

# Maple Tutorial: Solving Ordinary Differential Equations

Kurt Bryan and SIMIODE

This is intended as a very brief introduction to using Maple 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

```
> dsolve(u'(t) = u(t) + t, u(t))
```

There are two arguments to the *dsolve* command: The first argument is ODE itself,  $u'(t) = u(t) + t$ , and the second argument is  $u(t)$ , which indicates to Maple that  $u(t)$  is the dependent or unknown quantity to be found. The second argument  $u(t)$  can be omitted, but it is good practice to be explicit about what variable is being solved for, especially in more complicated situations. The general solution returned by Maple contains an unspecified constant, which Maple labels with a leading underscore (to avoid conflict with any constants that the user may have elsewhere defined).

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

```
> de := u'(t) = u(t) + t
```

(note Maple outputs the notation  $D(u)(t)$  for the derivative) and then calling the *dsolve* command,

```
> 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

```
> dsolve({de, u(0) = 2}, u(t))
```

The arguments to *dsolve* here are the ODE itself ("de") and initial condition, enclosed in curly braces {}, followed by the dependent variable  $u(t)$ .

Maple returned the solution above, but not in a form we can easily plot or into which we can substitute values. We can turn the output into a more usable form as follows. First, it's usually a good idea to name the output of the *dsolve* command (and more generally, the output of any Maple computation).

Thus, we execute

```
> sol := dsolve({de, u(0) = 2}, u(t))
```

The variable "sol" now contains the solution to the ODE of interest. We can pick off the right side of the solution as

```
> rhs(sol)
```

We can define a Maple variable "usol" as

```
> usol := rhs(sol)
```

The drawback here is that "usol" is not a function, so, for example,  $usol(5)$  will not return the solution value at  $t = 5$ . That can be accomplished with the syntax `"subs(t=5, usol)"`.

A convenient alternative is to define usol as a Maple function, as

```
> usol := unapply(rhs(sol), t)
```

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

```
> usol(2)
```

Or plotted as

```
> plot(usol(t), t=0..3)
```

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

```
> de := u''(t) + 3*u'(t) + 2*u(t) = sin(t)
```

Call the `dsolve` command with the initial data, as

```
> 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

```
> usol := unapply(rhs(sol), t)
```

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

```
> usol(3.1)
```

```
> plot(usol(t), t=0..20)
```

## Practice with dsolve

**Problem 1:** Use ***dsolve*** to find the general solution to  $u'(t) = 3u(t)$ .

**Problem 2:** Use ***dsolve*** to find the solution to  $u'(t) = 3u(t)$  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  $2u''(t) + 4u'(t) + 4u(t) = 0$  with initial conditions  $u(0) = 0$ ,  $u'(0)=2$ . Plot the solution on the interval  $t=0$  to  $t=10$ .