

## Part IV. Linear Differential Equations

### Section 5. The Laplace Transform

In this section, examples show how Maple's Laplace transform procedures are used to obtain transforms and their inverses. Applications to differential equations with discontinuous forcing functions are featured. See Simmons/Krantz, Chapter 7.

#### *The **inttrans** package*

The standard integral transform procedures of applied mathematics are contained in Maple's **inttrans** package.

> **with(inttrans);**

[*addtable, fourier, fouriercos, fouriersin, hankel, hilbert, invfourier, invhilbert, invlaplace, invmellin, laplace, mellin, savetable*] (1)

We are interested in the procedures named **laplace** and **invlaplace**. The syntax to generate the Laplace transform of an expression  $f(t)$  as an expression  $F(s)$  is

`laplace( f(t), t, s )`

The inverse Laplace transform of  $F(s)$  to  $f(t)$  is accomplished with

`invlaplace( F(s), s, t )`

Four textbook examples:

1. The Laplace transform of  $t e^{2t} \cos(3t)$

> **laplace( t\*exp(2\*t)\*cos(3\*t), t, s );**

$$\frac{s^2 - 4s - 5}{(s^2 - 4s + 13)^2} \quad (2)$$

2. The inverse Laplace transform of  $\frac{s^2 - 1}{s^3 + 8}$

> **invlaplace( (s^2-1)/(s^3+8), s, t);**

$$\frac{1}{4} e^{-2t} + \frac{1}{12} \left( 9 \cos(\sqrt{3} t) - \sqrt{3} \sin(\sqrt{3} t) \right) e^t \quad (3)$$

3. Add this to your Laplace transform table: The transform of  $t e^{at} \sin(kt)$

> **laplace( t\*exp(a\*t)\*sin(k\*t), t, s);**

$$\frac{2 k (s - a)}{\left( (s - a)^2 + k^2 \right)^2} \quad (4)$$

4. The transform of a discontinuous time function:  $\cos(t) + 3t^2 - \text{piecewise}(t < 3, 0, 1)$

> **laplace( cos(t) + 3\*t^2 - piecewise( t<3, 0, 1), t, s);**

$$\frac{s}{s^2 + 1} + \frac{6}{s^2} - \frac{e^{-3s}}{s} \quad (5)$$

"piecewise"? What is that all about?

Answer: The **piecewise** procedure is used to define functions that are piecewise continuous. The entry

```
piecewise( t<3, 0, 1)
```

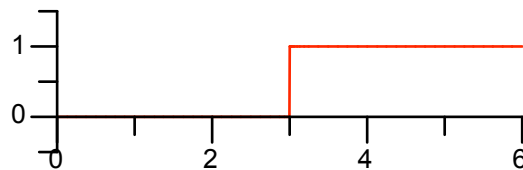
defines the function that, for  $t < 3$  has the value 0 and has the value 1 otherwise. It prettyprints using the "stacked" notation used in textbooks.

```
> piecewise(t<3,0,1);
```

$$\begin{cases} 0 & t < 3 \\ 1 & otherwise \end{cases} \quad (6)$$

See how it graphs.

```
> plots[setoptions]( tickmarks = [5,5] );
> plot( piecewise( t<3, 0, 1), t=0..6, -0.5..1.5);
```



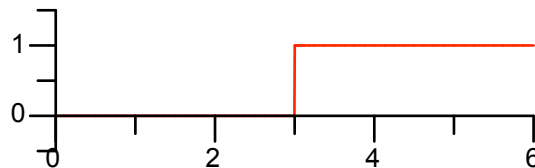
### ***Piecewise versus Heaviside: Both are winners***

If a piecewise function appears as part of an inverse Laplace transform, Maple uses the unit step function in the formula. This function is named **Heaviside** in honor of the man who was among the first to use it in engineering applications, Oliver Heaviside.

In Heaviside notation the function `piecewise(t<3,0,1)` is

```
Heaviside(t-3)
```

```
> plot( Heaviside(t-3), t=0..6, -0.5..1.5);
```



Maple's **convert** procedure can be used to convert from one form to the other.

```
> convert(piecewise(t<3,0,1), Heaviside);
Heaviside(t-3) \quad (7)
```

```
> convert(Heaviside(t-3), piecewise);
```

(8)

$$\begin{cases} 0 & t < 3 \\ \text{undefined} & t = 3 \\ 1 & 3 < t \end{cases} \quad (8)$$

Observe that Maple's version of the Heaviside function is undefined at the number 0. This can be changed if desired. For example, some applications require  $\text{Heaviside}(0) = 1$ . If this is the case, simply enter

```
> Heaviside(0) := 1;
```

$$\text{Heaviside}(0) := 1 \quad (9)$$

Maple makes note of this value in a special table reserved for function values and will use it in all subsequent calculations. Compare the following output to output (8) above.

```
> convert(Heaviside(t-3), piecewise);
```

$$\begin{cases} 0 & t < 3 \\ 1 & 3 \leq t \end{cases} \quad (10)$$

Of course, it makes no difference in the graphs (almost).

**Warning:** It does make a difference when plotting points. Moreover, the fact that default  $\text{Heaviside}(0)$  is undefined can cause problems in the plotting of other graphs. Maple's plotter sometimes stops dead when it encounters "undefined". You will know it when you see it. Therefore, we will leave the value of  $\text{Heaviside}(0)$  at 1.

Be on the lookout for *Heaviside* when inverting an s-domain function containing expressions the form  $e^{-as}$ , where the constant  $a$  positive.

```
> invlaplace( s/(s^2+1)+6/s^2-exp(-3*s)/s, s, t);
```

$$\cos(t) + 6t - \text{Heaviside}(t-3) \quad (11)$$

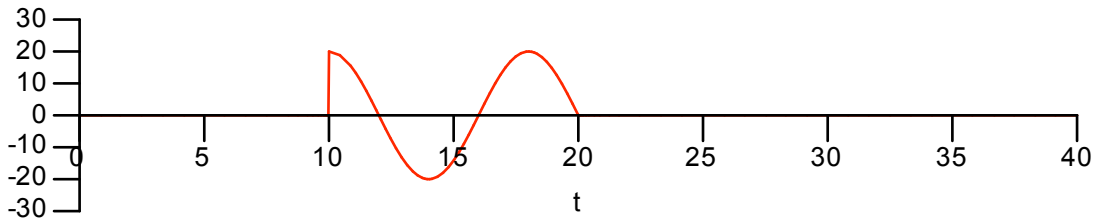
Be prepared to use Heaviside when dealing with piecewise continuous functions. They arise often in engineering and science applications.

### **Typical driver: Off, then On, then Off again**

Suppose, for example, an undamped mass spring system is set into motion from equilibrium with velocity 3 meters per second, then driven with a sinusoid for the 10 second interval between  $t = 10$  and  $t = 20$ . See the graph below.

```
> f := t -> (Heaviside(t-10) - Heaviside(t-20))*20*sin(Pi/4*t);
plot( f(t), t=0..40, -30..30,
      tickmarks=[[5*k$ k=0..8], [10*k$ k=-3..3]]);
```

$$f := t \rightarrow 20 (\text{Heaviside}(t-10) - \text{Heaviside}(t-20)) \sin\left(\frac{1}{4} \pi t\right)$$



Here is the Laplace transform of the driver.

> `laplace(f(t), t, s);`

$$\frac{80 \left( 4 e^{-10s} s + e^{-20s} \pi \right)}{16 s^2 + \pi^2} \quad (12)$$

For the forced differential equation we will use  $m = 40$  (kilograms),  $c = 0$ ,  $k = 10$ .

> `DE := 40*diff(y(t),t,t) + 10*y(t) = f(t);`

$$DE := 40 \left[ \frac{d^2}{dt^2} y(t) \right] + 10 y(t) = 20 \left( \text{Heaviside}(t - 10) - \text{Heaviside}(t - 20) \right) \sin \left[ \frac{1}{4} \pi t \right] \quad (13)$$

The `dsolve` solution to the IVP is obtained first.

> `soln := dsolve( {DE, y(0)=0, D(y)(0)=3}, y(t) );`

$$\begin{aligned} \text{soln} := y(t) = & 6 \sin \left[ \frac{1}{2} t \right] + \frac{1}{-4 + \pi^2} \left[ 4 \text{Heaviside}(t - 20) \pi \sin \left[ \frac{1}{2} t - 10 \right] \right. \\ & + 8 \text{Heaviside}(t - 10) \cos \left[ \frac{1}{2} t - 5 \right] \\ & \left. - 8 \left( \text{Heaviside}(t - 10) - \text{Heaviside}(t - 20) \right) \sin \left[ \frac{1}{4} \pi t \right] \right] \end{aligned} \quad (14)$$

- `dsolve` did *not* use the method of Laplace transforms to solve this equation.

Question: What method did `dsolve` use?

Answer: `dsolve` sees that the equation has constant coefficients, solves the homogeneous equation, then uses the method of variation of parameters to obtain a particular solution.

Question: How do you know?

Answer: I set the `infolevel` for `dsolve` to 3, like this. (The default `infolevel` for all procedures is 0.)

> `infolevel[dsolve] := 3;`

$$\text{infolevel}_{\text{dsolve}} := 3 \quad (15)$$

Then I ran the `dsolve` procedure again, suppressing output, and got the following information.

> `soln := dsolve( {DE, y(0)=0, D(y)(0)=3}, y(t) );`

```
Methods for second order ODEs:
--- Trying classification methods ---
trying a quadrature
trying high order exact linear fully integrable
```

```

trying differential order: 2; linear nonhomogeneous with symmetry
[0,1]
trying a double symmetry of the form [xi=0, eta=F(x)]
-> Try solving first the homogeneous part of the ODE
    checking if the LODE has constant coefficients
    <- constant coefficients successful
    -> Determining now a particular solution to the non-homogeneous
ODE
    building a particular solution using variation of parameters
<- solving first the homogeneous part of the ODE successful
That's how I know what dsolve did. And that is how you can discover some of the details about most of
Maple's procedures.

```

Before continuing we set the **infolevel** for **dsolve** back to level 0. Too much information is not always useful.

```
> infolevel[dsolve] := 0;
```

Question: Well then, can Maple apply the method of Laplace transforms using **dsolve**?

Answer: Sure, simply insert the unknown function right after {DE, inits}, and add the equation

```
method=laplace
```

```
> laplace_soln := dsolve( {DE, y(0)=0, D(y)(0)=3}, y(t), method=
laplace );
```

$$\begin{aligned}
 \text{laplace\_soln} := y(t) = & 6 \sin\left(\frac{1}{2} t\right) + \frac{4 \operatorname{Heaviside}(t-20) \left( \pi \sin\left(\frac{1}{2} t - 10\right) + 2 \sin\left(\frac{1}{4} \pi t\right) \right)}{-4 + \pi^2} \\
 & + \frac{8 \operatorname{Heaviside}(t-10) \left( \cos\left(\frac{1}{2} t - 5\right) - \cos\left(\frac{1}{4} \pi (t-10)\right) \right)}{-4 + \pi^2}
 \end{aligned} \tag{16}$$

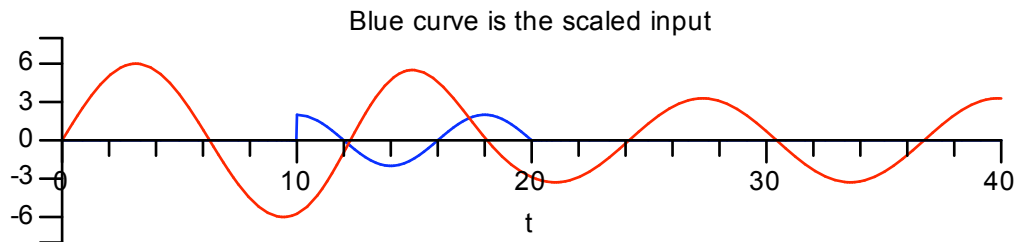
To see the effect of using the Laplace transform method, compare this form of the solution to the form obtained earlier. You might like to convert the first solution formula to piecewise form for easy interpretation.

```
> convert(soln, piecewise);
```

$$y(t) = \begin{cases} 6 \sin\left(\frac{1}{2} t\right) & t \leq 10 \\ 6 \sin\left(\frac{1}{2} t\right) + \frac{8 \cos\left(\frac{1}{2} t - 5\right) - 8 \sin\left(\frac{1}{4} \pi t\right)}{-4 + \pi^2} & 10 < t \leq 20 \\ 6 \sin\left(\frac{1}{2} t\right) + \frac{4 \pi \sin\left(\frac{1}{2} t - 10\right) + 8 \cos\left(\frac{1}{2} t - 5\right)}{-4 + \pi^2} & 20 < t \end{cases} \tag{17}$$

The graph of the solution below reveals that the input between 10 and 20 (shown scaled down to fit and in blue) caused the amplitude of the oscillations to decrease from 6 to 4 meters (approximately). The phase of the oscillations has also been affected, but not the period.

```
> plot([rhs(soln),f(t)/10], t=0..40, -8..8, color=[red,blue],
      title="Blue curve is the scaled input",
      ytickmarks=[3*k$ k=-2..2]);
```



The default **plot** failed to put any values on the tickmarks on the vertical axis. The equation

```
ytickmarks=[3*k$ k=-2..2]
```

was used to place the vertical tickmarks you see in this plot.

### Use an alias

Typing the word Heaviside in the input eventually gets to be quite a chore. Why can't we just use an H? Well, we can. The following entry tells Maple that from now on (in this session) every time we type H what we mean is Heaviside.

```
> alias(H=Heaviside);
```

*H* (18)

Maple will also use H for Heaviside in the display of its outputs.

```
> laplace_soln;
```

$$y(t) = 6 \sin\left(\frac{1}{2} t\right) + \frac{4 H(t-20) \left(2 \sin\left(\frac{1}{4} \pi t\right) + \pi \sin\left(\frac{1}{2} t-10\right)\right)}{-4 + \pi^2} \quad (19)$$

$$+ \frac{8 H(t-10) \left(-\cos\left(\frac{1}{4} \pi (t-10)\right) + \cos\left(\frac{1}{2} t-5\right)\right)}{-4 + \pi^2}$$

### The Dirac delta: Dirac

The Dirac delta function is written as **Dirac** in Maple.

```
> laplace(Dirac(t-3), t, s);
```

$e^{-3s}$  (20)

```
> invlaplace(1, t, s);
```

Dirac(*s*) (21)

You might like to use a delta, like your textbook probably does. No problem, make an alias.

```
> alias(delta=Dirac);
```

*H*,  $\delta$  (22)

The output is a sequence listing all of the aliases in effect at the present time in the session.

Lets drive the mass spring system introduced above with a blow of magnitude 200 in the positive direction at time  $t = 12$ .

```
> DE2 := lhs(DE) = 200*delta(t-12);
```

$$DE2 := 40 \left[ \frac{d^2}{dt^2} y(t) \right] + 10 y(t) = 200 \delta(t - 12) \quad (23)$$

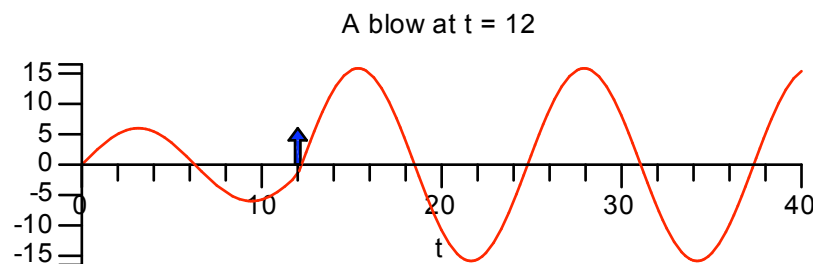
The solution formula, using the same initial conditions and the method of Laplace transforms, looks like this.

```
> lap_soln2 := dsolve( {DE2, y(0)=0,D(y)(0)=3},y(t),method=laplace);
```

$$lap\_soln2 := y(t) = 6 \sin\left[\frac{1}{2} t\right] + 10 H(t - 12) \sin\left[\frac{1}{2} t - 6\right] \quad (24)$$

The following plot displays the solution curve (output) and a scaled arrow placed at the time of the application of the blow (input). See the `plots[arrow]` help page to learn about arrows in plots.

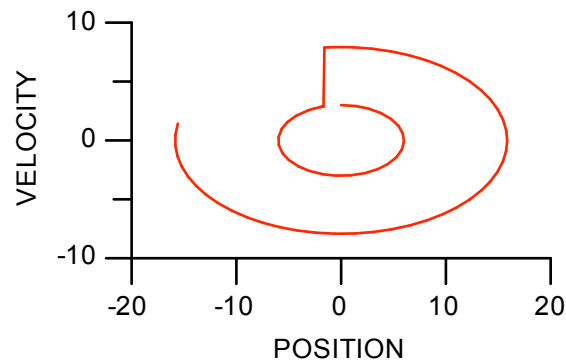
```
> with(plots):
input := arrow([12,0],[0,6],width=0.3, head_width=1,
               head_length=2, color=blue):
output := plot(rhs(lap_soln2), t=0..40, -16.5..16.5):
display( input, output, title="A blow at t = 12",
         ytickmarks=[5*k$ k=-3..3]);
```



The effect of the blow is to cause a kink in the curve for position as a function of time. There is also a jump discontinuity on the curve for velocity.

The phase plane plot shows the discontinuity in velocity as a vertical line joining the initial elliptical trajectory to the elliptical trajectory describing the motion of the system after the blow. Since the vertical axis in phase space records velocity this is always what will happen when a Dirac delta is applied to a system.

```
> plot( [rhs(lap_soln2),diff(rhs(lap_soln2),t),t=0..22],
        view=[-20..20,-10..10], axes=framed,
        labels=["POSITION","VELOCITY"],
        labeldirections=[horizontal,vertical]);
```



### Applying the Laplace transform method to a linear system

The equation `method=laplace` can also be added to `dsolve` when it is applied to a system of linear equations (constant coefficients), driven or undriven. However, if the system is driven by discontinuous forcing functions, the Laplace transform solution formulas may contain complex exponentials.

Consider the following IVP

$$x' = x - y + 10 H(t - 2)$$

$$y' = x + y$$

$$x(0) = 1, \quad y(0) = 1$$

```
> sys := diff(x(t),t) = x(t) - y(t) + 10*H(t-2),
      diff(y(t),t) = x(t) + y(t);
      inits := x(0)=1, y(0)=1;
```

$$\text{sys} := \frac{d}{dt} x(t) = x(t) - y(t) + 10 H(t - 2), \quad \frac{d}{dt} y(t) = x(t) + y(t)$$

$$\text{inits} := x(0) = 1, y(0) = 1 \quad (25)$$

Maple's Laplace transform solution contains expressions in complex form.

```
> lap_soln := dsolve( {sys,inits}, [x(t),y(t)], method=laplace);
```

$$\text{lap\_soln} := \left\{ x(t) = \frac{5}{2} \left( -2 + 4 e^{t-2} \sin(t-2) + (1 + I) e^{(1+I)(t-2)} \right. \right. \quad (26)$$

$$\left. + (1 - I) e^{(1-I)(t-2)} \right) H(t-2) + e^t (\cos(t) - \sin(t)),$$

$$y(t) = \left\{ \frac{5}{2} + \frac{5}{2} I \right\} H(t-2) \left( -e^{(1+I)(t-2)} + I e^{(1-I)(t-2)} + 1 - I \right) + e^t (\cos(t)$$

$$+ \sin(t)) \left. \right\}$$

The solution obtained using the default variation of parameters method does not.

```
> soln := dsolve( {sys,inits}, [x(t),y(t)]);
```

$$\text{soln} := \left\{ y(t) = e^t \sin(t) + e^t \cos(t) + 5 H(t-2) \right. \quad (27)$$

$$\left. + 5 H(t-2) e^{t-2} \sin(t-2) - 5 H(t-2) e^{t-2} \cos(t-2), \right.$$

$$x(t) = e^t \cos(t) - e^t \sin(t) + 5 H(t-2) e^{t-2} \cos(t-2) + 5 H(t-2) e^{t-2} \sin(t-2) - 5 H(t-2)$$

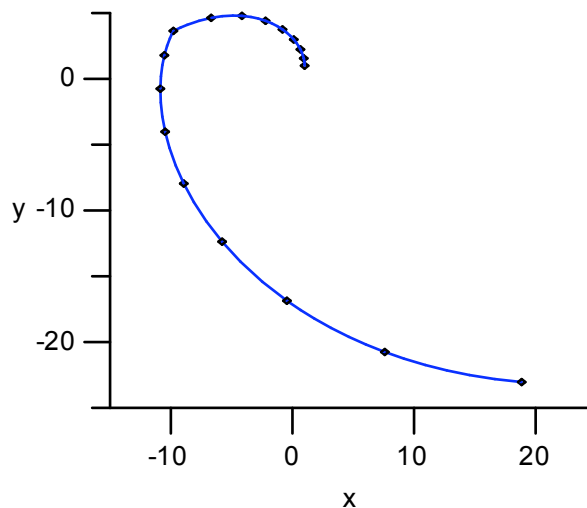
Plot the solution trajectory

The following entry shows how to get the *soln* into trajectory form for plotting. We make it into a vector expression. The messy output is suppressed.

```
> v := subs(soln, <x(t), y(t)>):
```

The trajectory *v* is now plotted along with a point every quarter of a second (assuming time is in seconds). See if you can determine the precise point where the Heaviside driver starts to have an effect on the output.

```
> Pts := [eval([v[1], v[2]], t=k/4) $ k=0..16]:
plot([v[1], v[2], t=0..4], Pts, x=-15..25, y=-25..5,
style=[line, point], color=[blue, black], axes=framed);
```



The last entry maps the **collect** procedure into the solution vector *v*, first collecting all terms with the Heaviside function, *H*, as a factor, then collecting the exponential terms. The output shows the two solution functions in simplified form.

```
> map(collect, v, [H, exp]);
```

$$\begin{bmatrix} (-5 + (5 \cos(t-2) + 5 \sin(t-2)) e^{t-2}) H(t-2) + e^t (\cos(t) - \sin(t)) \\ (5 + (5 \sin(t-2) - 5 \cos(t-2)) e^{t-2}) H(t-2) + e^t (\cos(t) + \sin(t)) \end{bmatrix} \quad (28)$$