Mathematica Tutorial: Solving Ordinary Differential Equations

Kurt Bryan and SIMIODE

This is intended as a very brief introduction to using Mathematica to solve ordinary differential equations (ODEs). The focus is primarily on first-order equations, but there is a second-order example as well.

Example 1: Consider the ordinary differential equation u'(t) = u(t) + t. The ***DSolve*** command can be used to find the general solution to this ODE, as

In[8]:=
```
DSolve[u'[t] == u[t] + t, u, t]
```

There are three arguments to the ***DSolve*** command: The first argument is ODE itself, u'[t] = u[t] + t, the second argument is u, which indicates to Mathematica that u is the dependent or unknown quantity to be found. The last argument is the independent variable t. The general solution returned by Mathematica contains an unspecified constant.

An alternative for better readability is to define the ODE separately, say as

In[9]:=
```
de = u'[t] == u[t] + t
```

and then call the ***DSolve*** command:

In[10]:=
```
DSolve[de, u, t]
```

Example 2: Let us solve the ODE from Example 1 but with an initial condition, specifically, u[0] = 2. We'll then plot the solution. We already defined the ODE in the variable "de" above, so we can execute

In[11]:=
```
DSolve[{de, u[0] == 2}, u, t]
```

In[13]:=
```
sol = DSolve[{de, u[0] == 2}, u, t]
```

We can turn this into a usable function "usol" into which we can substitute values for plot as

In[14]:=
```
usol = u /. sol[[1]]
```

The solution can then be evaluated using standard mathematical functional notation, as

In[15]:=
```
usol[2]
```

Or we can plot as

In[16]:=
```
Plot[usol[t], {t, 0, 3}]
```

Example 3: Let's solve the second order ODE u''(t) + 3*u'(t) + 2*u(t) = sin(t) with initial conditions u(0) = 1 and u'(0) = 2. Define the ODE as

In[17]:=
```
de = u''[t] + 3 * u'[t] + 2 * u[t] == Sin[t]
```

Call the ***DSolve*** command with the initial data, as

In[18]:=
```
sol = DSolve[{de, u[0] == 1, u'[0] == 2}, u, t]
```

Note that we named the output of the dsolve command. This makes it easier to pick off the right hand side that defines the solution as a function of t and turn it into a function, as

In[19]:=
```
usol = u /. sol[[1]]
```

Now we can easily evaluate the solution at a given point or plot

In[20]:=
```
usol[3.1]
```

In[21]:=
```
Plot[usol[t], {t, 0, 20}]
```

Practice with DSolve

Problem 1: Use ***DSolve*** to find the general solution to u'(t) = 3*u(t).
Problem 2: Use ***DSolve*** to find the solution to u'(t) = 3*u(t) with initial condition u(0)=2. Evaluate the solution at t= 0.3 and plot the solution on the interval t = 0 to t = 1.
Problem 3: Use ***DSolve*** to find the solution to 2*u''(t) + 4*u'(t) + 4*u(t) = 0 with initial conditions u(0) = 0, u'(0)=2. Plot the solution on the interval t = 0. to t = 10.