Differential Equations&*Mathematica*

Authors: Bruce Carpenter, Bill Davis and Jerry Uhl ©2001

Producer: Bruce Carpenter

Version 4.0 Publisher: Math Everywhere, Inc.

DE.02 Transition from Calculus to DiffEq: The Forced Oscillator DiffEq y″[t] + b y′[t] + c y[t] = f[t]

GIVE IT A TRY!

Mathematica Initializations

RC: 09/08/09: Excellent - just a little clean up needed. When you get imaginaries spit out from the characteristic equation method, you need to expand out those expressions, and get the imaginaries to cancel out, leaving a purely real function.

DZ: 09/09/09: The imaginaries have been simplified from the exponential.

RC: 1/25/10: Very good. The Chop[ ] function is nice to help out with clearing imaginaries out, too.

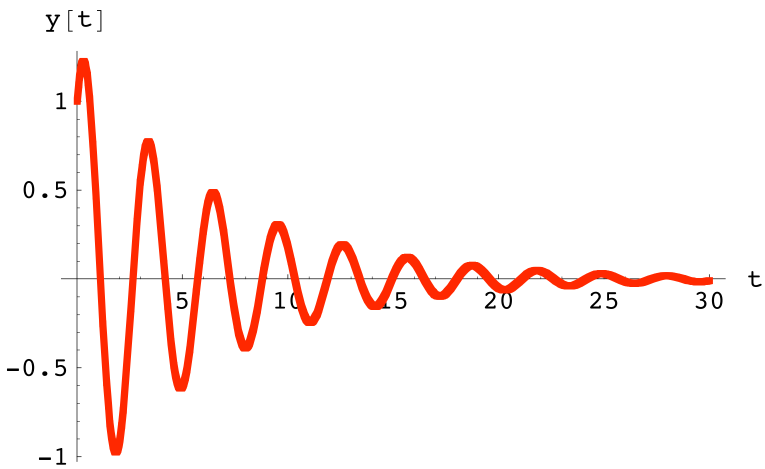
G.1) Using the convolution integral to go after formulas for some simple unforced and forced oscillators*

□ G.1.a) Damped, unforced

Here's a plot of the solution unforced damped oscillator
y″[t] + 0.3 y′[t] + 4.2 y[t] = 0 with y[0] = 1 and y′[0] = 1.5:

```
b = 0.3;
c = 4.2;
Clear[y, ndsy, t, f];
```

```
diffeq = y''[t] + b y'[t] + c y[t] == 0;
endtime = 30;
ndsol = NDSolve[{diffeq, y[0] == 1, y'[0] == 1.5}, y[t], {t, 0, endtime}];
ndsy[t_] = y[t] /. ndsol[[1]];
ndsplot = Plot[ndsy[t], {t, 0, endtime}, PlotStyle → {{Thickness[0.01], Red}},
   PlotRange → All, AxesLabel → {"t", "y[t]"}, AspectRatio → 1/GoldenRatio];
diffeq
```




```
4.2 y[t] + 0.3 y'[t] + y''[t] == 0
```

Use the characteristic equation to come up with a formula for this unforced damped oscillator.
DZ 2009-08-24

□ Step1: Write down the characteristic equation:

```
Clear[z];
charequation = z^2 + b z + c == 0
```

$4.2 + 0.3 z + z^2 == 0$

```
ColumnForm[Thread[linoscdiffeq]]
```

```
Thread::normal :
 Nonatomic expression expected at position 1 in Thread[linoscdiffeq]. More…

linoscdiffeq
```

□ Step 2: Solve the characteristic equation for z to set up the general solution of the diffeq

```
zsols = Solve[charequation, z]
```

```
{{z → -0.15 - 2.04389 i}, {z → -0.15 + 2.04389 i}}
```

Fish the solutions for z out:

```
z1 = zsols[[1, 1, 2]]
```

```
-0.15 - 2.04389 i
```

```
z2 = zsols[[2, 1, 2]]
```

```
-0.15 + 2.04389 i
```

Set up the general form of the solutions.

```
Clear[gensol, K1, K2];
gensol[t_] = K1 E^(z1 t) + K2 E^(z2 t)
```

$e^{(-0.15-2.04389\, i)\, t}\, K1 + e^{(-0.15+2.04389\, i)\, t}\, K2$

☐ Step 3: Solve for the K1 and K2 that correspond to the given starting data on y[0] and y′[0].

```
ystarteq = gensol[0] == startery;
yprimestarteq = gensol'[0] == starteryprime;
Ksols = Solve[{ystarteq, yprimestarteq}, {K1, K2}]
```

```
{{K1 → (0.5 + 0.0366947 i) startery + (0. + 0.244631 i) starteryprime,
  K2 → (0.5 - 0.0366947 i) startery - (0. + 0.244631 i) starteryprime}}
```

Substitute these values of K1 and K2 in to get the raw form of the exact formula:

```
gensol[t] /. Ksols[[1]]
```

$e^{(-0.15+2.04389\, i)\, t}\, ((0.5 - 0.0366947\, i)\, startery - (0. + 0.244631\, i)\, starteryprime) + e^{(-0.15-2.04389\, i)\, t}\, ((0.5 + 0.0366947\, i)\, startery + (0. + 0.244631\, i)\, starteryprime)$

Make it look nice by hitting gensol[t] with the fundamental identity

$$E^{(a + I b) t} = E^{a t} Cos[b t] + I E^{a t} Sin[b t]:$$

```
Clear[yformula];
yformula[t_] = Chop[ComplexExpand[gensol[t] /. Ksols[[1]]]]
```

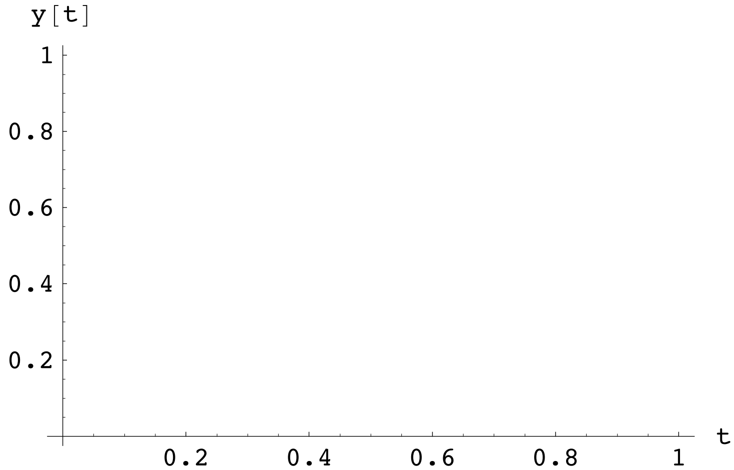
$1.\, e^{-0.15\, t}\, startery\, Cos[2.04389\, t] + 0.0733893\, e^{-0.15\, t}\, startery\, Sin[2.04389\, t] + 0.489262\, e^{-0.15\, t}\, starteryprime\, Sin[2.04389\, t]$

```
ColumnForm[Thread[linoscdiffeq]]

Plot[yformula[t], {t, 0, 30}, PlotStyle → {{Thickness[0.01], Red}},
  AspectRatio → 1/GoldenRatio, PlotRange → All, AxesLabel → {"t", "y[t]"}];
```

```
Thread::normal :
 Nonatomic expression expected at position 1 in Thread[linoscdiffeq]. More…
```

```
linoscdiffeq
```

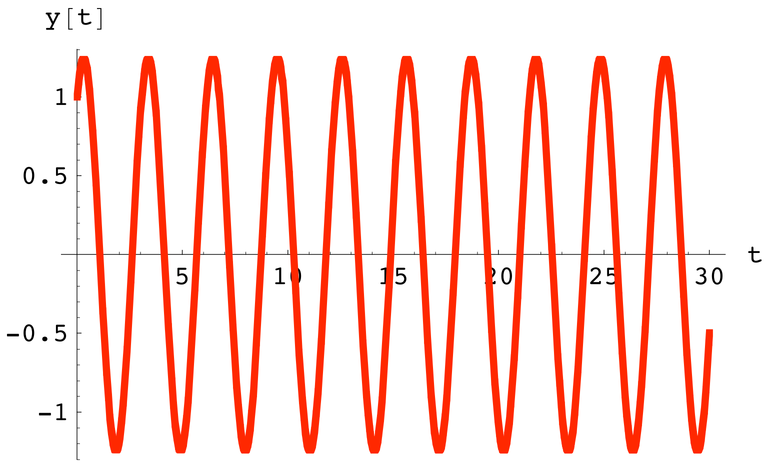


RC: 09/08/09: Excellent. A better answer would be to show these two graphs (yformula and the one from NDSolve together on the same graph, to verify they are the same)

G.1.b) Undamped,unforced

Here's a plot of the solution unforced undamped oscillator
y′′[t] + 4.2 y[t] = 0 with y[0] = 1 and y′[0] = 1.5:

```
b = 0;
c = 4.2;
Clear[y, ndsy, t, f];
diffeq = y''[t] + b y'[t] + c y[t] == 0;
endtime = 30;
ndsol = NDSolve[{diffeq, y[0] == 1, y'[0] == 1.5}, y[t], {t, 0, endtime}];
ndsy[t_] = y[t] /. ndsol[[1]];
ndsplot = Plot[ndsy[t], {t, 0, endtime}, PlotStyle → {{Thickness[0.01], Red}},
   PlotRange → All, AxesLabel → {"t", "y[t]"}, AspectRatio → 1/GoldenRatio];
diffeq
```




```
4.2 y[t] + y''[t] == 0
```

Use the characteristic equation to come up with a formula for this unforced undamped oscillator.

DZ 2009-08-24:

```
Clear[t, y];
linoscdiffeq = {y''[t] + b y'[t] + c y[t] == 0,
      y[0] == startery, y'[0] == starteryprime};
ColumnForm[Thread[linoscdiffeq]]
```

```
4.2 y[t] + y''[t] == 0
y[0] == startery
y'[0] == starteryprime
```

The characteristic equation for this oscillator diffeq is:

```
Clear[z];
z^2 + b z + c == 0
```

$4.2 + z^2 == 0$

Step1: Write down the characteristic equation:

```
Clear[z];
charequation = z^2 + b z + c == 0
```

$4.2 + z^2 == 0$

Note the relation between the characteristic equation and the oscillator diffeq:

```
ColumnForm[Thread[linoscdiffeq]]
```

```
4.2 y[t] + y''[t] == 0
y[0] == startery
y'[0] == starteryprime
```

□ Step 2: Solve the characteristic equation for z to set up the general solution of the diffeq

```
zsols = Solve[charequation, z]
```

```
{{z → 0. − 2.04939 i}, {z → 0. + 2.04939 i}}
```

Fish the solutions for z out:

```
z1 = zsols[[1, 1, 2]]
```

```
0. − 2.04939 i
```

```
z2 = zsols[[2, 1, 2]]
```

```
0. + 2.04939 i
```

Set up the general form of the solutions.

```
Clear[gensol, K1, K2];
gensol[t_] = K1 E^(z1 t) + K2 E^(z2 t)
```

$$e^{(0.-2.04939\, i)\, t}\, K1 + e^{(0.+2.04939\, i)\, t}\, K2$$

Here K1 and K2 are cleared constants.

□ Step 3: Solve for the K1 and K2 that correspond to the given starting data on y[0] and y′[0].

```
ystarteq = gensol[0] == startery;
yprimestarteq = gensol'[0] == starteryprime;
Ksols = Solve[{ystarteq, yprimestarteq}, {K1, K2}]
```

```
{{K1 → (0.5 + 0. i) startery + (0. + 0.243975 i) starteryprime,
  K2 → (0.5 + 0. i) startery − (0. + 0.243975 i) starteryprime}}
```

Substitute these values of K1 and K2 in to get the raw form of the exact formula:

```
gensol[t] /. Ksols[[1]]
```

$$e^{(0.+2.04939\, i)\, t}\, ((0.5 + 0.\, i)\, startery - (0. + 0.243975\, i)\, starteryprime) + e^{(0.-2.04939\, i)\, t}\, ((0.5 + 0.\, i)\, startery + (0. + 0.243975\, i)\, starteryprime)$$

Make it look nice by hitting gensol[t] with the fundamental identity

$$E^{(a + I b) t} = E^{a t} \text{Cos}[b t] + I E^{a t} \text{Sin}[b t]:$$

```
Clear[yformula];
yformula[t_] = Chop[ComplexExpand[gensol[t] /. Ksols[[1]]]]
```

```
1. startery Cos[2.04939 t] + 0.48795 starteryprime Sin[2.04939 t]
```

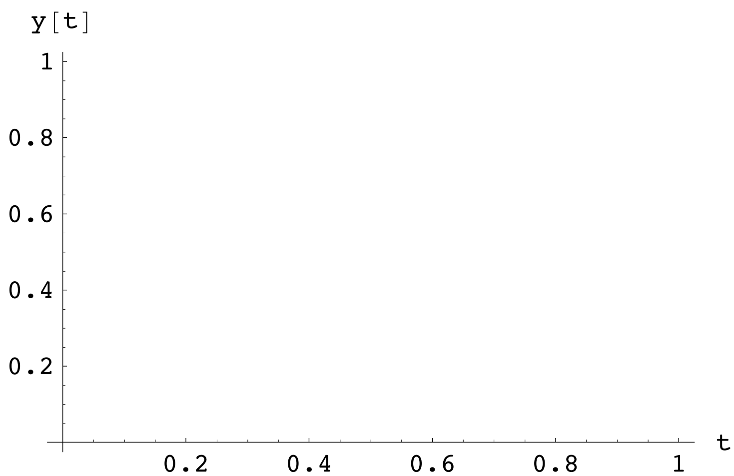
RC: 09/08/09: Excellent

See the solution :

```
ColumnForm[Thread[linoscdiffeq]]

Plot[yformula[t], {t, 0, 30}, PlotStyle → {{Thickness[0.01], Red}},
  AspectRatio → 1/GoldenRatio, PlotRange → All, AxesLabel → {"t", "y[t]"}];
```

```
4.2 y[t] + y''[t] == 0
y[0] == startery
y'[0] == starteryprime
```

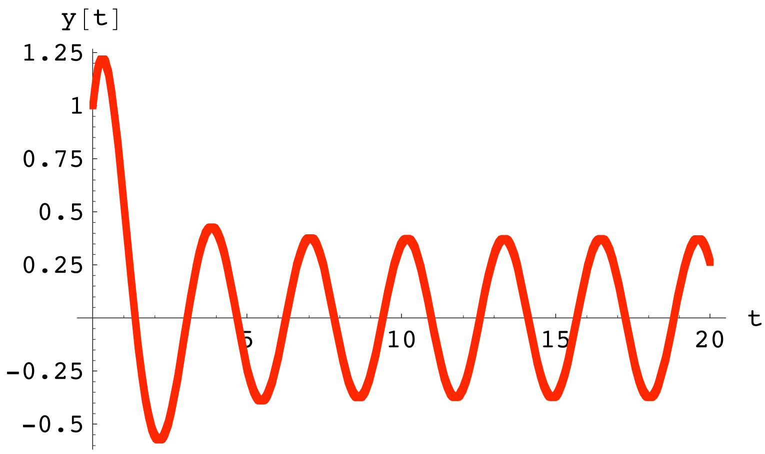


□ G.1.c) Forced, damped

Here's a forced damped oscillator plotted via `NDSolve`:

```
b = 1.6;
c = 4.2;
Clear[y, ndsy, t, f];
f[t_] = 1.2 Cos[2 t];
diffeq = y''[t] + b y'[t] + c y[t] == f[t];
endtime = 20;
ndsol = NDSolve[{diffeq, y[0] == 1, y'[0] == 1.5}, y[t], {t, 0, endtime}];
ndsy[t_] = y[t] /. ndsol[[1]];
```

```
ndsplot = Plot[ndsy[t], {t, 0, endtime}, PlotStyle → {{Thickness[0.01], Red}},
   PlotRange → All, AxesLabel → {"t", "y[t]"}, AspectRatio → 1/GoldenRatio];
diffeq
```



4.2 y[t] + 1.6 y′[t] + y″[t] == 1.2 Cos[2 t]

Use the convolution integral method to come up with a formula for this forced oscillator.
Use your formula to say what the global behavior (steady state) of this oscillator is.
DZ 2009-08-24:

Step 1: Calculate **yunforced[t]**

This involves coming up with the formula for the unforced solution.
This is the solution of the unforced damped oscillator

$$y''[t] + 1.6\, y'[t] + 4.2\, y[t] = 0$$

with $y[0] = 1.0$ and $y'[0] = 1.5$.
You already know how to do this:

```
Clear[f, t];
f[t_] = 1.2 Cos[2 t];
(b = 1.6;)
c = 4.2;
ystarter = 1.0;
yprimestarter = 1.5;
Clear[C1, C2, z, z1, z2, generalsol, t];
charequation = z^2 + b z + c == 0;
zsols = Solve[charequation, z];
generalsol[t_] = C1 E^(z1 t) + C2 E^(z2 t) /. {z1 → zsols[[1, 1, 2]], z2 → zsols[[2, 1, 2]]}
```

$C1\, e^{(-0.8-1.8868 i)\, t} + C2\, e^{(-0.8+1.8868 i)\, t}$

```
Csols = Solve[{generalsol[0] == ystarter, generalsol′[0] == yprimestarter}];
Clear[yunforced];
yunforced[t_] = Chop[ComplexExpand[generalsol[t] /. Csols[[1]]]]
```

$1.\, e^{-0.8 t} \text{Cos}[1.8868\, t] + 1.219\, e^{-0.8 t} \text{Sin}[1.8868\, t]$

□ Step 2: Calculate the unit impulse response, **yunitimpulse[t]**

This involves coming up with the formula of the solution of the **unforced** oscillator

$$y''[t] + 1.6\, y'[t] + 4.2\, y[t] = 0$$

with $y[0] = 0$ and $y'[0] = 1$.

You already know how to do this:

```
Csols = Solve[{generalsol[0] == 0, generalsol′[0] == 1}];
Clear[yunitimpulse];
yunitimpulse[t_] = Chop[ComplexExpand[generalsol[t] /. Csols[[1]]]]
```

$0.529999\, e^{-0.8 t} \text{Sin}[1.8868\, t]$

□ Step 3: Calculate

$$\textbf{yzeroinput[t]} = \int_0^t \textbf{yunitimpulse[t} - \textbf{x] f[x]}\, dx$$

This involves going after a formula for the zero input solution, yzeroinput[t], of the forced oscillator

$$y''[t] + 1.6\, y'[t] + 4.2\, y[t] = f[t]$$

with zeroed out starter data $y[0] = 0$ and $y'[0] = 0$:

A miracle of calculus (to be explained later) is that you can get a formula for yzeroinput[t] by setting

$$\text{yzeroinput}[t] = \int_0^t \text{yunitimpulse}[t - x]\, f[x]\, dx$$

```
Clear[yzeroinput, x];
yzeroinput[t_] = Apart[Chop[∫_0^t yunitimpulse[t - x] f[x] dx]]
```

$-0.0233463\, e^{-0.8 t} (1.\, \text{Cos}[1.8868\, t] + 17.384\, \text{Sin}[1.8868\, t]) +$
$0.0233463\, (1.\, \text{Cos}[2.\, t] + 16.\, \text{Sin}[2.\, t])$

Step 4: Set yformulay[t] = yzeroinput[t] + yunforced[t]:

Get your shot at an exact formula for the solution of

$$y''[t] + 1.6\,y'[t] + 4.2\,y[t] = f[t]$$

with f[t] = 1.2 Cos[2 t], y[0] = 1 and y′[0] = 1.5
by putting

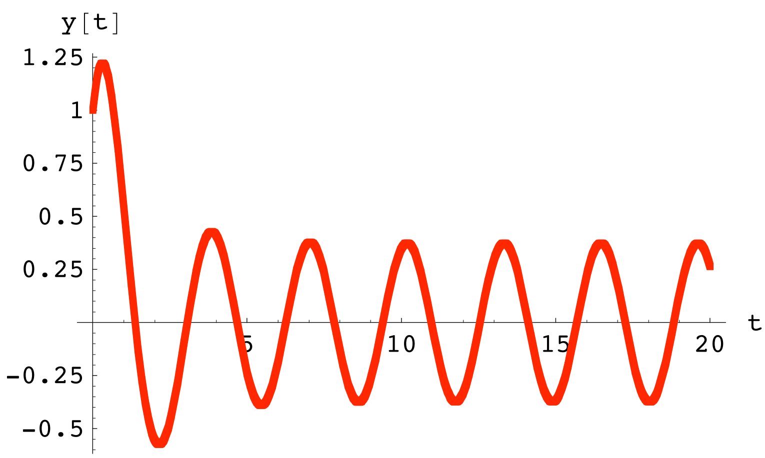
$$formulay[t] = yzeroinput[t] + yunforced[t]:$$

```
Clear[yformula];
yformula[t_] = Expand[yzeroinput[t] + yunforced[t]]
```

```
0.976654 e^(-0.8 t) Cos[1.8868 t] + 0.0233463 Cos[2. t] +
 0.813146 e^(-0.8 t) Sin[1.8868 t] + 0.373541 Sin[2. t]
```

RC: 09/08/09: Excellent

```
Plot[yformula[t], {t, 0, 20}, PlotStyle → {{Thickness[0.01], Red}},
  AspectRatio → 1/GoldenRatio, AxesLabel → {"t", "y[t]"}];
```

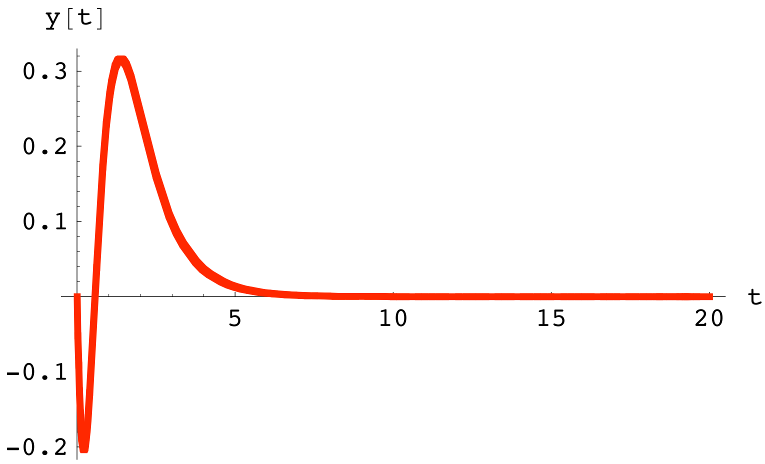


G.1.d) Another forced, damped

Here's another forced damped oscillator plotted courtesy of **NDSolve**:

```
b = 4.7;
c = 6.8;
Clear[y, ndsy, t, f];
f[t_] = 6 E^-t;
diffeq = y''[t] + b y'[t] + c y[t] == f[t];
endtime = 20;
ndsol = NDSolve[{diffeq, y[0] == 0, y'[0] == -2.3}, y[t], {t, 0, endtime}];
ndsy[t_] = y[t] /. ndsol[[1]];
```

```
ndsplot = Plot[ndsy[t], {t, 0, endtime}, PlotStyle → {{Thickness[0.01], Red}},
   PlotRange → All, AxesLabel → {"t", "y[t]"}, AspectRatio → 1/GoldenRatio];
diffeq
```



$6.8\, y[t] + 4.7\, y'[t] + y''[t] == 6\, e^{-t}$

Use the convolution integral method to come up with a formula for this forced oscillator.

Use your formula to confirm that the global behavior (steady state) of this oscillator is 0.

DZ 2009-08-24:

Step 1: Calculate **yunforced[t]**

This involves coming up with the formula for the unforced solution.

This is the solution of the unforced damped oscillator

$$y''[t] + 4.7\, y'[t] + 6.8\, y[t] = 0$$

with $y[0] = 0$ and $y'[0] = -2.3$.

You already know how to do this:

```
Clear[f, t];
f[t_] = 6 E^-t;
(b = 4.7;)
c = 6.8;
ystarter = 0;
yprimestarter = -2.3;
Clear[C1, C2, z, z1, z2, generalsol, t];
charequation = z^2 + b z + c == 0;
zsols = Solve[charequation, z];
generalsol[t_] = C1 E^(z1 t) + C2 E^(z2 t) /. {z1 → zsols[[1, 1, 2]], z2 → zsols[[2, 1, 2]]}
```

$C1\, e^{(-2.35-1.13027\, i)\, t} + C2\, e^{(-2.35+1.13027\, i)\, t}$

```
Csols = Solve[{generalsol[0] == ystarter, generalsol′[0] == yprimestarter}];
Clear[yunforced];
yunforced[t_] = Chop[ComplexExpand[generalsol[t] /. Csols[[1]]]]
```

$-2.03492\, e^{-2.35\, t}\, \text{Sin}[1.13027\, t]$

□ Step 2: Calculate the unit impulse response, yunitimpulse[t]

This involves coming up with the formula of the solution of the **unforced** oscillator

$$y''[t] + 4.7\, y'[t] + 6.8\, y[t] = 0$$

with $y[0] = 0$ and $y'[0] = 1$.

You already know how to do this:

```
Csols = Solve[{generalsol[0] == 0, generalsol′[0] == 1}];
Clear[yunitimpulse];
yunitimpulse[t_] = Chop[ComplexExpand[generalsol[t] /. Csols[[1]]]]
```

$0.884748\, e^{-2.35\, t}\, \text{Sin}[1.13027\, t]$

□ Step 3: Calculate

$$\textbf{yzeroinput[t]} = \int_0^t \textbf{yunitimpulse[t - x] f[x]}\, dx$$

This involves going after a formula for the zero input solution, yzeroinput[t], of the forced oscillator

$$y''[t] + 4.7\, y'[t] + 6.8\, y[t] = f[t]$$

with zeroed out starter data $y[0] = 0$ and $y'[0] = 0$:

A miracle of calculus (to be explained later) is that you can get a formula for yzeroinput[t] by setting

$$\text{yzeroinput}[t] = \int_0^t \text{yunitimpulse}[t - x]\, f[x]\, dx$$

```
Clear[yzeroinput, x];
yzeroinput[t_] = Apart[Chop[∫_0^t yunitimpulse[t - x] f[x] ⅆx]]
```

$1.93548\, e^{-1.\, t} - 1.93548\, e^{-2.35\, t}\, (1.\, \text{Cos}[1.13027\, t] + 1.19441\, \text{Sin}[1.13027\, t])$

Step 4: Set yformulay[t] = yzeroinput[t] + yunforced[t]:

Get your shot at an exact formula for the solution of

$$y''[t] + 4.7\,y'[t] + 6.8\,y[t] = f[t]$$

with $f[t] = 6\,E^{-t}$, $y[0] = 0$ and $y'[0] = -2.3$

by putting

$$\text{formulay}[t] = \text{yzeroinput}[t] + \text{yunforced}[t]:$$

```
Clear[yformula];
yformula[t_] = Expand[yzeroinput[t] + yunforced[t]]
```

$1.93548\,e^{-1.\,t} - 1.93548\,e^{-2.35\,t}\,\text{Cos}[1.13027\,t] - 4.34668\,e^{-2.35\,t}\,\text{Sin}[1.13027\,t]$

RC: 09/08/09: Careful here. We've got imaginaries running around in here, and we need to clean those up first. Try ComplexExpand, Simplify.

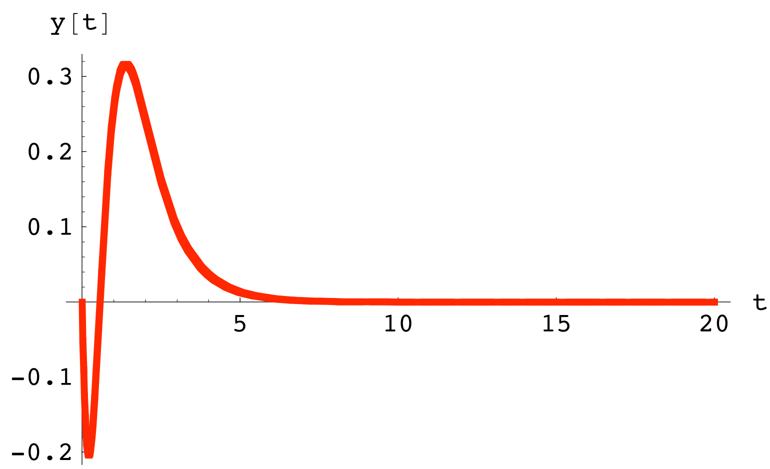
DZ: 09/09/09: The imaginaries have been simplified from the exponential.

```
Clear[yformula];
yformula[t_] = Simplify[ComplexExpand[yzeroinput[t] + yunforced[t]]]
```

$1.93548\,e^{-1.\,t} - 1.93548\,e^{-2.35\,t}\,\text{Cos}[1.13027\,t] - 4.34668\,e^{-2.35\,t}\,\text{Sin}[1.13027\,t]$

RC: 1/25/10: I had to rerun the notebook to get these to imaginaries to go away.

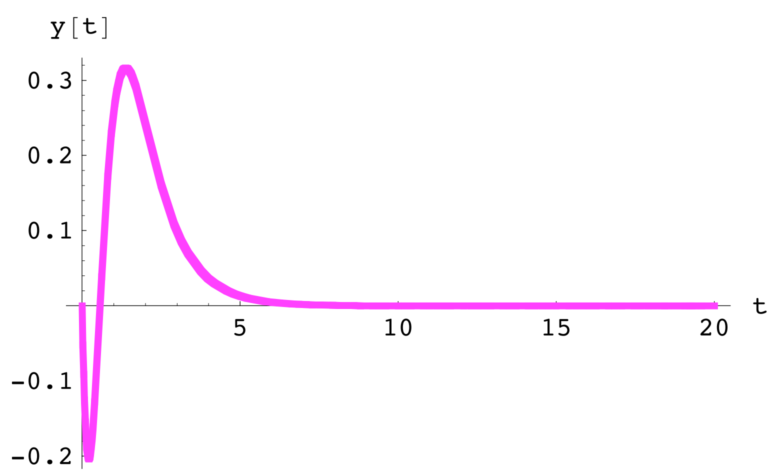
```
Plot[yformula[t], {t, 0, 20}, PlotStyle → {{Thickness[0.01], Red}},
  AspectRatio → 1/GoldenRatio, AxesLabel → {"t", "y[t]"}];
```



□ G.1.e) Impulse-forced

Calculus Cal used `NDSolve` to plot another forced damped oscillator. Here's what he did:

```
b = 4.7;
c = 6.8;
Clear[y, t, f];
f[t_] = 6 E^-t - 7 DiracDelta[t - 5] + 8 DiracDelta[t - 10];
diffeq = y''[t] + b y'[t] + c y[t] == f[t];
endtime = 20;
ndsol = NDSolve[{diffeq, y[0] == 0, y'[0] == -2.3}, y[t], {t, 0, endtime}];
ndsy[t_] = y[t] /. ndsol[[1]];
ndsplot = Plot[ndsy[t], {t, 0, endtime}, PlotStyle → {{Thickness[0.01], Magenta}},
   PlotRange → All, AxesLabel → {"t", "y[t]"}, AspectRatio → 1/GoldenRatio];
diffeq
```



$6.8\, y[t] + 4.7\, y'[t] + y''[t] == 6\, e^{-t} + 8\, \text{DiracDelta}[-10 + t] - 7\, \text{DiracDelta}[-5 + t]$

Inspect the formula for the forcing function:

```
f[t]
```

$6\, e^{-t} + 8\, \text{DiracDelta}[-10 + t] - 7\, \text{DiracDelta}[-5 + t]$

Tell Cal why you know his plot is wrong. And then tell Cal where to go.
DZ 2009-08-24: At t=5 and t=10, the DiracDelta functions should cause the function to change quickly. In Cal's plot, the function is not changed when t=5 or t=10. This should not be the case.
Use the convolution integral method to come up with a formula for this forced oscillator and use it to give an accurate plot.
What happens to y′[t] as t passes through t = 5?
What happens to y′[t] as t passes through t = 10?
DZ 2009-08-24:

RC: 1/25/10: Good.

Step 1: Calculate yunforced[t]

This involves coming up with the formula for the unforced solution.
This is the solution of the unforced damped oscillator

$$y''[t] + 4.7\, y'[t] + 6.8\, y[t] = 0$$

with $y[0] = 0$ and $y'[0] = -2.3$.

```
Clear[f, t];
f[t_] = 6 E^-t - 7 DiracDelta[t - 5] + 8 DiracDelta[t - 10];
b = 4.7;
c = 6.8;
ystarter = 0;
yprimestarter = -2.3;

Clear[C1, C2, z, z1, z2, generalsol, t];
charequation = z^2 + b z + c == 0;
zsols = Solve[charequation, z];
generalsol[t_] = C1 E^(z1 t) + C2 E^(z2 t) /. {z1 → zsols[[1, 1, 2]], z2 → zsols[[2, 1, 2]]}
```

$C1\, e^{(-2.35-1.13027\, i)\, t} + C2\, e^{(-2.35+1.13027\, i)\, t}$

```
Csols = Solve[{generalsol[0] == ystarter, generalsol'[0] == yprimestarter}];
Clear[yunforced];
yunforced[t_] = Chop[ComplexExpand[generalsol[t] /. Csols[[1]]]]
```

$-2.03492\, e^{-2.35\, t}\, \text{Sin}[1.13027\, t]$

□ Step 2: Calculate the unit impulse response, yunitimpulse[t]

This involves coming up with the formula of the solution of the **unforced** oscillator

$y''[t] + 4.7\, y'[t] + 6.8\, y[t] = 0$

with $y[0] = 0$ and $y'[0] = 1$.

You already know how to do this:

```
Csols = Solve[{generalsol[0] == 0, generalsol′[0] == 1}];
Clear[yunitimpulse];
yunitimpulse[t_] = Chop[ComplexExpand[generalsol[t] /. Csols[[1]]]]
```

```
0.884748 e^(-2.35 t) Sin[1.13027 t]
```

□ Step 3: Calculate

$$\textbf{yzeroinput[t]} = \int_0^t \textbf{yunitimpulse[t} - \textbf{x] f[x]}\, dx$$

This involves going after a formula for the zero input solution, yzeroinput[t], of the forced oscillator

$y''[t] + 4.7\, y'[t] + 6.8\, y[t] = f[t]$

with zeroed out starter data $y[0] = 0$ and $y'[0] = 0$:

A miracle of calculus (to be explained later) is that you can get a formula for yzeroinput[t] by setting

$\text{yzeroinput}[t] = \int_0^t \text{yunitimpulse}[t - x]\, f[x]\, dx$

```
Clear[yzeroinput, x];
yzeroinput[t_] = Apart[Chop[∫_0^t yunitimpulse[t - x] f[x] dx]]
```

```
1.93548 e^(-1. t) - 1.93548 e^(-2.35 t)
  (1. Cos[1.13027 t] + 1.19441 Sin[1.13027 t] + 5.87544 × 10^10 Sin[11.3027 - 1.13027 t]
     UnitStep[10 - t UnitStep[-t]] UnitStep[-10 + t UnitStep[t]] -
    405591. Sin[5.65133 - 1.13027 t] UnitStep[5 - t UnitStep[-t]]
     UnitStep[-5 + t UnitStep[t]])
```

□ Step 4: Set yformulay[t] = yzeroinput[t] + yunforced[t]:

Get your shot at an exact formula for the solution of

$y''[t] + 4.7\, y'[t] + 6.8\, y[t] = f[t]$

with $f[t] = 6\, E^{-t} - 7\, \text{DiracDelta}[t - 5] + 8\, \text{DiracDelta}[t - 10]$, $y[0] = 0$ and $y'[0] = -2.3$.

by putting

yformula[t] = yzeroinput[t] + yunforced[t]:

```
Clear[yformula];
yformula[t_] = Expand[yzeroinput[t] + yunforced[t]]
```

$1.93548\, e^{-1.\,t} - 1.93548\, e^{-2.35\,t} \text{Cos}[1.13027\,t] - 4.34668\, e^{-2.35\,t} \text{Sin}[1.13027\,t] - 1.13718\times10^{11}\, e^{-2.35\,t} \text{Sin}[11.3027 - 1.13027\,t]\, \text{UnitStep}[10 - t\,\text{UnitStep}[-t]]\, \text{UnitStep}[-10 + t\,\text{UnitStep}[t]] + 785015.\, e^{-2.35\,t} \text{Sin}[5.65133 - 1.13027\,t]\, \text{UnitStep}[5 - t\,\text{UnitStep}[-t]]\, \text{UnitStep}[-5 + t\,\text{UnitStep}[t]]$

RC: 09/08/09: Careful, we have imaginaries running around again. Must "smooth" them out.

DZ: 09/09/09: The imaginaries have been simplified from the exponential.

```
Clear[yformula];
yformula[t_] = Simplify[ComplexExpand[yzeroinput[t] + yunforced[t]]]
```

$1.93548\, e^{-1.\,t} - 1.93548\, e^{-2.35\,t} \text{Cos}[1.13027\,t] - 4.34668\, e^{-2.35\,t} \text{Sin}[1.13027\,t]$

$1.93548\, e^{-1.\,t} - (463667. + 0.\,i)\, e^{-2.35\,t} \text{Cos}[1.13027\,t] - (633457. + 0.\,i)\, e^{-2.35\,t} \text{Sin}[1.1$

$1.93548\, e^{-1.\,t} + (1.08398\times10^{11} + 0.\,i)\, e^{-2.35\,t} \text{Cos}[1.13027\,t] + (3.43737\times10^{10} + 0.\,i)\, e^{-2.}$

RC: 1/25/10: I had to rerun this once more to get the imaginaries out..

```
yformula[t_] = Chop[ Simplify[ComplexExpand[yzeroinput[t] + yunforced[t]]], 0.00001]
```

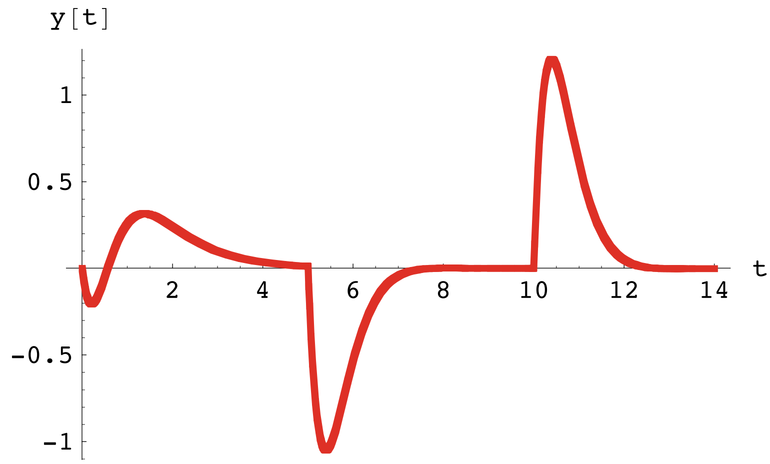
$1.93548\, e^{-1.\,t} - 1.93548\, e^{-2.35\,t} \text{Cos}[1.13027\,t] - 4.34668\, e^{-2.35\,t} \text{Sin}[1.13027\,t]$

$1.93548\, e^{-1.\,t} - 463667.\, e^{-2.35\,t} \text{Cos}[1.13027\,t] - 633457.\, e^{-2.35\,t} \text{Sin}[1.13027\,t]$

$1.93548\, e^{-1.\,t} + 1.08398\times10^{11}\, e^{-2.35\,t} \text{Cos}[1.13027\,t] + 3.43737\times10^{10}\, e^{-2.35\,t} \text{Sin}[1.13027$

RC: 1/25/10: Chop[ ] is nice to help out, too.

```
Plot[yformula[t], {t, 0, 14}, PlotStyle → {{Thickness[0.01], VenetianRed}},
  AspectRatio → 1/GoldenRatio, PlotRange → All, AxesLabel → {"t", "y[t]"}];
```



What happens to y′[t] as t passes through t = 5?
What happens to y′[t] as t passes through t = 10?
DZ 2009-08-24:
When t=5, DiracDelta causes the function to move negative quickly.
When t=10, the function quickly shifts positive.
Cal should have realized that a change in the behavior of the function should be exhibited when the forcing equation triggers.

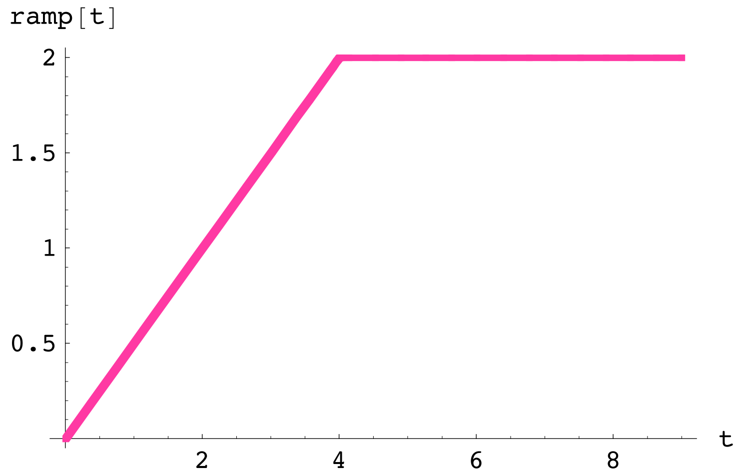
RC: 09/08/09: Excellent

□ G.1.f) Ramp-forced

Here is a ramp forcing function and its plot:

```
Clear[ramp, g, t];
g[t_] = 0.5 t;
changeover = 4.0;
ramp[t_] = g[t] UnitStep[changeover - t] + g[changeover] UnitStep[t - changeover];
Plot[ramp[t], {t, 0, 9}, PlotStyle → {{Thickness[0.01], DeepPink}},
  PlotRange → All, AxesLabel → {"t", "ramp[t]"}, AspectRatio → 1/GoldenRatio];

ramp[t]
```




```
0.5 t UnitStep[4. - t] + 2. UnitStep[-4. + t]
```

Ramp[t] runs
→ with g[t] = 0.5 t for t < 4.0
→ with g[4] = 2 for t > 4.0.
Here is the plot of a damped oscillator forced by ramp[t]:

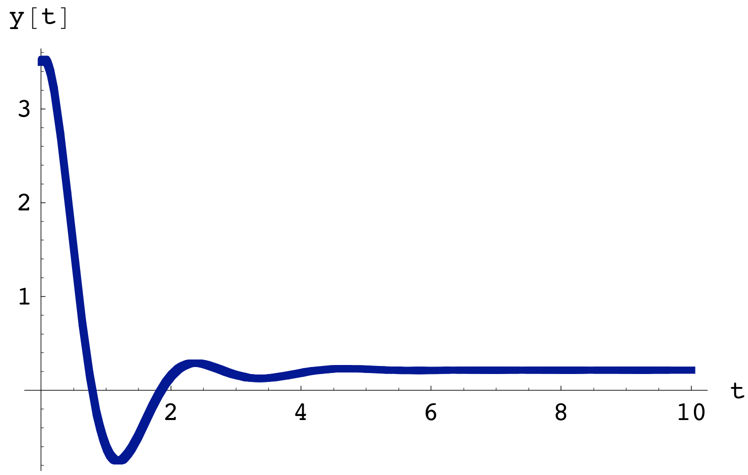
```
Clear[y, t, f];
f[t_] = ramp[t];
b = 2.6;
c = 9.2;
ystarter = 3.5;
yprimestarter = 1.5;

diffeq = (y''[t] + b y'[t] + c y[t] == f[t]);

endtime = 10;

ndsol =
 NDSolve[{diffeq, y[0] == ystarter, y'[0] == yprimestarter}, y[t],
 {t, 0, endtime}];
ndsy[t_] = y[t] /. ndsol[[1]];

ndsplot = Plot[ndsy[t], {t, 0, endtime},
 PlotStyle -> {{Thickness[0.01], NavyBlue}}, PlotRange -> All,
   AspectRatio -> 1 / GoldenRatio, AxesLabel -> {"t", "y[t]"}];
 diffeq
```




```
9.2 y[t] + 2.6 y′[t] + y″[t] == 0.5 t UnitStep[4. - t] + 2. UnitStep[-4. + t]
```

Use the convolution integral method to produce an exact formula for this ramp-forced oscillator.

Zuercher 2009-08-24:

□ Step 1: Calculate yunforced[t]

```
Clear[C1, C2, z, z1, z2, generalsol, t];
charequation = z^2 + b z + c == 0;
zsols = Solve[charequation, z];
generalsol[t_] = C1 E^(z1 t) + C2 E^(z2 t) /. {z1 → zsols[[1, 1, 2]], z2 → zsols[[2, 1, 2]]}
```

$$C1\, e^{(-1.3-2.74044\, i)\, t} + C2\, e^{(-1.3+2.74044\, i)\, t}$$

```
Csols = Solve[{generalsol[0] == ystarter, generalsol′[0] == yprimestarter}];
Clear[yunforced];
yunforced[t_] = Chop[ComplexExpand[generalsol[t] /. Csols[[1]]]]
```

$$3.5\, e^{-1.3\, t} \operatorname{Cos}[2.74044\, t] + 2.20768\, e^{-1.3\, t} \operatorname{Sin}[2.74044\, t]$$

□ Step 2: Calculate the unit impulse response, yunitimpulse[t]

```
Csols = Solve[{generalsol[0] == 0, generalsol′[0] == 1}];
Clear[yunitimpulse];
yunitimpulse[t_] = Chop[ComplexExpand[generalsol[t] /. Csols[[1]]]]
```

```
0.364905 ⅇ^(-1.3 t) Sin[2.74044 t]
```

□ Step 3: Calculate

$$\text{yzeroinput}[t] = \int_0^t \text{unitimpulse}[t - x]\, f[x]\, dx$$

```
Clear[yzeroinput, x];
yzeroinput[t_] = Chop[Apart[Chop[∫_0^t yunitimpulse[t - x] f[x] ⅆx]]]
```

```
0.0543478 (-0.282609 UnitStep[4. - t] + 1. t UnitStep[4. - t] + 4. UnitStep[-4. + t]) +
 0.0153592 ⅇ^(-1.3 t)
  (1. Cos[2.74044 t] UnitStep[4. - t] - 0.816826 Sin[2.74044 t] UnitStep[4. - t] +
    155.113 Cos[2.74044 t] UnitStep[-4. + t] +
    175.345 Sin[2.74044 t] UnitStep[-4. + t])
```

□ Step 4: Set formulay[t] = yzeroinput[t] + yunforced[t]:

```
Clear[formulay];
formulay[t_] = Simplify[yzeroinput[t] + yunforced[t]]
```

```
ⅇ^(-1.3 t)
 (3.5 Cos[2.74044 t] + 2.20768 Sin[2.74044 t] + (ⅇ^(1.3 t) (-0.0153592 + 0.0543478 t) +
      0.0153592 Cos[2.74044 t] - 0.0125458 Sin[2.74044 t]) UnitStep[4. - t] +
   (0.217391 ⅇ^(1.3 t) + 2.38241 Cos[2.74044 t] + 2.69315 Sin[2.74044 t])
    UnitStep[-4. + t])
```

RC: 09/08/09: Excellent

Here is the formula plot together with the NDSolve plot:

```
formulaplot = Plot[formulay[t], {t, 0, endtime},
   PlotStyle → {{Thickness[0.008], Red}}, DisplayFunction → Identity];
Show[ndsplot, formulaplot];
```