Differential Equations

Table of Contents

Introduction

Matlab is quite powerful when it comes to solving differential equations. The standard command is dsolve (think Differential Solve) and has the format `dsolve('equation','variable')`. Here equation is a differential equation in the form of a string and variable is the independent variable. For example suppose we wished to solve `y'=2y+x`.

We type this into Matlab as:

```
dsolve('Dy=2*y+x')
```

```
ans =

(C4*exp(2*t))/2 - x/2
```

Note that `Dy` represents the derivative of the variable y. Also we should note here that you may get a differently numbered constant `C1`, `C2`, etc. depending upon how many constants that have shown up in Matlab for you. Don't worry for now about how they're numbered.

Having No Independent Variable.

Consider the differential equation `y'=3y`. If you do this in Matlab:

```
dsolve('Dy=3*y')
```

```
ans =

C6*exp(3*t)
```

You see the solution is given as a function of `t`. Why `t`? The answer is that first, the differential equation has no independent variable so Matlab doesn't know what `y` is supposed to be a function of. Second, `t` is pretty normal since often differential equations are used for functions of time. If you want another variable you can tell Matlab:

```
dsolve('Dy=3*y','x')
```

```
ans =

C8*exp(3*x)
```

We can use higher derivatives like `D2y` and `D3y` too. For example we can solve `y''=2y` treating `y` as a function of `z` by typing:

```
dsolve('D2y=2*y','z')
```

```
ans =

C10*exp(2^(1/2)*z) + C11*exp(-2^(1/2)*z)
```

Initial Value Problems

We can also state initial conditions using the form `dsolve('equation','initial condition','variable')` where initial condition is also a string. For example:

```
dsolve('Dy=2*y','y(0)=5','x')
```

```
ans =

5*exp(2*x)
```

And higher order with multiple initial conditions:

```
dsolve('D2y+Dy=x','y(0)=1,Dy(1)=2','x')
```

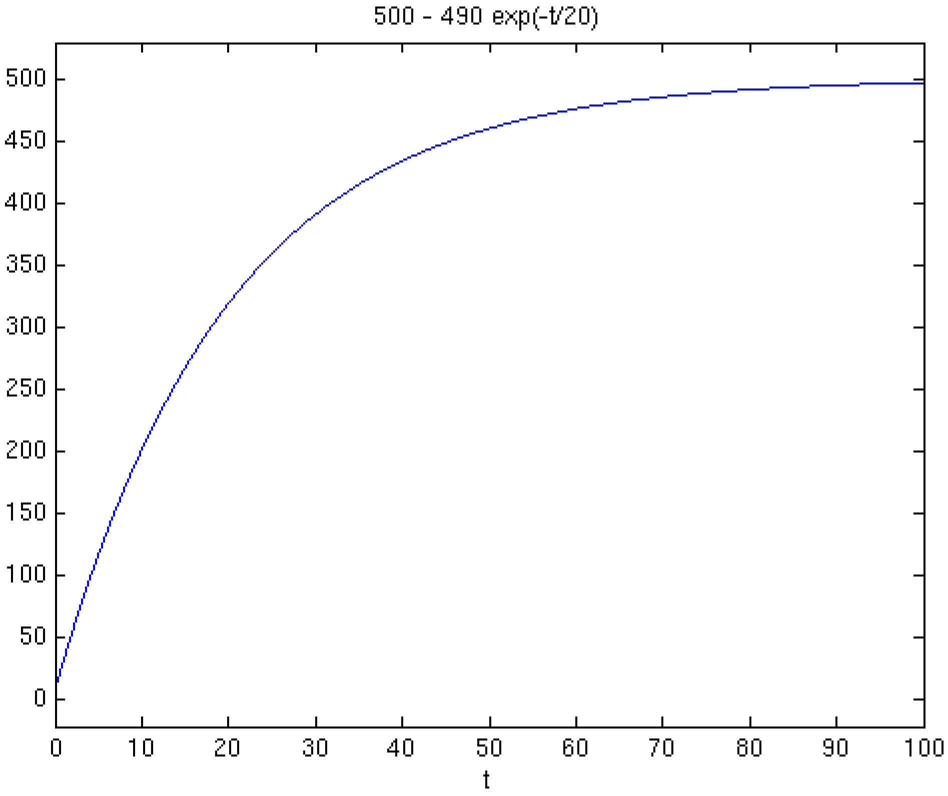
```
ans =

2*exp(1) - x - 2*exp(1)*exp(-x) + x^2/2 + 1
```

Matlab can of course do much more with differential equations as we'll see but for now just appreciate that it can handle most straightfoward examples with no problem at all.

Plotting a solution.

Plotting a solution is as easy as wrapping `dsolve` in `ezplot`:

```
ezplot(dsolve('Dy=0.05*(500-y)','y(0)=10','t'),[0,100])
```



Compound Example

Here's an example of a single Matlab line which will solve the initial value problem `y'+3y+10=0` with `y(1)=2`, set the result equal to `0` and solve for `x`.

```
solve(dsolve('Dy+3*y+10=0','y(1)=2','x'),'x')
```

```
ans =

-log((5*exp(-3))/8)/3
```

You can also set the result to a nonzero value with this totally confusing command:

```
solve(strcat(char(dsolve('Dy+3*y+10=0','y(1)=2','x')),'=20000'))
```

```
ans =

-log((30005*exp(-3))/8)/3
```

An Example with Variables

You may notice that if you try the following:

```
a=2;b=3;
dsolve('Dy=a*x+b','x')
```

```
ans =

(a*x^2)/2 + b*x + C2
```

This is annoying. You wanted `a` and `b` to be in the answer! The point is that `'Dy=a*x+b'` is treated as a string of characters and therefore `a` and `b` as just letters. They're not given their values. To get around this you can solve and then substitute:

```
clear all;
subs(dsolve('Dy=a*x+b','x'),{'a','b'},{2,3})
```

```
ans =

x^2 + 3*x + C2
```

Published with MATLAB® 8.0