In [1]:

```
Autosave disabled
```

Chapter 24: Solutions of Linear Differential Equations with Constant Coefficients by Laplace Transform

Introduction to a far reaching system for simplification of the ODE solving process. The transform is from the time domain to the frequency domain, where algebraic operations are performed before inverse transportation back to the time domain. The main variable governing the frequency domain is s, and the time domain t.

When a Laplace transform is performed, it should be accomplished cleanly, with no leftover x or y factors. The presence of any such is an ill omen, because it can involve complexities. Like hunting through the transform table trying to make things come out right. In this notebook the first attempt will be without Laplace. Then Laplace will be brought in as necessary.

Whereas functions got Laplace-transformed or Laplace-inverse-transformed in previous chapters, this chapter, for the first time, includes ODEs which see transformations in both directions within the same problem. This set of 'complete' cases includes Problems 24.7a, 24.8, 24.9, and 24.10.

24.1 Solve $y' - 5y = 0; \quad y(0) = 2.$

This problem can be entered into Wolfram Alpha:

① !| y' - 5 y = 0, y(0) = 2 |!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$2e^{5x}$

24.2 Solve $y' - 5y = e^{5x}; \quad y(0) = 0.$

This problem can be entered into Wolfram Alpha:

!| y' - 5 y = e^(5 * x), y(0) = 0 |!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$y(x) = e^{5x}\, x$

24.3 Solve $y' + y = \sin x; \quad y(0) = 1.$

This problem can be entered into Wolfram Alpha:

!| y' + y = sin(x), y(0) = 1 |!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$y(x) = \frac{1}{2}(3\, e^{-x} + \sin(x) - \cos(x))$

24.4 Solve $y'' + 4y = 0; \quad y(0) = 2, \; y'(0) = 2.$

This problem can be entered into Wolfram Alpha:

![y'' + 4 y = 0, y(0) = 2, y'(0) = 2]!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$y(x) = \sin(2x) + 2\cos(2x))$

24.5 Solve $y'' - 3y' + 4y = 0; \quad y(0) = 1,\ y'(0) = 5.$

This problem can be entered into Wolfram Alpha:

![y'' - 3 y' + 4 y = 0, y(0) = 1, y'(0) = 5]!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$$y(x) = e^{3x/2}\left(\sqrt{7}\sin\left(\frac{\sqrt{7}x}{2}\right) + \cos\left(\frac{\sqrt{7}x}{2}\right)\right)$$

24.6 Solve $y'' - y' - 2y = 4x^2; \quad y(0) = 1,\ y'(0) = 4.$

This problem can be entered into Wolfram Alpha:

![y'' - y' - 2 y = 4 x^2, y(0) = 1, y'(0) = 4]!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$y(x) = -2x^2 + 2x + 2e^{-x} + 2e^{2x} - 3$

24.7 Solve $y'' + 4y' + 8y = \sin x; \quad y(0) = 1,\ y'(0) = 0.$

This problem can be entered into Wolfram Alpha:

![y'' + 4 y' + 8 y = sin(x), y(0) = 1, y'(0) = 0]!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

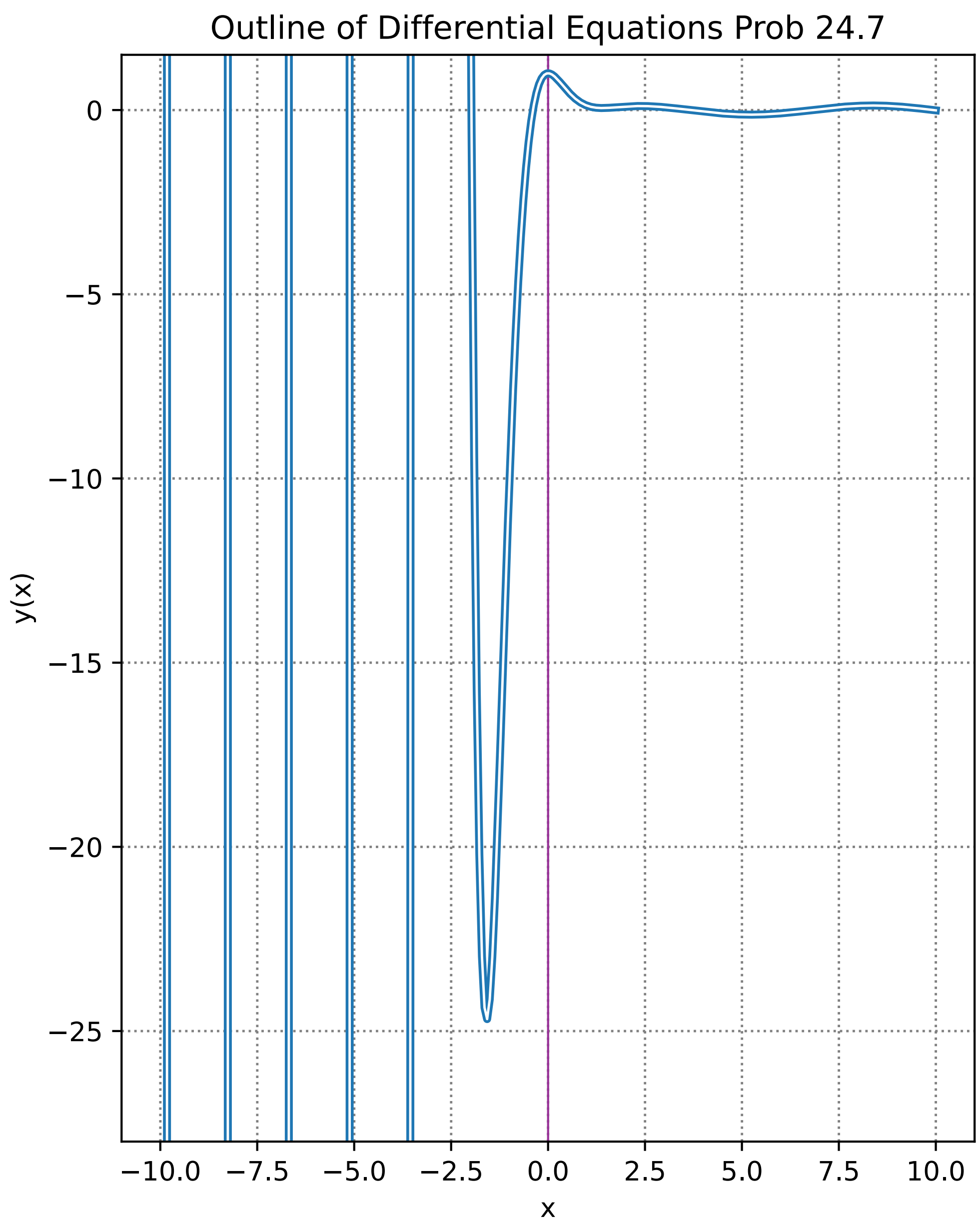
$$y(x) = \frac{1}{65}e^{-2x}\left(7e^{2x}\sin(x) + 69\cos(2x) + (131\sin(x) - 4e^{2x})\cos(x)\right)$$

The form assumed by the text answer versus the above W|A answer were somewhat tricky to reconcile, so a plot appears below, demonstrating their equality.

In [50]:

```
import numpy as np
import matplotlib.pyplot as plt

%config InlineBackend.figure_formats = ['svg']

#x = np.arange(0, 3., 0.005)
x = np.linspace(-10,10,300)
y3 = (1/65)*np.exp(-2*x)*(7*np.exp(2*x)*np.sin(x) + 69*np.cos(2*x) + (131*np.sin(x) - 4*np.exp(2*
y4 = (np.exp(-2*x))*((69/65)*np.cos(2*x) + (131/130)*np.sin(2*x)) + (7/65)*np.sin(x) - (4/65)*np.


plt.grid(True, linestyle='dotted', color='gray', linewidth=0.9)
plt.xlabel("x")
plt.ylabel("y(x)")
plt.title("Outline of Differential Equations Prob 24.7")
plt.rcParams['figure.figsize'] = [9, 7.5]


ax = plt.gca()
#ax.axhline(y=0, color='#993399', linewidth=0.8)
ax.axvline(x=0, color='#993399', linewidth=0.8)
ratio = 0.95
xleft, xright = ax.get_xlim()
```

```
ybottom, ytop = ax.get_ylim()
ax.set_aspect(abs((xright-xleft)/(ybottom-ytop))*ratio)


#plt.text(965, -2.5e43, "-np.log(abs((np.cos(x/2))**2 - (np.sin(x/2))**2)))
#plt.text(965, 2.25e43, "SOLN: y = 3*np.exp(x**2) + 1/2", size=10,\
        # bbox=dict(boxstyle="square", ec=('#8C564B'),fc=(1., 1., 1),))
plt.ylim(-28,1.5)
plt.plot(x, y3, linewidth = 3)
plt.plot(x, y4, linewidth = 0.9, color = 'w')
plt.show()
```



24.7a

Solve the IVP

$$y'' + 3y' + 2y = f(t), \qquad y(0) = 2, \; y'(0) = 3$$

when the right-hand function equals

$$f(t) = \begin{cases} 1 & t < 3 \\ t-2 & 3 < t < 6 \\ 2 & t > 6 \end{cases}$$

Note: this problem (performed in Matlab) can be seen at the following address: https://www.math.umd.edu/~petersd/246h/matlablaplace.html (https://www.math.umd.edu/~petersd/246h/matlablaplace.html)

To understand how the author puts the piecewise form into a general equation takes a little bit of Heaviside step function knowledge. To write the rhs in a general form would look like the following:

$$f(t) = f_1(t) + (f_2(t) - f_1(t))\theta(t - t_1) + (f_3(t) - f_2(t))\theta(t - t_2)$$

where the subscripts on f correspond to the piecewise function pieces, numbered from top to bottom, and the symbol θ represents the Heaviside function.

Most of the calculation for the problem can be done by Maxima (replacing the original Matlab procedure.) The wxMaxima worksheet section follows the procedure on the Maxima tutorial page, (except that in the tutorial there is no need to insert a rhs piecewise $f(t)$): https://

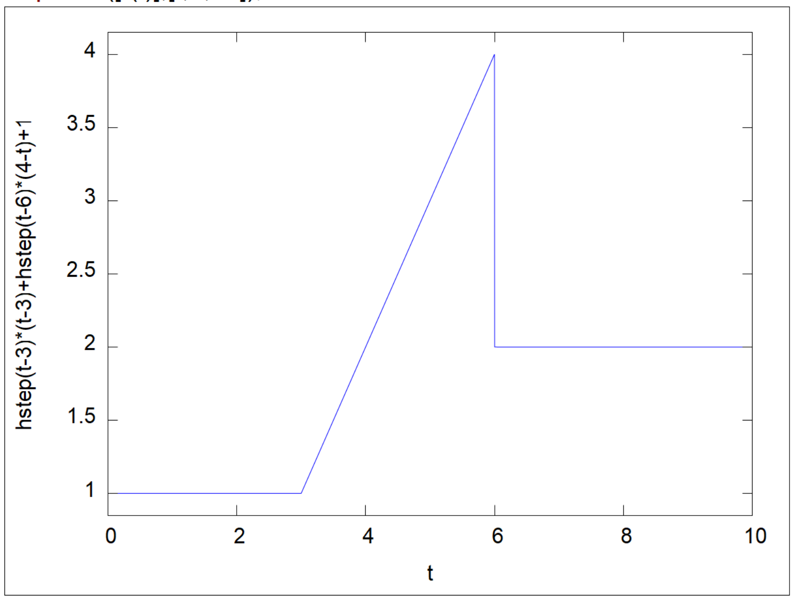
maxima.sourceforge.io/docs/tutorial/en/gaertner-tutorial-revision/Pages/ODE0002.htm

```
→ f (t) := hstep(t−3)·(t−3)+hstep(t−6)·(4−t)+1;
```

(%o2) $f(t):=\mathrm{hstep}(t-3)(t-3)+\mathrm{hstep}(t-6)(4-t)+1$

```
→ wxplot2d([f(t)],[t,0,10]);
```

(%t3)



(%o3)

```
→ ode : 'diff(y(t), t, 2) + 3·'diff(y(t), t) + 2·y(t) = f(t);
```

(%o5) $\frac{d^2}{d t^2}\,y(t)+3\left(\frac{d}{d t}\,y(t)\right)+2\,y(t)=\mathrm{hstep}(t-3)(t-3)+\mathrm{hstep}(t-6)(4-t)+1$

```
→ atvalue(y(t), t=0, 2);
```

(%o6) 2

```
→ atvalue('diff(y(t), t), t= 0, 3);
```

(%o9) 3

```
→ lap_ode:laplace(ode,t,s);
```

(%o10) $3\left(s\,\mathrm{laplace}(y(t),t,s)-2\right)+s^2\,\mathrm{laplace}(y(t),t,s)+2\,\mathrm{laplace}(y(t),t,s)-2\,s-3=\frac{\%e^{-3\,s}}{s^2}+\left(-\frac{2}{s}-\frac{1}{s^2}\right)\%e^{-6\,s}+\frac{1}{s}$

The following cell shows the Maxima solution to the ODE. The only place Maxima fails is in grappling with the 'map' command line and thus failing to elucidate the Regularland (time series) solution. Maxima inserts a little yellow blob to admit its failure. But the omission can be made up by Wolfram Alpha, using the transcribed 'map' line's determined value as input:

!| inverse laplace transform [e^(-6 s) ((2 s^3+9 s^2+s) e^(6 s)+e^(3 s)-2s-1)/(s^4+3 s^3+2 s^2),s,t] |!

Wolfram's output is $g(t)$. The resulting plot matches the original site's plot perfectly.

(Maxima trivia note. A good deal of web and doc searching will not reveal the way to voluntarily insert a line break into an input cell. A little experimentation shows that it's merely necessary to enter a keyboard return to effect the desired line-split, which affects visual appearance but not functionality.)

```
→ sol: solve(%, 'laplace(y(t), t, s));
```

(%o11) $\left[\mathrm{laplace}(y(t),t,s)=\frac{\%e^{-6\,s}\left(\left(2\,s^3+9\,s^2+s\right)\%e^{6\,s}+\%e^{3\,s}-2\,s-1\right)}{s^4+3\,s^3+2\,s^2}\right]$

```
→ map( lambda( [eq], ilt(eq, s, t)), sol);
```

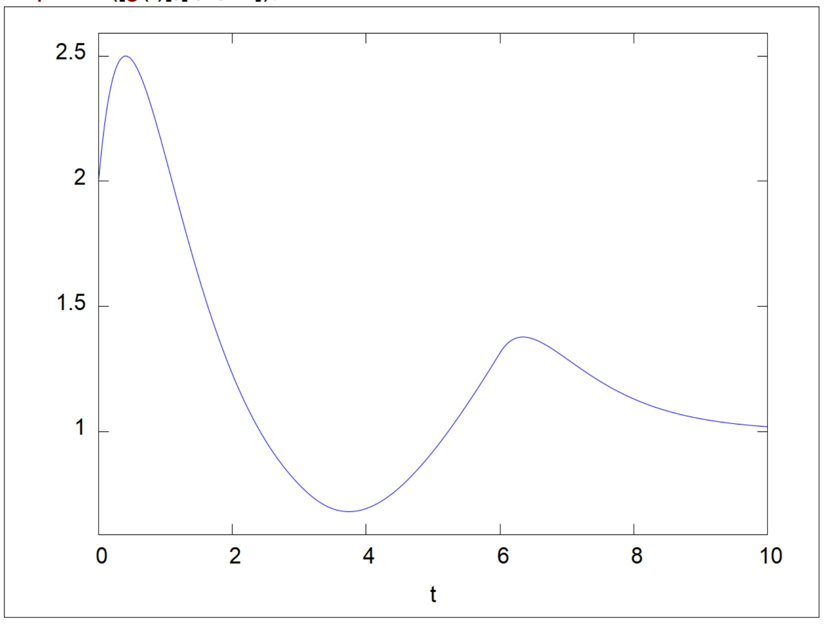
(%o12) $\left[y(t)=\mathrm{ilt}\left(\frac{\%e^{-6\,s}\left(\left(2\,s^3+9\,s^2+s\right)\%e^{6\,s}+\%e^{3\,s}-2\,s-1\right)}{s^4+3\,s^3+2\,s^2},s,t\right)\right]$

```
(%i6) g(t) := (1/4)·%e^(−2·t)·((%e^(2·t)·(11−2·t)+4·%e^(t+6)−3·%e^(12))·hstep(t−6) +
    (%e^(2·t)·(2·t−9)+4·%e^(t+3)−%e^6)·hstep(t−3)+2·(12·%e^t+%e^(2·t)−9));
```

(%o6) $g(t):=\frac{1}{4}\,\%e^{(-2)\,t}\left(\left(\%e^{2\,t}(11-2\,t)+4\,\%e^{t+6}+(-3)\,\%e^{12}\right)\mathrm{hstep}(t-6)+\left(\%e^{2\,t}(2\,t-9)+4\,\%e^{t+3}-\%e^{6}\right)\mathrm{hstep}(t-3)+2\left(12\,\%e^{t}+\%e^{2\,t}-9\right)\right)$

```
(%i7) wxplot2d([g(t)],[t,0,10]);
```

(%t7)



(%o7)

24.8 Solve $y'' - 2y' + y = f(x); \quad y(0) = 0,\ y'(0) = 0.$

This problem cannot be entered into Wolfram Alpha. W|A rejects it in both interpretations: as a differential equation for solving, or as an input to a Laplace transform conversion.

The problem seems a little odd as posed, but the intent is apparently to show how to deal with an unspecified rhs.

Proceeding in this direction, the following Maxima worksheet is produced.

(%i7) ode : 'diff(y(t), t, 2) − 2·'diff(y(t), t) + y(t) = f(t);

(%o7) $\frac{d^2}{dt^2}\mathrm{y}(t)-2\left(\frac{d}{dt}\mathrm{y}(t)\right)+\mathrm{y}(t)=\mathrm{f}(t)$

(%i8) atvalue(y(t), t=0, 0);

(%o8) 0

(%i9) atvalue('diff(y(t), t), t=0, 0);

(%o9) 0

(%i10) lap_ode: laplace(ode, t, s);

(%o10) $s^2\,\mathrm{laplace}(\mathrm{y}(t),t,s)-2s\,\mathrm{laplace}(\mathrm{y}(t),t,s)+\mathrm{laplace}(\mathrm{y}(t),t,s)=\mathrm{laplace}(\mathrm{f}(t),t,s)$

(%i11) sol : solve(%, 'laplace(y(t), t, s));

(%o11) $\left[\mathrm{laplace}(\mathrm{y}(t),t,s)=\frac{\mathrm{laplace}(\mathrm{f}(t),t,s)}{s^2-2s+1}\right]$

(%i12) map(lambda([eq], ilt(eq, s, t)), sol);

(%o12) $\left[\mathrm{y}(t)=\mathrm{ilt}\left(\frac{\mathrm{laplace}(\mathrm{f}(t),t,s)}{s^2-2s+1},s,t\right)\right]$

Above: Maxima gets confused about how to handle the inverse transform of a dummy function. Looking at the 'map' line, the inverse transform of the transform of a dummy function logically needs to recover the function. So the numerator of the floundering fraction can be nabbed for later re-insertion.

→ sol2 : ilt(1/(s^2 − 2·s + 1), s, t);

(%o13) $t\%e^t$

→ sol3 : %·f(t);

(%o14) $t\%e^t\,\mathrm{f}(t)$

Above: The 'map' line as written previously proceeds without its numerator, then the numerator is applied artificially, as described. The form matches the text answer.

24.9 Solve $y'' + y = f(x); \quad y(0) = 0,\ y'(0) = 0$ if $f(x) = \begin{cases} 0 & x < 1 \\ 2 & x \geq 1 \end{cases}.$

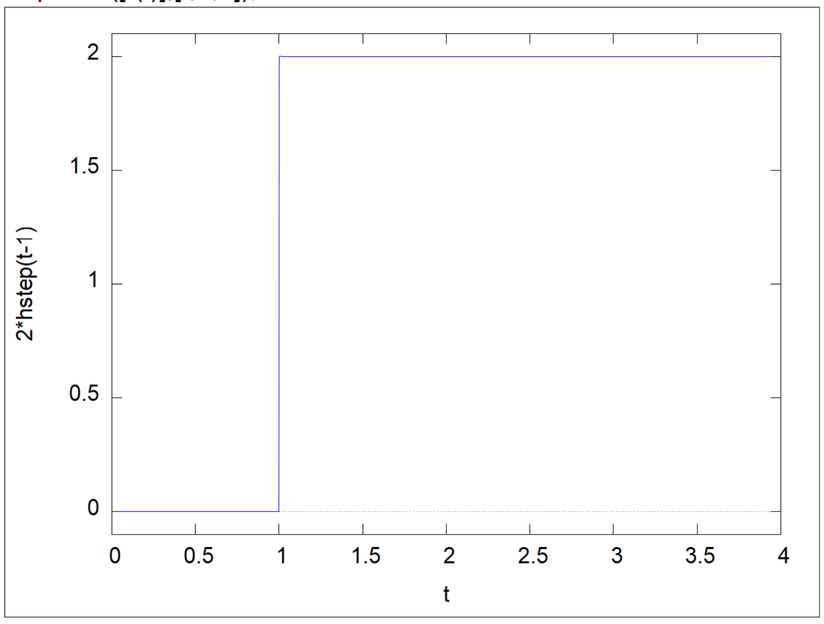
This problem is similar to 24.7a above. The Heaviside configuration is simpler on this one than the former.

In [ ]:

```
```

```
(%i37) f(t) := 2·hstep(t − 1) ;
```
(%o37) $f(t):=2\,\mathrm{hstep}(t-1)$

```
(%i38) wxplot2d([f(t)],[t,0,4]);
```
(%t38)



(%o38)

The problem is pretty simple except for the hstep function. Apparently this increases the difficulty to the point that Maxima is unable to calculate the inverse transform of the solution. So as in Problem 24.7a, Wolfram Alpha must be called on, with the input line:

!| inverse laplace transform[2 e^(-s)/(s^3+s),s,t] |!

```
ode : 'diff(y(t), t, 2) + y(t) = f(t);
```
(%o39) $\frac{d^2}{d t^2}\,y(t)+y(t)=2\,\mathrm{hstep}(t-1)$

```
atvalue('diff(y(t), t), t=0, 0);
```
(%o40) 0

```
atvalue(y(t), t=0, 0);
```
(%o41) 0

```
lap_ode: laplace(ode, t, s);
```
(%o42) $s^2\,\mathrm{laplace}(y(t),t,s)+\mathrm{laplace}(y(t),t,s)=\frac{2\,\%e^{-s}}{s}$

```
sol : solve(%, 'laplace(y(t), t, s));
```
(%o43) $\left[\mathrm{laplace}(y(t),t,s)=\frac{2\,\%e^{-s}}{s^3+s}\right]$

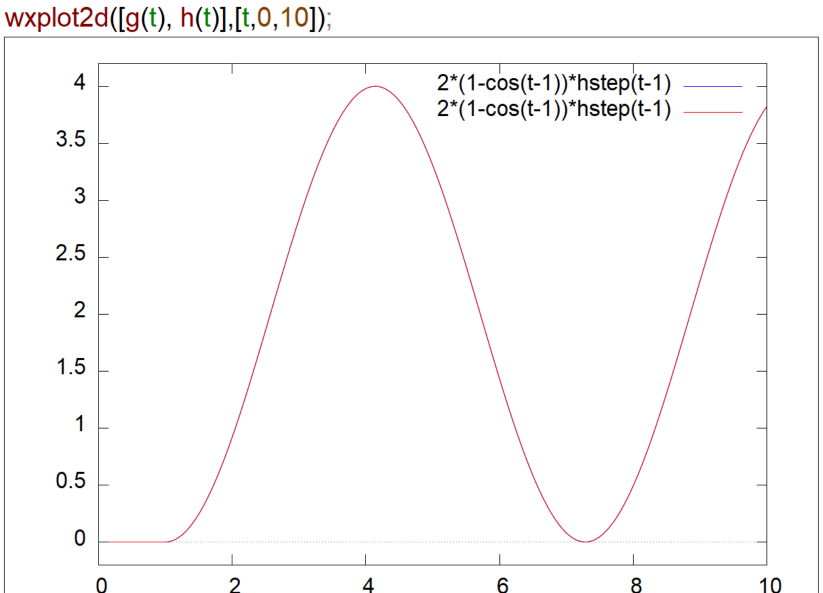
```
map(lambda([eq], ilt(eq, s, t)), sol);
```
(%o44) $\left[y(t)=\mathrm{ilt}\left(\frac{2\,\%e^{-s}}{s^3+s},s,t\right)\right]$

The output from Wolfram Alpha is the function $g(t)$. However, the text expresses the answer as in the function $h(t)$. These two functions are plotted on top of each other by Gnuplot, and so it is assumed that they are equivalent.

```
g(t) := 2·hstep(t − 1)·(1 − cos(1 − t));
```
(%o51) $g(t):=2\,\mathrm{hstep}(t-1)\,(1-\cos(1-t))$

```
h(t) := 2·hstep(t − 1)·(1 − cos(t − 1));
```
(%o52) $h(t):=2\,\mathrm{hstep}(t-1)\,(1-\cos(t-1))$

```
wxplot2d([g(t), h(t)],[t,0,10]);
```
(%t53)



(%o53)

24.10 Solve $y''' + y' = e^x; \quad y(0) = y'(0) = y''(0) = 0.$

Now the Heaviside regimen appears to be over. However, Maxima will still take the starting role.

(%i1) ode: 'diff(y(t), t, 3) + 'diff(y(t), t) = %e^t;

(%o1) $\frac{d^3}{dt^3} y(t) + \frac{d}{dt} y(t) = \%e^t$

(%i2) atvalue(y(t), t=0, 0);

(%o2) 0

(%i3) atvalue('diff(y(t), t), t=0, 0);

(%o3) 0

(%i4) atvalue('diff(y(t), t, 2), t=0, 0);

(%o4) 0

(%i5) lap_ode: laplace(ode, t, s);

(%o5) $s^3 \mathrm{laplace}(y(t), t, s) + s\,\mathrm{laplace}(y(t), t, s) = \frac{1}{s-1}$

(%i6) sol: solve(%, 'lapace(y(t), t, s));

(%o6) []

From the empty bracket which is presented above, it appears that third order is beyond Maxima's capabilities. With the transform of the target function available, Wolfram Alpha can take it from here.

!| solve[s^3 y + s y = 1/(s -1)], y |!

gives the solution to $Y(s)$ in s-space, which is

$$y = \frac{1}{s\,(s^3 - s^2 + s - 1)}$$

Then

!| inverselaplacetransform[1/(s (s^3 - s^2 + s - 1)) |!

takes the solution back to t-space, resulting in

$$\frac{1}{2}(e^t - \sin(t) + \cos(t) - 2)$$

or, un-substituting,

$$\frac{1}{2}(e^x - \sin(x) + \cos(x) - 2)$$

24.11 Solve $y' - 5y = 0$

This problem can be entered into Wolfram Alpha:

!| y' - 5 y = 0 |!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$y(x) = c_1\, e^{5x}$

Without an initial condition, an arbitrary constant attaches.

24.12 Solve $y'' - 3y' + 2y = e^{-x}$

This problem can be entered into Wolfram Alpha:

!| y'' - 3 y' + 2 y = e^(-x) |!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$y(x) = c_1 e^x + c_2 e^{2x} + \frac{e^{-x}}{6}$

Without an initial conditions, arbitrary constants for each level of order attach.

24.13 Solve $y'' - 3y' + 2y = e^{-x}; \quad y(1) = 0,\ y'(1) = 0.$

This problem can be entered into Wolfram Alpha:

!| {y'' - 3 y' + 2 y = e^(-x), {y(1) = 0, y'(1) = 0 }} |!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$y(x) = \frac{1}{6} e^{-x-3} (e - e^x)^2 (2e^x + e)$

24.14 Solve $\frac{dN}{dt} = 0.05N; \quad N(0) = 20000\,.$

Wolfram Alpha seems inclined to refuse this problem, but with the right technique it can be run. The form of the entry, even the specific symbols, matter.

!| {y' = 0.05 y}, { y(0) = 20000} |!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$y(x) = 20000\, e^{0.05x}$

that is,

$N(t) = 20000\, e^{0.05t}$

24.15 Solve $\frac{dI}{dt} + 50I = 5; \quad I(0) = 0\,.$

Wolfram Alpha might be inclined to refuse this problem, but with the technique shown in Prob 24.14 it can be run. The form of the entry, even the specific symbols, matter.

!| {y' + 50 y = 5}, {y(0) = 0} |!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$y(x) = \frac{1}{10} (1 - e^{-50\,x})$

that is,

$I(t) = \frac{1}{10} (1 - e^{-50\,t})$

24.16 Solve $\ddot{x} + 16x = 2 \sin 4t; \quad x(0) = -\frac{1}{2},\ \dot{x}(0) = 0\,.$

This problem is set out in a straightforward way and needs no tricks.

!| {x'' + 16 x = 2 sin(4 t)}, { x(0) = -1/2, x'(0) = 0} |!

In Wolfram Alpha the expression is referred to as the "Differential equation solution":

$$x(t) = \frac{1}{16}(\sin(4t) - 4(t + 2)\cos(4t))$$

In [ ]:
```
1
```