ME 163
Using DSolve in *Mathematica* to Solve First Order Equations

■ Introduction

The purpose of this notebook is to show how to use the *Mathematica* function DSolve to find analytical solutions of first order ordinary differential equations. As a prerequisite, it is assumed that you have worked through the section of the tutorial on Replacement Rules. Although there are a variety of ways to accomplish some of the tasks presented here, for simplicity we consider only one straightforward method in each case.

The general form of the DSolve command is

DSolve[equation, dependent variable, independent variable] .

We will first learn how to specify an equation for DSolve. Then we will try it out, thereby learning something about the form of the output provided by DSolve (a replacement rule). After showing how to get a function rather than a replacement rule as an output from DSolve, we carry out a number of examples. We conclude with examples of how to automate the production of tables or a graph sequence from solutions of the equation.

■ Specifying an Equation for DSolve

The syntax required by DSolve for a differential equation is very similar to standard mathematical notation. Derivatives are denoted by a prime, which in fact is typed as the apostrophe character. Multiple derivatives are denoted by repeated apostrophes. Dependent variables always exhibit explicitly the independent variable as a functional argument. Equality in the equation is denoted by the double equals sign. Here is an example, typed first in standard mathematical notation,

$$\frac{dy}{dx} + y = \sin(3x) \;\; ,$$

and now in a form suitable for DSolve,

y'[x] + y[x] == Sin[3x] .

The second argument of DSolve, the dependent variable, is written with the functional argument as y[x], and the independent variable as x. Let's try all of this.

```
DSolve[y'[x] + y[x] == Sin[3 x], y[x], x]
```

$$\left\{\left\{y[x] \to e^{-x}\, C[1] + \frac{1}{10}\,(-3\,\text{Cos}[3\,x] + \text{Sin}[3\,x])\right\}\right\}$$

It appears to work, although we might not have expected the replacement rule form of the answer. Notice that we have an arbitrary constant, which we would need to satisfy an initial condition. We will see shortly how to impose initial conditions and how to check our solutions.

■ Working With the Output of DSolve

Although the replacement rule form of the solution is very flexible, we often want a function for our output -- a function that we can easily evaluate or plot. Let's repeat the above calculation, only now we will give a name to the replacement rule answer. We call it ans.

```
ans = DSolve[y'[x] + y[x] == Sin[3 x], y[x], x]
```

$$\left\{\left\{y[x] \to e^{-x} C[1] + \frac{1}{10} (-3\,\mathrm{Cos}[3\,x] + \mathrm{Sin}[3\,x])\right\}\right\}$$

Note that there are two pairs of curly brackets enclosing the replacement rule (there is a logical reason for this, having to do with the solution of systems of equations). A proper replacement rule has only a single pair of curly brackets, so we eliminate one by the command Flatten

```
ans = Flatten[ans]
```

$$\left\{y[x] \to e^{-x} C[1] + \frac{1}{10} (-3\,\mathrm{Cos}[3\,x] + \mathrm{Sin}[3\,x])\right\}$$

Now we are going to use the replacement rule to construct a function which is the solution. We call the function sol[x].

```
sol[x_] = y[x] /. ans
```

$$e^{-x} C[1] + \frac{1}{10} (-3\,\mathrm{Cos}[3\,x] + \mathrm{Sin}[3\,x])$$

As our first exercise with this solution, we check it by substituting it back into the equation.

```
D[sol[x], x] + sol[x] - Sin[3 x]
```

$$-\mathrm{Sin}[3\,x] + \frac{1}{10} (-3\,\mathrm{Cos}[3\,x] + \mathrm{Sin}[3\,x]) + \frac{1}{10} (3\,\mathrm{Cos}[3\,x] + 9\,\mathrm{Sin}[3\,x])$$

We should have gotten zero. Before we give up, we try simplifying the above expression:

```
Simplify[%]
```

```
0
```

We do indeed get zero, verifying that our solution is correct.

Let's impose an initial condition and thereby determine the constant C[1]. We ask that sol[0]=2.

```
answerC = Solve[sol[0] == 2, C[1]]
```

$\left\{\left\{C[1] \to \frac{23}{10}\right\}\right\}$

We get our final answer by substituting this expression back into the expression for sol[x]. We call this final answer sol2[x].

```
sol2[x_] = sol[x] /. Flatten[answerC]
```

$\frac{23\, e^{-x}}{10} + \frac{1}{10}\,(-3\,\mathrm{Cos}[3\,x] + \mathrm{Sin}[3\,x])$

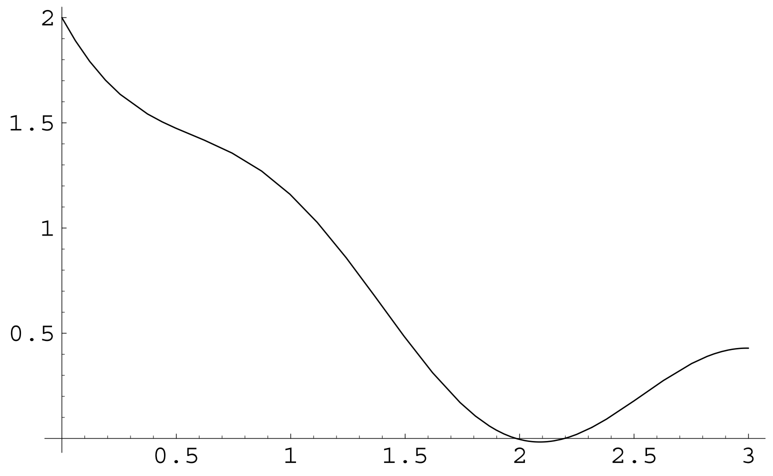
This solution may be evaluated at any point

```
sol2[0.5]
```

```
1.47355
```

or plotted

```
Plot[sol2[x], {x, 0, 3}];
```



■ Incorporating Initial Conditions in the Equation

Even though *Mathematica* did all of the labor in imposing the initial condition, the calculation was a bit involved. There is a much easier way to do this. We can simply include the initial condition in the original specification of the equation. We do this for the problem just solved.

```
DSolve[{y'[x] + y[x] == Sin[3 x], y[0] == 2}, y[x], x]
```

$\left\{\left\{y[x] \to -\frac{1}{10}\, e^{-x}\,(-23 + 3\, e^{x}\,\mathrm{Cos}[3\,x] - e^{x}\,\mathrm{Sin}[3\,x])\right\}\right\}$

And there we have it all in one step. Note that the equation argument is now a list, with the first element of the list being the equation, the second element being the initial condition.

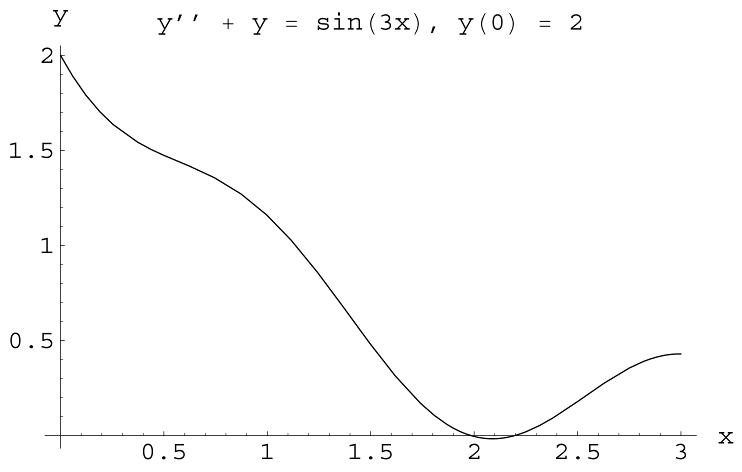
Now we combine all of the steps above into a single command, which produces a solution of the initial value problem named ivsol[x]:

```
ivsol[x_] =
 y[x] /. Flatten[DSolve[{y'[x] + y[x] == Sin[3 x], y[0] == 2}, y[x], x]]
```

$$-\frac{1}{10} e^{-x} (-23 + 3 e^{x} \mathrm{Cos}[3 x] - e^{x} \mathrm{Sin}[3 x])$$

We plot this.

```
Plot[ivsol[x], {x, 0, 3}, AxesLabel → {"x", "y"},
  PlotLabel → "y'' + y = sin(3x), y(0) = 2"];
```



More Examples

A Separable Equation

The example we have used above is a linear equation, and as we have seen DSolve handles it quite well. Let's try a nonlinear but separable equation. We look at homework problem A in assignment #2, which was the motion of a projectile fired upward, including the effects of a quadratic drag. The problem was

$$\frac{dV}{dt} = -g - \frac{k}{m} V^2 \text{ , with } V(0) = V0 \text{ .}$$

We first define the equation and initial condition for use in DSolve.

```
eqn = {V'[t] == -g - (k / m) V[t]^2, V[0] == V0};
```

Now we see if DSolve can handle this with the parameters remaining as symbols.

```
ans = DSolve[eqn, V[t], t]
```

$$\left\{\left\{V[t] \to -\frac{\sqrt{g}\,\sqrt{m}\,\mathrm{Tan}\left[\frac{\sqrt{g}\,\sqrt{k}\,t}{\sqrt{m}} - \mathrm{ArcTan}\left[\frac{\sqrt{k}\,V0}{\sqrt{g}\,\sqrt{m}}\right]\right]}{\sqrt{k}}\right\}\right\}$$

Pretty impressive! Exactly right. Notice that DSolve handled an initial condition which was a literal rather than a numerical constant. Let's use the solution to find the time at which the projectile reaches maximum height -- i.e., the time at which $V = 0$. We first use the replacement rule to construct a function.

```
proj[t_] = V[t] /. Flatten[ans]
```

$$-\frac{\sqrt{g}\,\sqrt{m}\,\mathrm{Tan}\left[\frac{\sqrt{g}\,\sqrt{k}\,t}{\sqrt{m}} - \mathrm{ArcTan}\left[\frac{\sqrt{k}\,V0}{\sqrt{g}\,\sqrt{m}}\right]\right]}{\sqrt{k}}$$

Now we find the time at which V vanishes.

```
Solve[proj[t] == 0, t]
— Solve::ifun :
  Inverse functions are being used by Solve,
    so some solutions may not be found.
```

$$\left\{\left\{t \to \frac{\sqrt{m}\,\mathrm{ArcTan}\left[\frac{\sqrt{k}\,V0}{\sqrt{g}\,\sqrt{m}}\right]}{\sqrt{g}\,\sqrt{k}}\right\}\right\}$$

We get a warning but also the answer we sought. The answer agrees with the one we obtained "by hand" in the homework assignment.

■ A Linear Equation with Variable Coefficients

Now we try a somewhat harder linear equation. The problem is

$$\frac{dy}{dt} - \frac{3}{t}y = t^4 e^t \quad \text{with } y(1) = 1.$$

```
ans = DSolve[{y'[t] - (3 / t) y[t] == t⁴ eᵗ, y[1] == 1}, y[t], t]
```

$$\{\{y[t] \to t^3\,(1 - e^t + e^t\,t)\}\}$$

We convert this to a function form of solution and then check it.

```
lin[t_] = y[t] /. Flatten[ans]
```

$$t^3\,(1 - e^t + e^t\,t)$$

```
D[lin[t], t] - (3 / t) lin[t] - t⁴ eᵗ
```

```
0
```

It checks the equation. Now we check the initial condition.

```
lin[1]
```

```
1
```

■ An Exact Equation

We now try an equation which is neither separable nor linear, but can be solved by hand as an exact equation. The equation is

$$\frac{dy}{dx} = - \frac{2\,xy^2 + 1}{2\,x^2\,y} \,.$$

We may put this into the differential form

$$(2xy^2{+}1\,)\,dx + (2x^2\,y)dy = \;0\;\;,$$

which is easily shown to be exact. The integral is the implicit solution

$$x + x^2\,y^2 \;=\; C\;\;.$$

Let's see what DSolve does with this.

```
DSolve[y'[x] == -(2 x y[x]^2 + 1) / (2 x^2 y[x]), y[x], x]
```

$$\left\{\left\{\mathrm{y[x]} \to -\frac{\sqrt{-\mathrm{x} + \mathrm{C[1]}}}{\mathrm{x}}\right\},\ \left\{\mathrm{y[x]} \to \frac{\sqrt{-\mathrm{x} + \mathrm{C[1]}}}{\mathrm{x}}\right\}\right\}$$

This is fully equivalent with our implicit solution found above.

This is perhaps as good at time as any to warn you that it is essential that in all appearances of the dependent variable, it must be written as y[x] rather than just y. If you forget, you will get wrong answers and no warnings. For example, we redo the above problem, this time carelessly omitting the arguments of the y's in the slope.

```
DSolve[y'[x] == -(2 x y^2  + 1) / (2 x^2 y), y[x], x]
```

$$\left\{\left\{\mathrm{y[x]} \to \frac{1}{2\,\mathrm{x}\,\mathrm{y}} + \mathrm{C[1]} - \mathrm{y}\,\mathrm{Log[x]}\right\}\right\}$$

This is completely wrong. What has happened is that *Mathematica* has interpreted y as a constant completely unrelated to y[x]. It is a treacherous error because there are no error messages -- it is not an error as far as *Mathematica* is concerned.

■ Bernoulli Equation

DSolve even knows how to solve Bernoulli equations. In class we solved the Bernoulli equation

$$\frac{dy}{dx} + 2\,\mathrm{y} \;=\; \frac{x}{y^2}\;,\ \mathrm{y}[1] = 2\;,$$

and found the solution to be

$$y(x) = \{\frac{6x - 1 + 91\,e^{6(1-x)}}{12}\}^{1/3}\ .$$

Now we ask DSolve for the solution.

```
DSolve[{y'[x] + 2 y[x] == x / (y[x])^2, y[1] == 2}, y[x], x]
```

$$\{\{y[x] \to -\frac{(-1)^{2/3}\, e^{-2x}\, (-91\, e^{6} + e^{6x} - 6\, e^{6x}\, x)^{1/3}}{2^{2/3}\, 3^{1/3}}\}\}$$

Although the form is a little less convenient it is essentially the same answer.

■ Making Tables and Plot Sequences from Solutions

We go back to our first example, and show how to make tables of values of the solution and sequences of graphs. The equation solved was

$$\frac{dy}{dx} + y = \sin(3x)\ \ ,\ \ y(0) = y0.$$

We define a function which returns the solution for a specified initial value $y0$. The solution will be returned as an expression in x rather than a function of x.

```
Clear[sol]

sol[y0_] :=
 y[x] /. Flatten[DSolve[{y'[x] + y[x] == Sin[3 x], y[0] == y0}, y[x], x]]
```

To compare with a case done earlier, we choose y0 = 2, and we assign the answer to the variable solA.

```
solA = sol[2]
```

$$-\frac{1}{10}\, e^{-x}\, (-23 + 3\, e^{x}\, \mathrm{Cos}[3\, x] - e^{x}\, \mathrm{Sin}[3\, x])$$

We make a table of values of this solution for x running from 0 to 1 in increments of 0.1.

```
TableForm[Table[{i * 0.1, solA /. x → i * 0.1}, {i, 0, 10}],
 TableHeadings → {None, {"x", "y(x)"}}]
```

```
x       y(x)
0       2
0.1     1.82408
0.2     1.69194
0.3     1.59573
0.4     1.52623
0.5     1.47355
0.6     1.42781
0.7     1.37992
0.8     1.32222
0.9     1.24907
1.      1.15723
```

Let's look at this a little more closely. The heart of the command is Table, which actually produces the list. We execute that separately to see how it works.

```
Table[{i * 0.1, solA /. x → 0.1 * i}, {i, 0, 10}]
```

```
{{0, 2}, {0.1, 1.82408}, {0.2, 1.69194}, {0.3, 1.59573},
 {0.4, 1.52623}, {0.5, 1.47355}, {0.6, 1.42781},
 {0.7, 1.37992}, {0.8, 1.32222}, {0.9, 1.24907}, {1., 1.15723}}
```

There is an implied Do loop using the index i, called an iterator in *Mathematica*. The last argument, {i,0,10}, tells *Mathematica* to perform the operation 11 times, starting at i = 0 and ending at i = 10. The first argument, {i*0.1, solA /. x->0.1*i}, tells *Mathematica* what to compute for each i value. In this case a pair of numbers are computer. The first is i*0.1, which is the x value in the table, and the second is solA /. x -> 0.1*i, which is the solution solA evaluated at that x-value. The output from Table is the linear list that you see above. TableForm rearranges it into a more useful array, and also allows us to add headings.

As our final task, we will construct a sequence of graphs for various initial conditions, and display them on a single graph. We continue with the example we are already using. We already know that sol[y0] will produce the solution for the initial value y0. Now we construct a function which will produce a graph of this solution. We call the function graph[y0].

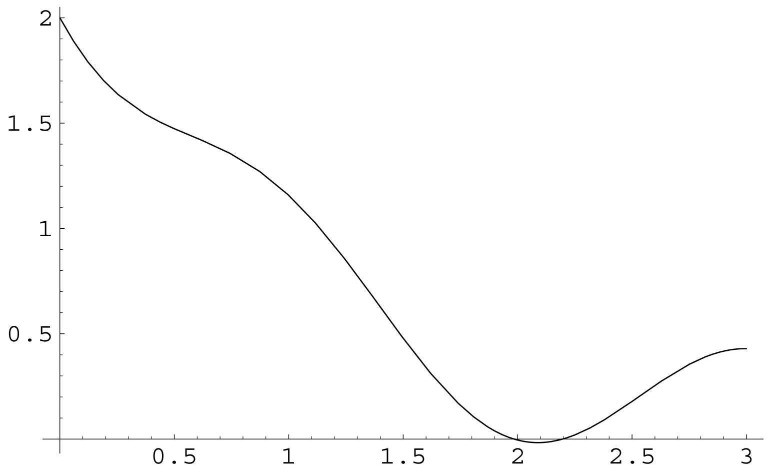
```
sol[2]
```

$$-\frac{1}{10}\, e^{-x}\, (-23 + 3\, e^{x}\, \mathrm{Cos}[3\, x] - e^{x}\, \mathrm{Sin}[3\, x])$$

```
graph[y0_] := Module[{expr}, expr = sol[y0]; Plot[expr, {x, 0, 3}]]
```

We try this out.

```
graph[2]
```



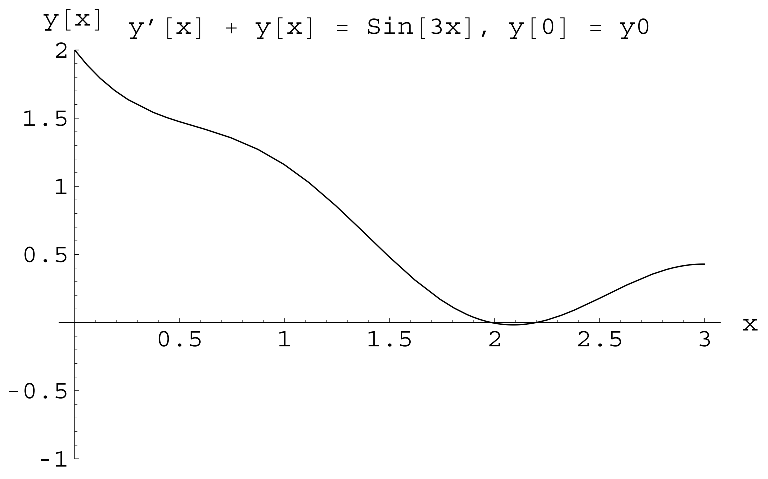
```
- Graphics -
```

Now we add some labels to the axes, and a label for the plot, and we fix the y-range of the plot to be {-1,2}.

```
graph[y0_] := Module[{expr}, expr = sol[y0]; Plot[expr,
   {x, 0, 3}, PlotRange → {-1, 2}, AxesLabel → {"x", "y[x]"},
   PlotLabel → "y'[x] + y[x] = Sin[3x], y[0] = y0"]]
```

We try it for y0 = 2.

```
graph[2]
```



```
- Graphics -
```

We make one more change. We want to display a sequence of curves for different y0 on a single graph, but we don't want to display each one separately. So we add an option to the Plot command that causes the graph not to be displayed. The option is DisplayFunction -> Identity.

```
graph[y0_] := Module[{expr}, expr = sol[y0]; Plot[expr,
   {x, 0, 3}, PlotRange → {-1, 2}, AxesLabel → {"x", "y[x]"},
   PlotLabel → "y'[x] + y[x] = Sin[3x], y[0] = y0",
   DisplayFunction → Identity]]
```

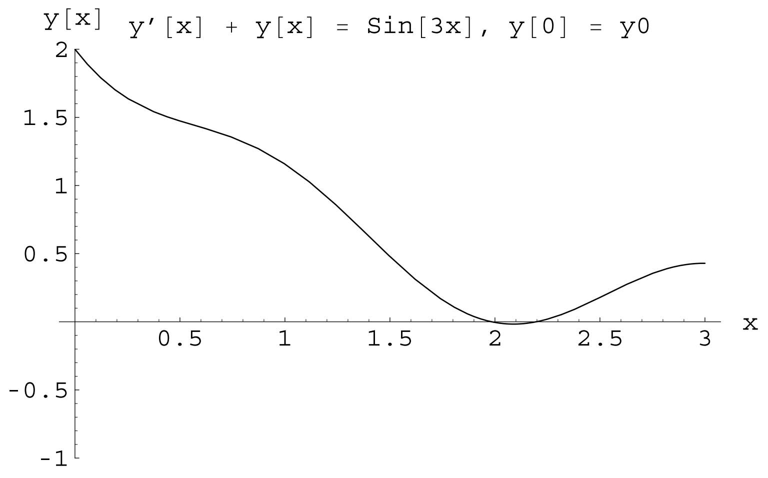
Now the graph can be constructed without being displayed.

```
testgraph = graph[2]
```

```
- Graphics -
```

To see this graph which has already been constructed, we use the Show command with the option DisplayFunction -> $DisplayFunction.

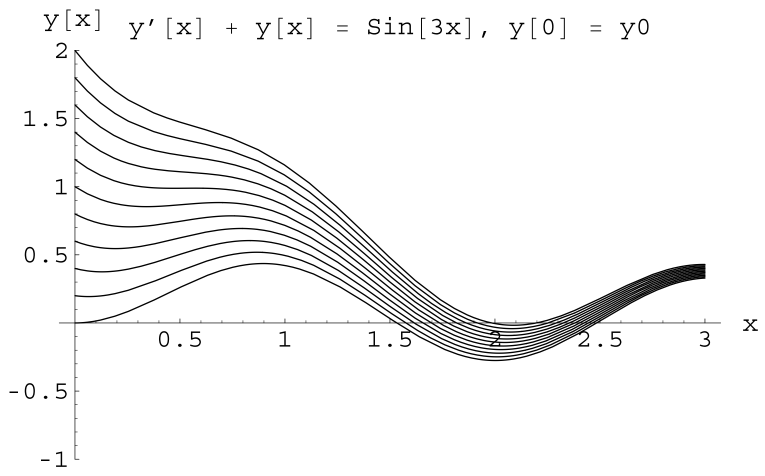
```
Show[testgraph, DisplayFunction → $DisplayFunction]
```



```
- Graphics -
```

Now we construct the solutions for 10 different initial conditions, ranging from 0 to 2 in increments of 0.2. We show them all on the same graph.

```
(compgraph = {}; Do[y0 = i * 0.2;
  compgraph = Append[compgraph, graph[y0]], {i, 0, 10}];
 Show[compgraph, DisplayFunction → $DisplayFunction])
```



```
- Graphics -
```

A few comments on the code. The set of graphs is accumulated in the list named compgraph, which is initially set equal to the empty list. The Do loop constructs the graphs for the different initial conditions, and uses the command Append to add them to compgraph. The final Show command displays all of the graphs in the list compgraph.

All of the solutions seem to be converging onto the same curve. Let's see if we can understand that. We construct the solution for an initial condition y0 = *a*, with *a* unspecified numerically.

```
sol[a]
```

$$-\frac{1}{10}\,e^{-x}\left(-10\left(\frac{3}{10}+a\right)+3\,e^{x}\,\mathrm{Cos}[3\,x]-e^{x}\,\mathrm{Sin}[3\,x]\right)$$

We force *Mathematica* to carry out the multiplication commands by using Expand.

```
Expand[%]
```

$$\frac{3\,e^{-x}}{10}+a\,e^{-x}-\frac{3}{10}\,\mathrm{Cos}[3\,x]+\frac{1}{10}\,\mathrm{Sin}[3\,x]$$

Now we see that as *x* gets large, the solution reduces to the Cos and Sin part, which is independent of *a*. Thus all initial conditions produce solutions which look the same for *x* sufficiently large.