Math 103: Algebra 1

Objectives

- Scripts in Matlab
- Functions in Matlab
- Plotting in Matlab
- Solving f(x) = 0

Algebra

The purpose of algebra is, essentially, solving mathematical equations. For example, a thermistor is a sensor whose resistance is related to temperature, such as

$$R = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

A photoresistor is a sensor whose resistance is related to light as

$$R = \left(\frac{10{,}000}{(Lux)^{0.6}}\right)\Omega$$

In each of these examples, given temperature or Lux, you can plug in numbers and solve for the resistance. A harder problem is going the other way: given resistance, solve for temperature or Lux. Algebra provides tools for doing this.

A second example of where algebra is used is when dealing with two equations with two unknowns. A common problem in electrical and computer engineering is solving for the voltages in a circuit where you have two equations:

$$3V_1 + 5V_2 = 10$$

$$V_1 - V_2 = 6$$

Algebra can be used to solve this problem as well.

In this lecture, we will be covering

- Rules of Algebra: valid ways to manipulate mathematical equations
- Plotting mathematical relationships,
- Solving a mathematical equation using graphical techniques, and
- Solving a mathematical equation using numerical techniques.

Rules of Algebra

Consider an equation such as

$$A = B$$

Equals is a very powerful symbol: it means the two sides are identical and interchangeable. When you manipulate this equation, you have to do it in a way that keeps the two sides identical. Essentially, whatever you do to the left side, you have to do to the right side as well.

Some things that are allowed are as follows:

Addition: You can add or subtract the same value from both sides. For example

$$A + 5 = B + 5$$

Multiplication: You can multiply or divide both sides by the same number:

$$(A + 5) \cdot 7 = (B + 5) \cdot 7$$

Distribution: When multiplying stuff within parenthesis, you have to multiply each element

$$(A + 5) \cdot 7 = A \cdot 7 + 5 \cdot 7$$

Commutative Property: The order of addition and multiplication doesn't matter

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

Some other useful properties relate to ln() and exp()

$$\exp(x) \equiv e^x$$

$$\exp(\ln(x)) = x$$

$$\ln(\exp(x)) = x$$

Multiplying by one: You can multiply one side of the equation by one and still have a valid equation

$$A \cdot 1 = A$$

$$A \cdot \left(\frac{B}{B}\right) = A$$

Adding Zero: You can add zero to one side and still have a valid equation

$$A + 0 = A$$

$$A + (B - B) = A$$

Multiplying by Zero: This is a no-no: multiplying by zero makes anything work.

$$5 \cdot 0 = 3 \cdot 0$$

Dividing by zero: This is also a no-no: it also makes anything work

$$\frac{A}{0} = \frac{B}{0} = \text{undefined (or infinity)}$$

For example, determine the value of X:

$$\left(\frac{4(x+6)-7(2x+3)}{15+2x}\right) = 25$$

Multiply both sides by (15+2x) to clear the fraction

$$\left(\frac{4(x+6)-7(2x+3)}{15+2x}\right)(15 + 2x) = 25(15 + 2x)$$

$$4(x + 6) - 7(2x + 3) = 25(15 + 2x)$$

Multiply out each term

$$(4x + 24) - (14x + 21) = (375 + 50x)$$

Group terms and simplify

$$-10x + 3 = 375 + 50x$$

Add 10x to each side

$$(-10x + 3) + (10x) = (375 + 50x) + (10x)$$

$$3 = 375 + 60x$$

Subtract 375 from each side

$$3 - 375 = 375 + 60x - 375$$

$$-372 = 60x$$

Divide both sides by 60

$$\frac{-372}{60} = \frac{60x}{60} = x$$

Proof that 2 = 1

Using these rules, you can prove that 2 = 1. Assume

$$a = b = 1$$

Multiply both sides by a:

$$a \cdot a = ab$$

Subtract b^2 from both sides:

$$a^2 - b^2 = ab - b^2$$

note:

$$(a+b)(a-b) = a^2 + ab - ab - b^2 = a^2 - b^2$$

Rewrite the left and right sides as

$$(a+b)(a-b) = b(a-b)$$

Divide both sides by (a-b)

$$\frac{(a+b)(a-b)}{(a-b)} = \frac{b(a-b)}{(a-b)}$$

$$a + b = b$$

Substitute a = b = 1

$$2 = 1$$

Look over the proof and see if you can find the mistake (next page).

The problem with this proof is line 5:

$$(a+b)(a-b) = b(a-b)$$

$$2 \cdot 0 = 1 \cdot 0$$

While this is valid, canceling the zeros is not valid: you can't divide by zero

$$2 \neq 1$$

Application of Algebra (take 1)

Suppose a thermistor is 500 Ohms. What temperature is it?

$$R = 500\Omega = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

Solution: Divide both sides by 1000

$$\frac{500}{1000} = \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)$$

Take the natural log of both sides

$$\ln\left(\frac{1}{2}\right) = \frac{3905}{T+273} - \frac{3905}{298}$$

Add 3905/298 to both sides

$$\ln\left(\frac{1}{2}\right) + \frac{3905}{298} = \frac{3905}{T+273}$$

Take the inverse of both sides

$$\left(\ln\left(\frac{1}{2}\right) + \frac{3905}{298}\right)^{-1} = \frac{T+273}{3905}$$

Multiply both sides by 3905

$$3905\left(\ln\left(\frac{1}{2}\right) + \frac{3905}{298}\right)^{-1} = T + 273$$

Subtract 273 from both sides

$$3905\left(\ln\left(\frac{1}{2}\right) + \frac{3905}{298}\right)^{-1} - 273 = T$$

Solving Two Equations for Two Unknowns (Gauss Elimination)

Using rules of algebra, you can also solve two equations with two unknowns. For example, determine x and y:

$$2x + 3y = 10$$

$$3x - 7y = 4$$

Equals is a very powerful symbol: it tells you that (2x+3) and (10) are identical and interchangeable. This means that you can add the equations

$$(2x + 3y) + (3x - 7y) = (10) + (4)$$

To be useful, add them in a way that one term (x) cancels.

- Multiply the first equation by 3
- Multiply the second equation by 4
- Subtract

$$3(2x + 3y = 10)$$

$$-2(3x - 7y = 4)$$

$$(6x - 6x) + (9y + 14y) = 30 - 12$$

$$23y = 18$$

$$y = \frac{18}{23}$$

x comes from either equation

$$2x + 3y = 10$$

$$2x + 3\left(\frac{18}{23}\right) = 10$$

$$2x = 10 - 3\left(\frac{18}{23}\right)$$

$$x = \frac{1}{2}\left(10 - 3\left(\frac{18}{23}\right)\right)$$

Solving Equations using Graphical Techniques

Another way to solve algebraic equations to plot them in Matlab. For example, for a thermistor

$$R = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

you can calculate the resistance at 10 degrees C in Matlab as

```
>> T = 10;
>> R = 1000 * exp( 3905/(T+273) - 3905/298)

R =  2.0028e+003
```

You can calculate the resistance from -20C to +20C as

```
>> T = [-20:10:20]'

   -20
   -10
     0
    10
    20

>> R = 1000 * exp( 3905 ./(T+273) - 3905/298)

  1.0e+004 *

    1.0286          resistance at -20C
    0.5720
    0.3320
    0.2003
    0.1251          resistance at +20C
```

Note: Matlab is a matrix language. The Matlab command

```
A * B
```

does a matrix multiply. The Matlab command

```
A .* B
```

does an element-by-element multiply (it treats this as separate scalar operations stacked together).

Repeating with more points and plotting:

```
>> T = [-20:0.01:20]';
>> R = 1000 * exp( 3905 ./(T+273) - 3905/298);
>> plot(T,R)
>> xlabel('Degrees C');
>> ylabel('Ohms');
```

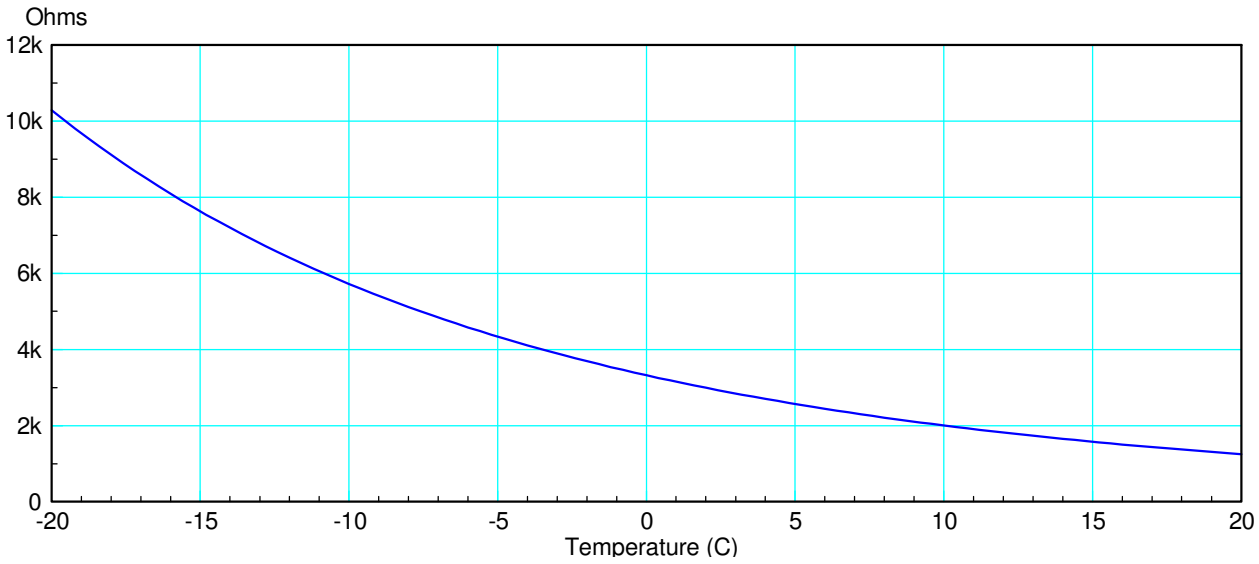

Figure 1: A graph shows the relationship between temperature and resistance

This graph is another way to

- Determine the resistance given temperature, and
- Determine the temperature given the resistance.

For example,

- If the temperature is -10C, the resistance is 5800 Ohms
- If the resistance is 5800 Ohms, the temperature is -10C

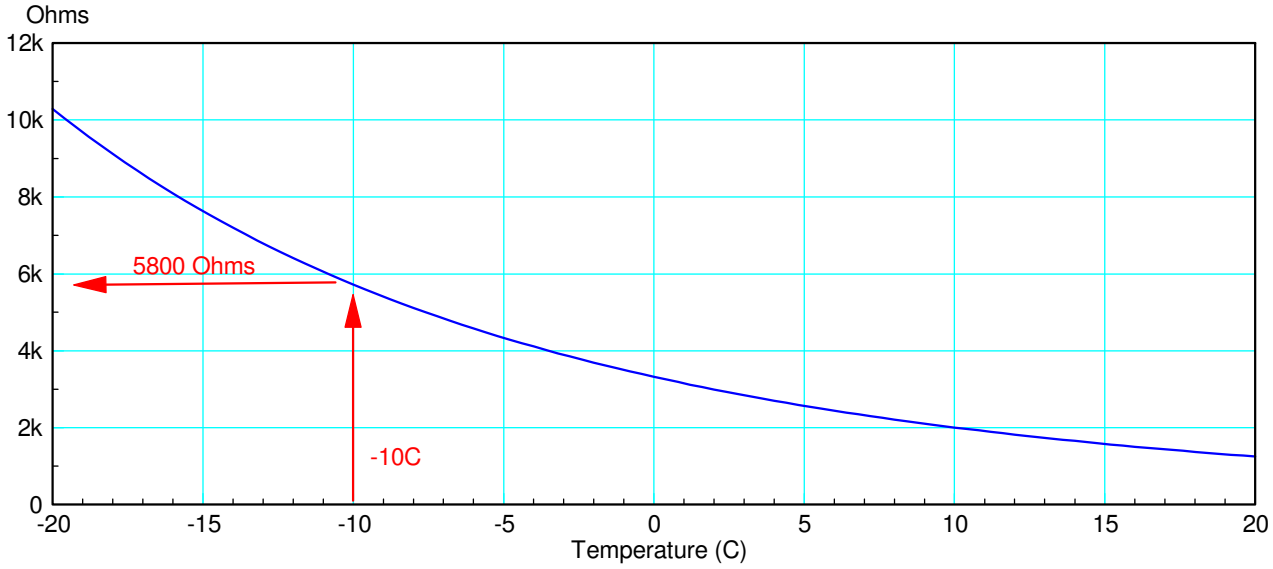

Figure 2: A graph allows you to determine the resistance given the temperature

Example #2: A thermistor is connected to a voltage divider:

$$R = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

$$V = \left(\frac{R}{R+1000}\right) \cdot 10V$$

Plot the temperature - voltage relationship.

Solution: In Matlab, input T, compute R, and compute V:

```
>> T = [-20:0.4:20]';
>> R = 1000 * exp( 3905 ./(T+273) - 3905/298);
>> V = R ./ (1000+R) * 10;
>> plot(T,V);
>> xlabel('Degrees C');
>> ylabel('Volts');
```

From the graph

- If the temperature is -10C, the voltage is 8.5V
- If the voltage is 8.5V, the temperature is -10C

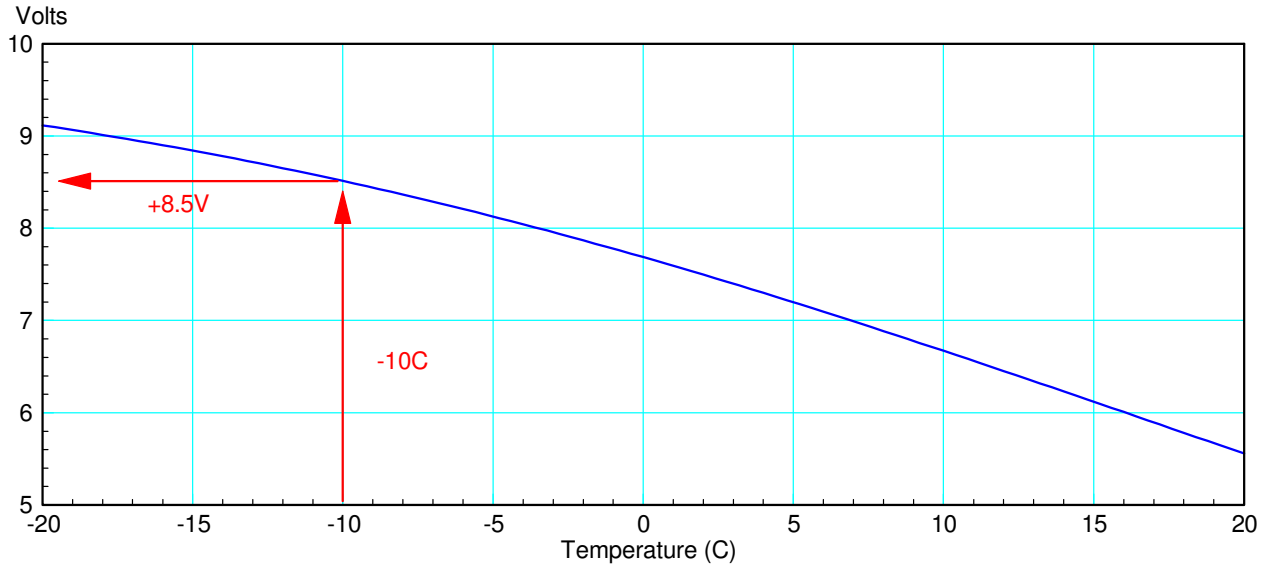

Figure 3: The graph tells you the voltage when the temperature is -10C

Solving f(x) = 0 Using Numerical Techniques

- Matlab Scripts
- Matlab Functions

Scripts and functions are slightly different in Matlab:

- Scripts are similar to instructions you type in the command window. When you run a script, Matlab acts like you just typed everything in the script into the command window.
- Functions, in contrast, are subroutines you can call. For example, plot() is a function.

Unlike scripts, you cannot execute a function. Instead, it has to be called by someone else.

For example, let's write a function called *Therm* which

- Is passed the temperature, and
- Returns the resistance of thermistor with the R-T relationship of

$$R = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)$$

Initially, in Matlab if you try to call this function from the command window, you'll get an error message

```
>> Therm(15)
??? Undefined function or method 'Therm' for input arguments of type
'double'.
```

What Matlab is doing when you type in *Therm(15)* is

- If first checks if there is a varable called *Therm*. If so, it returns the 15th element of that array.
- If no variable *Therm* exists, it then checks if there is a file called *Therm.m* If Matlab finds that file, it then tries to call it.
- If that fails, then an error message is given: Matlab can't find *Therm* and doesn't know what to do.

To Create a function called *Therm*, go to File - New Fuction and type in the following:

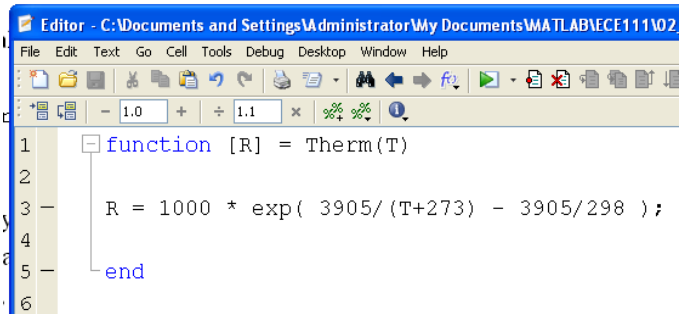

Figure 4: Example of a Matlab function

Now save this in the default directory with the default name, *Therm.m*

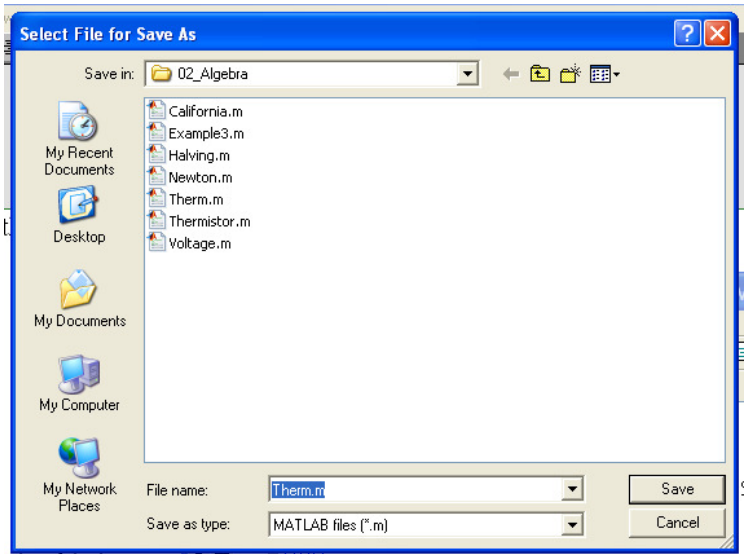

Figure 5: When saving a funciton, the file name should match the function name, with a dot-m extension

The keyword *function* tells Matlab that this is a subroutine: you cannot run it but you can call it from the command window. For example, if you want to know the resistance when T = 0C, type in:

```
>> Therm(0)

ans =  3.3201e+003
```

Here, you called the m-file Therm.m, passing the number 0. It took this number, then computed and returned the resistance: 3320.1 Ohm.

If you want to know the resistance at 30C, pass the number 30:

```
>> Therm(30)

ans =  805.5435
```

The resistance at 30C is 805.5435 Ohms.

Now let's return to the problem of finding the temperature given the resistance. Assume

$$R = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

One way to solve this problem is to change this to a function-equals-zero problem. Rewrite this as:

$$f(T) = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right) - R$$

The solition is whatever T is that makes f(T) = 0.

Solving f(x) = 0 is a very common type of problem in electrical and computer engineering. With Matlab, you can solve this sort of problem several ways.

First, let's create a function in matlab where

- We pass the temperature, T, and
- It returns f(T).

The goal is to find the value of T that results R = 2000 Ohms or f(T) = 0

```
function [e] = Thermistor(T)

   R0 = 2000;    % resistance of thermistor

   R = 1000 * exp( 3905/(T+273) - 3905/298 );
   e = R - R0;

   end
```

Save this file in Matlab in the default directory with the name **Thermistor.m**

Once you save this file as Thermistor.m, you now have a new function you can call from the command window.

```
>> Thermistor(0)

ans =  1.3201e+003

>> Thermistor(30)

ans = -1.1945e+003
```

The temperature that results in an error of zero is the correct temperature (somewhere between 0C and 30C).

Interval Halving: Start with two guesses

- Guess #1 has a positive result (0C)
- Guess #2 has a negative result (30C)

The next guess is the midpoint between the two (+15C)

- If this result is positive, replace guess #1
- If the result is negative, replace guess #2

Repeat

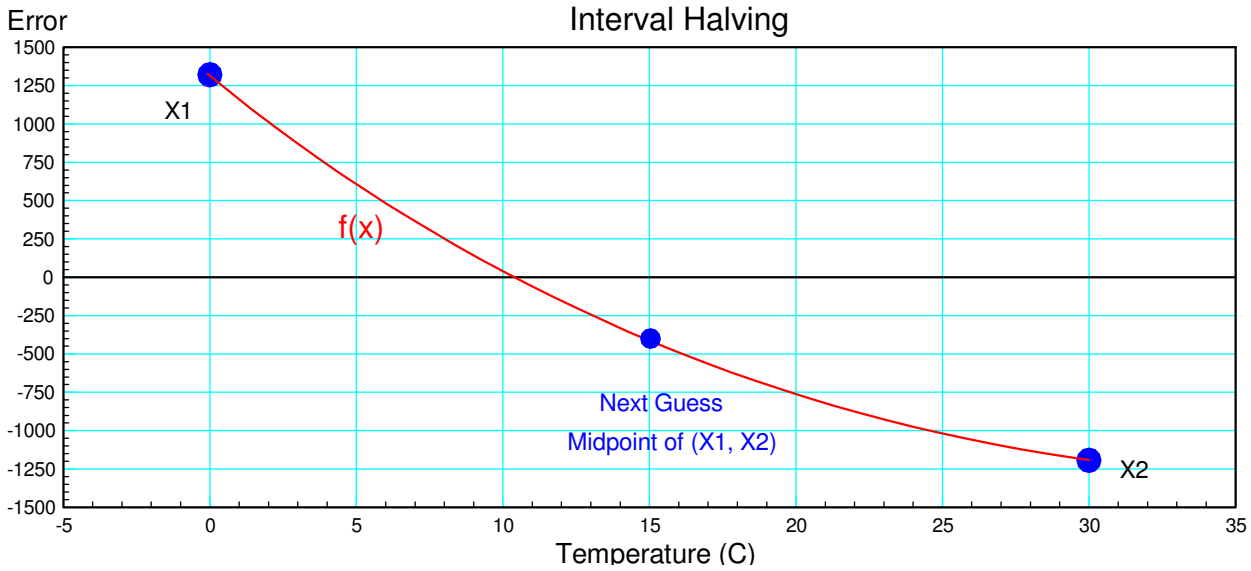

Figure 6: Example of interval halving. The next guess is always the midpoint between the previous two guesses

You can place this in a Matlab script (iterates ten times)

```
X1 = 0;
X2 = 30;

for n=1:20
   X3 = (X1+X2)/2;
   Y3 = Thermistor(X3);

   if(Y3 > 0)
      X1 = X3;
   else
      X2 = X3;
   end

   disp([n X3, Y3]);
end
```

The result converges to the answer (10.0290C)

n	T	error
1.0000	15.0000	-423.8251
2.0000	7.5000	264.9253
3.0000	11.2500	-115.0903
4.0000	9.3750	64.9398
5.0000	10.3125	-27.4254
6.0000	9.8438	18.1523
7.0000	10.0781	-4.7855
8.0000	9.9609	6.6459
9.0000	10.0195	0.9208
10.0000	10.0488	-1.9347
11.0000	10.0342	-0.5075
12.0000	10.0269	0.2065
13.0000	10.0305	-0.1505
14.0000	10.0287	0.0280
15.0000	10.0296	-0.0613
16.0000	10.0291	-0.0167
17.0000	10.0289	0.0057
18.0000	10.0290	-0.0055
19.0000	10.0290	0.0001
20.0000	10.0290	-0.0027

California Method: Take one shot that's long and one that's short. Your next guess is interpolated between the two as

$$X_3 = X_1 + \left(\frac{\delta X}{\delta error}\right) E_1$$

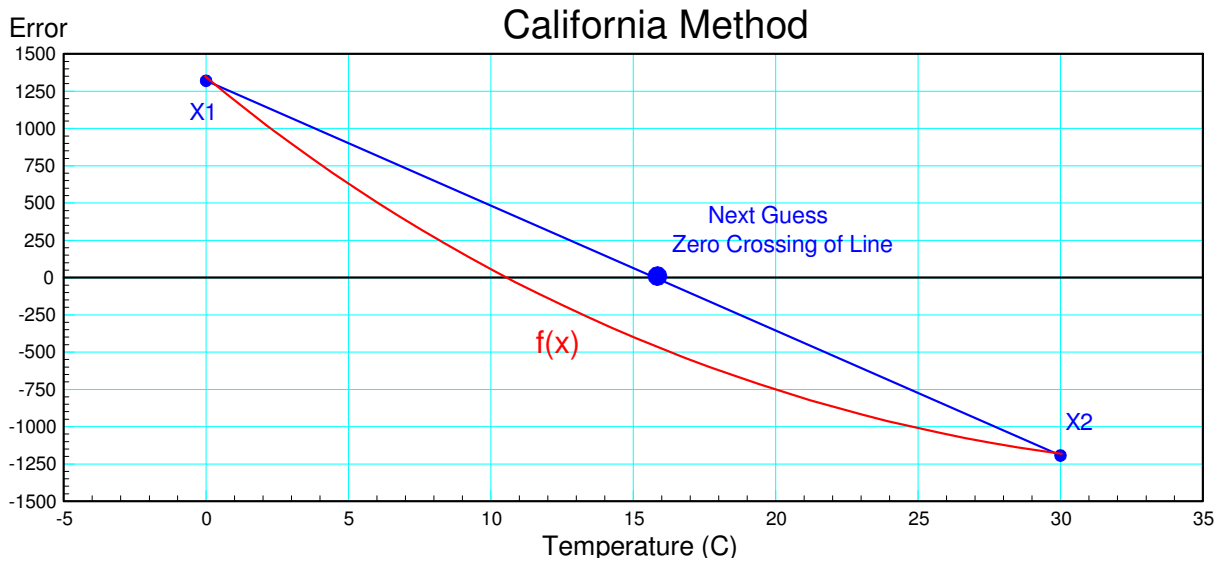

Figure 7: With the California Method, the next guess is the zero-crossing found using lienar interpolation

For example, with the above data, guess #3 would be

```
X1 = 0;
Y1 = Thermistor(X1);
X2 = 30;
Y2 = Thermistor(X2);

for n=1:10
   X3 = X2 - (X2-X1)/(Y2-Y1)*Y2;
   Y3 = Thermistor(X3);

   if(Y3 > 0)
      X1 = X3;
      Y1 = Y3;
   else
      X2 = X3;
      Y2 = Y3;
   end

   disp([n, X3, Y3]);
end
```

Result

```
       x        T       error
    1.0000   15.7496 -478.3434
    2.0000   11.5607 -143.1496
    3.0000   10.4297  -38.6368
    4.0000   10.1331  -10.1259
    5.0000   10.0560   -2.6331
    6.0000   10.0360   -0.6833
    7.0000   10.0308   -0.1772
    8.0000   10.0294   -0.0460
    9.0000   10.0291   -0.0119
   10.0000   10.0290   -0.0031
```

Note that California Method converges much faster than interval halving.

Newton's Method: Take a guess. Take another guess slightly larger. Based upon this, determine where the zero-crossing would be if it were a linear system.

$$X_2 = X_1 - \left(\frac{\delta X}{\delta e}\right) e_1$$

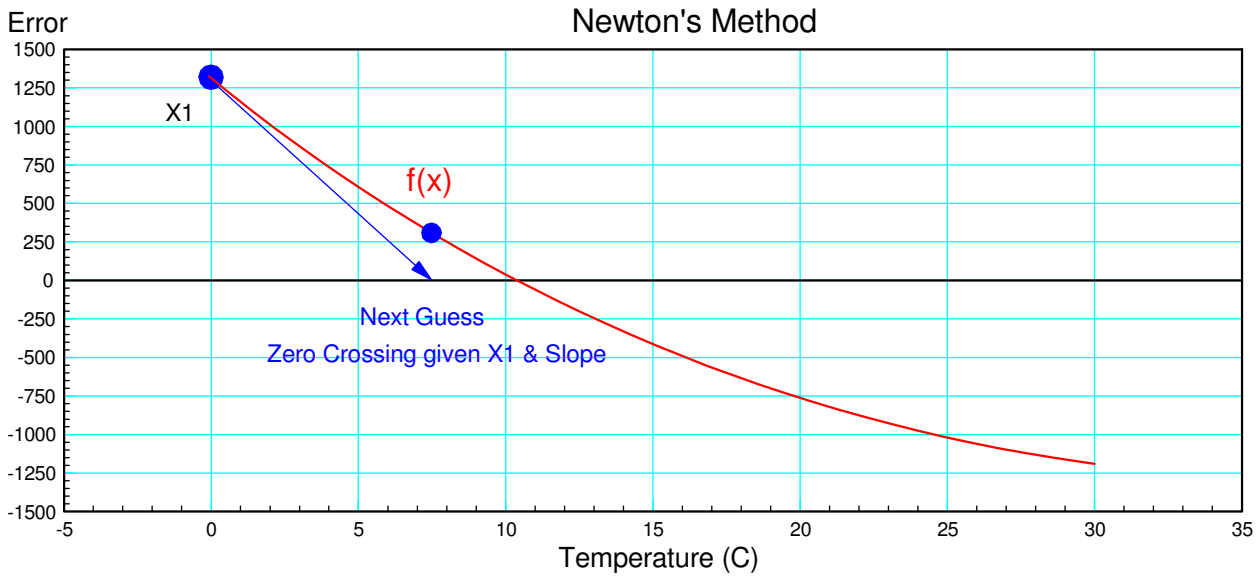

Figure 8: With Newton's method, the slope and error determine the next guess

Example:

```
X3 = 0;

for n=1:10
   X1 = X3;
   Y1 = Thermistor(X1);
   X2 = X1 + 0.01;
   Y2 = Thermistor(X2);
   X3 = X2 - (X2-X1)/(Y2-Y1)*Y2;
   disp([n, X1, Y1]);
   X1 = X3;
   end
```

Result

```
    n           T       error
1.0000         0    1.3201
2.0000    7.5909  254.7309
3.0000    9.8694   15.6317
4.0000   10.0283    0.0648
5.0000   10.0290   -0.0000
```

The nice thing about Newton's method is

- If converges very fast, and
- Any initial guess works: you don't need one guess that's high and one thet's low

Newton's Method with Voltage:

Going back to a thermistor and a voltage divider, assume

$$R = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

$$V = \left(\frac{R}{R+1000}\right) \cdot 10V$$

Find the temperature when

- V = 8V
- V = 7V
- V = 6V

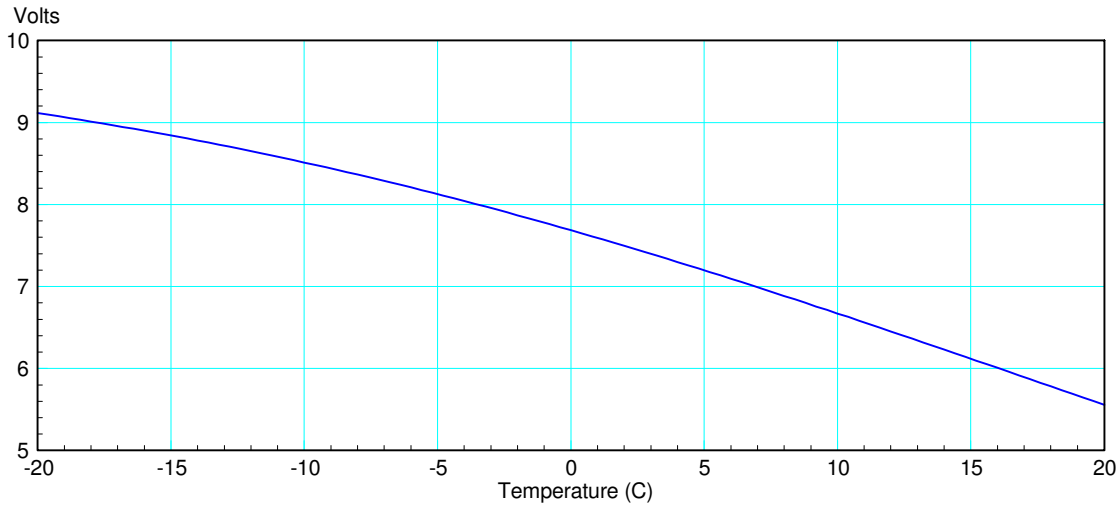

Figure 8: Temperature - voltage relationship for a thermistor - voltage divider example

Solution using Newton's Method: Create a Matlab function which

- Is passed your guess at the temperature, T, and
- Returns the error in the voltage

```
function [e] = Voltage(T)
   V0 = 8.0;    % target voltage

   R = 1000 * exp( 3905/(T+273) - 3905/298 );
   V = R / (1000 + R) * 10;

   e = V - V0;

   end
```

Use Newton's method to solve

```
X3 = 0;           % initial guess

for n=1:10
   X1 = X3;
   Y1 = Voltage(X1);
   X2 = X1 + 0.01;
   Y2 = Voltage(X2);
   X3 = X2 - (X2-X1)/(Y2-Y1)*Y2;

   disp([n, X1, Y1]);
   X1 = X3;
end
```

Note: the only thing that changes is the name of the file which defines f(x).

Solving with V0 = 8.0V:

n	T	error
1.0000	0	-0.3147
2.0000	-3.3764	-0.0115
3.0000	-3.5095	-0.0000
4.0000	-3.5098	-0.0000
5.0000	**-3.5098**	-0.0000

Solving with V0 = 7.0V:

n	T	error
1.0000	0	0.6853
2.0000	7.3510	-0.0472
3.0000	6.9030	-0.0001
4.0000	6.9017	-0.0000
5.0000	**6.9017**	-0.0000

Solving with V0 = 6.0V:

n	T	error
1.0000	0	1.6853
2.0000	18.0785	-0.2272
3.0000	16.0580	-0.0002
4.0000	16.0560	-0.0000
5.0000	**16.0560**	-0.0000

Note:

- Newton's method converges very fast
- The answer is very accurate (>6 decimal places of accuracy after five iterations)
- Numeric techniques are *way* easier than using algebra to find the temperature

Solving f(x) = 0 with multiple solutions

If a function has multiple solutions, Newton's method tends to converge to the solution close to your initial guess. This means it helps to know what the solution are (and how many there are).

For example, find (x,y) which satisfy the following two equations:

$$y = \frac{\cos(3x)}{x^2+1}$$

$$y = 0.1\exp\left(\frac{x}{2}\right)$$

Method #1: Graphical Methods: Treat these as two separate functions and plot them together

$$y_1 = \frac{\cos(3x)}{x^2+1}$$

$$y_2 = 0.1\exp\left(\frac{x}{2}\right)$$

The intersections are the solutions

```
>> x = [-4:0.04:4]';
>> y1 = cos(3*x) ./ (x.^2 + 1);
>> y2 = 0.1*exp(x/2);
>> plot(x,y1,x,y2)
```

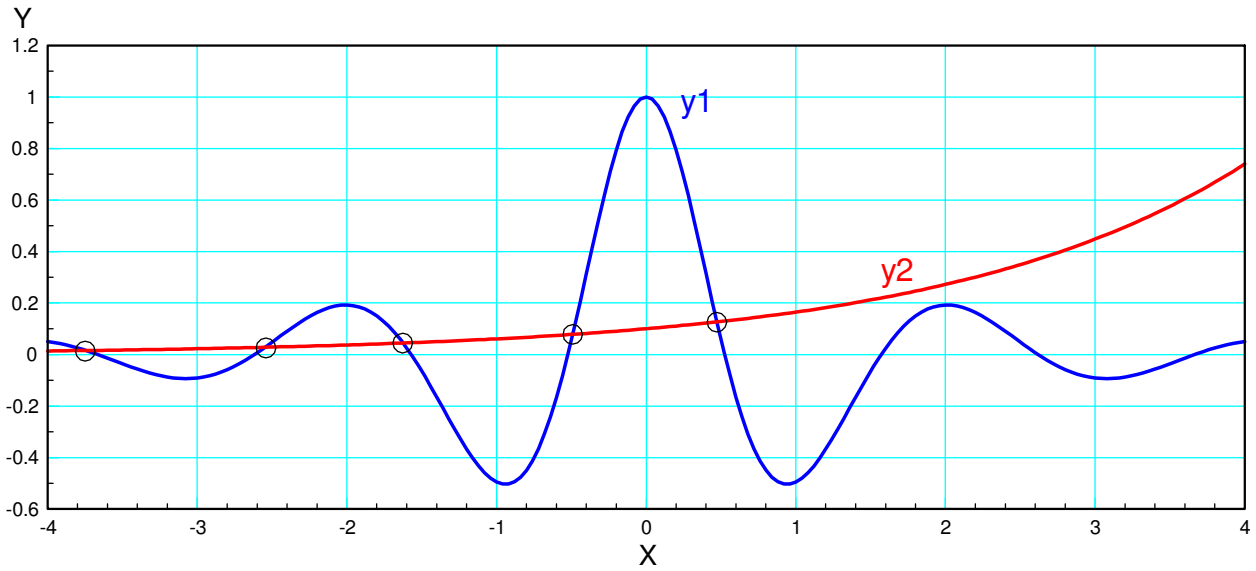

Figure 9: The intersection of the two graphs shows the solutions

Note: There are five solutions in the range of -4 < x < 4.

Method #2: Newton's Method. Create a Matlab function that returns the error: y1 - y2:

```
function [e] = Example3(x)

   y1 = cos(3*x) / (x^2 + 1);
   y2 = 0.1*exp(x/2);

   e = y1 - y2;

end
```

Use Newton's method to solve. The initial guess pretty much determines which solution you converge to:

```
X3 = -3.6;

for n=1:10
   X1 = X3;
   Y1 = Example3(X1);
   X2 = X1 + 0.01;
   Y2 = Example3(X2);
   X3 = X2 - (X2-X1)/(Y2-Y1)*Y2;

   disp([n, X1, Y1]);
   X1 = X3;

end
```

Solving: Where you end up depends upon your initial guess (at n = 1)

```
      n         x      y1-y2
 1.0000   -3.6000   -0.0305
 2.0000   -3.7343   -0.0017
 3.0000   -3.7428   -0.0000
 4.0000   -3.7429   -0.0000
 5.0000   -3.7429   -0.0000

      n         x      y1-y2
 1.0000   -2.4000    0.0599
 2.0000   -2.5498   -0.0009
 3.0000   -2.5476   -0.0000
 4.0000   -2.5476   -0.0000
 5.0000   -2.5476   -0.0000

      n         x      y1-y2
 1.0000   -1.6000   -0.0204
 2.0000   -1.6240   -0.0007
 3.0000   -1.6249   -0.0000
 4.0000   -1.6249   -0.0000
 5.0000   -1.6249   -0.0000

      n         x      y1-y2
 1.0000   -0.4000    0.2305
 2.0000   -0.4891    0.0049
 3.0000   -0.4912    0.0000
 4.0000   -0.4912    0.0000
 5.0000   -0.4912    0.0000
```

n	x	y1-y2
1.0000	0.4000	0.1902
2.0000	0.4708	0.0025
3.0000	0.4718	-0.0000
4.0000	0.4718	0.0000
5.0000	**0.4718**	-0.0000

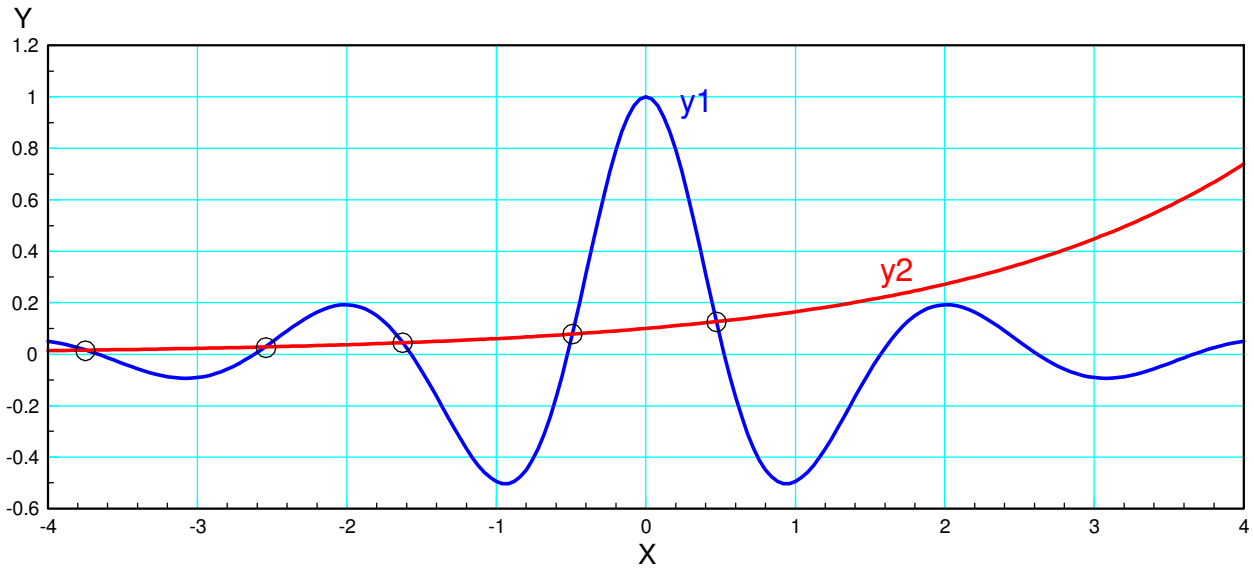

Figure 10: The five solutions are x = {-3.7429, -2.5476, -1.6249, -0.4912, 0.4718}

Summary:

Algebra is useful when you want to solve a mathematical equation.

You can also solve mathematical equations in Matlab using

- Graphical techniques, and
- Numeric techniques.

Methods with the name of Gauss or Newton tend to be really good methods.