer from SolveD/SimultD contains Dirac Delta-functions, it may indicate that something is wrong. Use the function Check to see if the solution is correct. If Check return something different from zero the solution may be false. In most cases just remove the Dirac-functions, the rest of the answer will be the correct solution to equation.
2. When the equation contains constants like this: ∫(f(t),t)+f(t)+sin(t)=const. LaPlace transformation will give the solution for: ∫(f(t),t)+f(t)+sin(t)=0.

Remember when interpreting the results from Check:

$$\int d(t)dt = u(t) = 1$$

$$\frac{d}{d^{n}t}\left(d(t-a)\right)^{n} = 0$$

I actually do not know why LaPlace-transformation gives these false solutions. If somebody knows the explanation/solution to this, I would like to know it.

Check

Function constructed to check that the results from SolveD/SimultD are correct. Check will replace the functions in the equations with the output from SolveD/SimultD and return the result.

Check always the results from SolveD/SimultD.

Syntax: *Check(equation, result from SolveD/SimultD)*

equation	has to be exactly the same as past to SimultD/SolveD.
result from SolveD/SimultD	the second parameter has to be the result returned from SolveD/SimultD.

Menu

Creates a Custom Menu with functions included in this package.
After executing the program press '2nd' 'CUSTUM' to activate/deactivate the menu.

Syntax: Menu()

LaPlace

Syntax: LaPlace(f(var), var)
Transforms the expression "f(t)" from time domain to LaPlace domain.

f(var)   can be any expression, which have a LaPlace transform.
var   is the name of the variable to transform normally 't', but can be any legal name.

The expression may contain constants of any kind except ‘μ’, which is reserved for the program.
.

Special transforms:

Unit step function (Heaviside function):
    LaPlace(u(t - a),t) = e^(-a*s)/s

Dirac delta function:
    LaPlace(δ(t - a),t) = e^(-a*s)

You can get ‘δ’ by pressing 'green diamond' + G + D.

iLaPlace

Syntax: iLaPlace(F(var), var):

Transforms F(var) from LaPlace domain to time domain.

F(var)   can be any polynomial. It may contain ln() and arctan().
var   is the name of the variable to transform normally 's', but can be any name.

The expression may contain constants of any kind. The letters ‘μ’ is reserved for the program.

Special transforms:

iLaPlace(e^(-a*s)/s,s) = u(t - a)
iLaPlace(e^(-a*s),s)   = δ(t - a)

iLaPlace may never give an error, if used correct. It shall be able to transform any real polynomial. If this is not the case please report it to
me.

Fold

Syntax: fold(f(var), g(var), var)

Solving convolution integrals f*g.

f(var) and g(var)	can be any expressions, which has a LaPlace transform.
var	is the name of the variable to integrate normally 't', but can be any name.

This function use the fact that, if f(t) and g(t) are of exponential order, piecewise-continuous on t>=0 and have LaPlace transforms F(s) and G(s) respectively, then, for Re(s)>0 f*g=InvL{F(s)G(s)}

Example:

```
f(t) = t*u(t)
g(t) = sin(2t)*u(t)
```

To solve f*g write following on the commandline:

```
fold(t*u(t),sin(2t)*u(t),t)
```

This gives the result on the home screen:

$$\frac{t}{2}-\frac{\sin(2t)}{4}$$

About

Programs of same type:

Type	Name
Inverse Z-transformation	inverseZ
Fourier/inverse Fourier transformation	Fourier92

Please send questions and comments to Lars Frederiksen or Roberto Peres-Franco.

Author: Lars Frederiksen
E-mail: ASK@POST8.TELE.DK
Supporter on the TI-89 version: Roberto Perez-Franco
E-mail: Perez-Franco@iname.com

Thanks to Roberto Perez-Franco for his great support on the TI-89 version.

PS. Please do not ask for more programs.