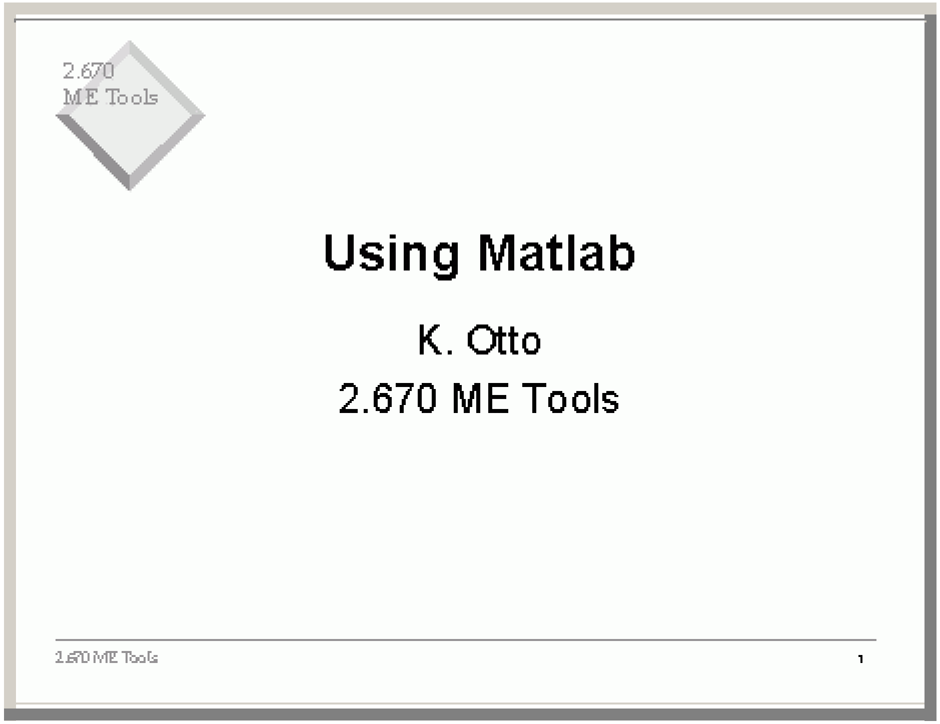
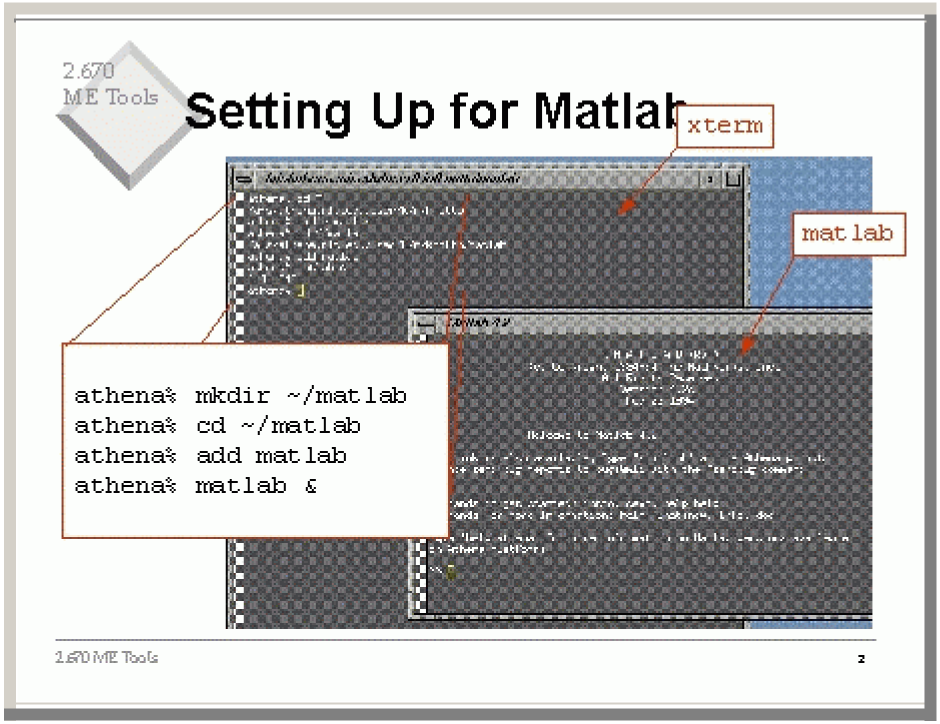
Lecture 1: Using Matlab



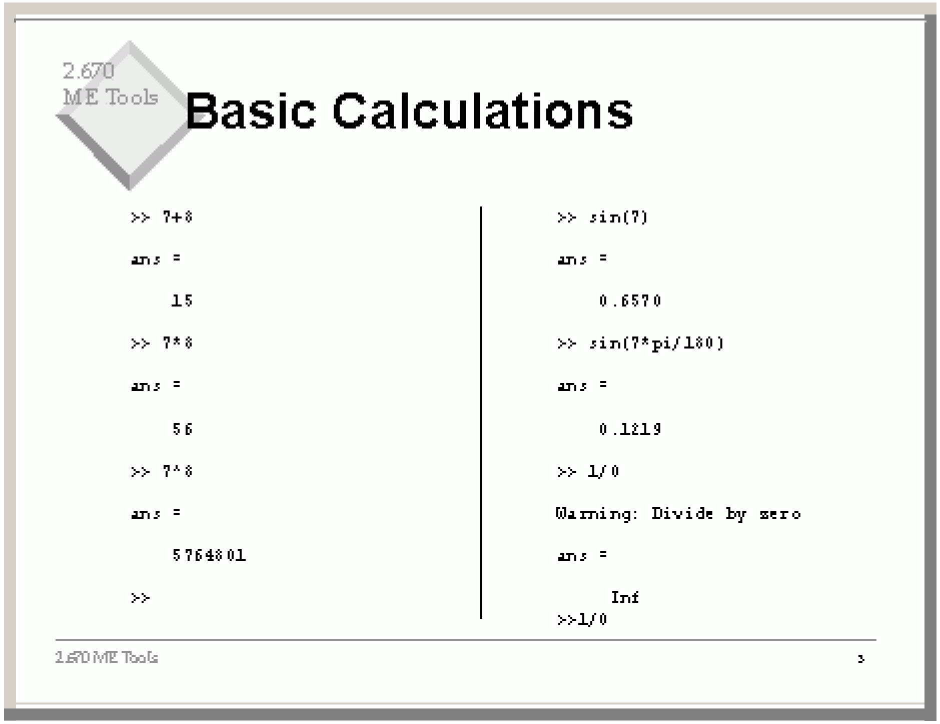
Notes:

This the first day of matlab, and presumes you have sat through the “Introduction to Matlab” training session.



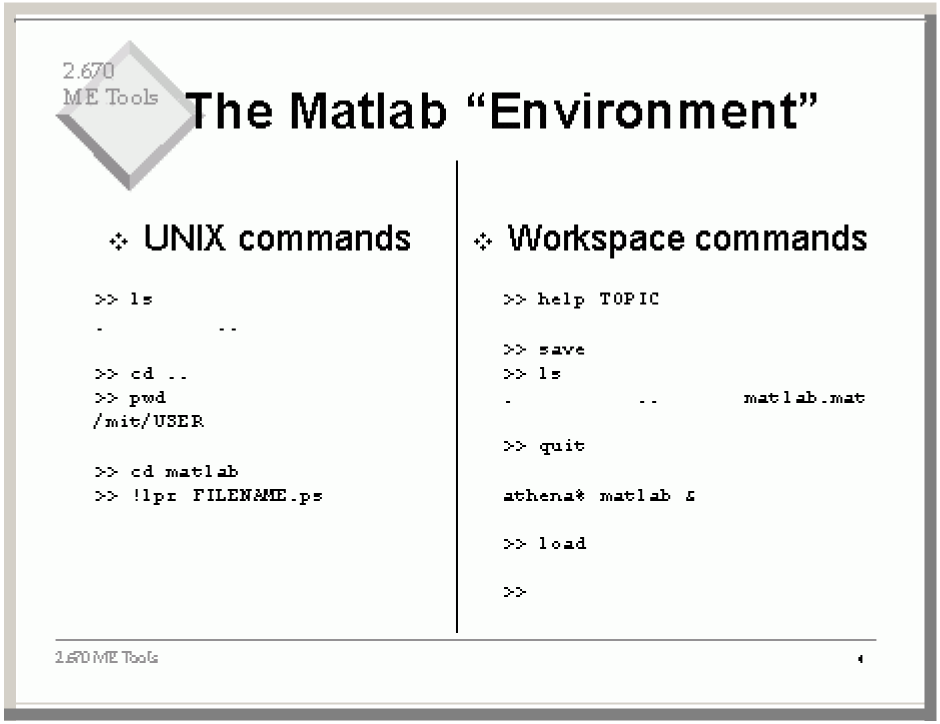
Notes:

First you need to set up your athena (server) environment to be matlab-happy.



Notes:

Matlab does basic calculations as you would expect.

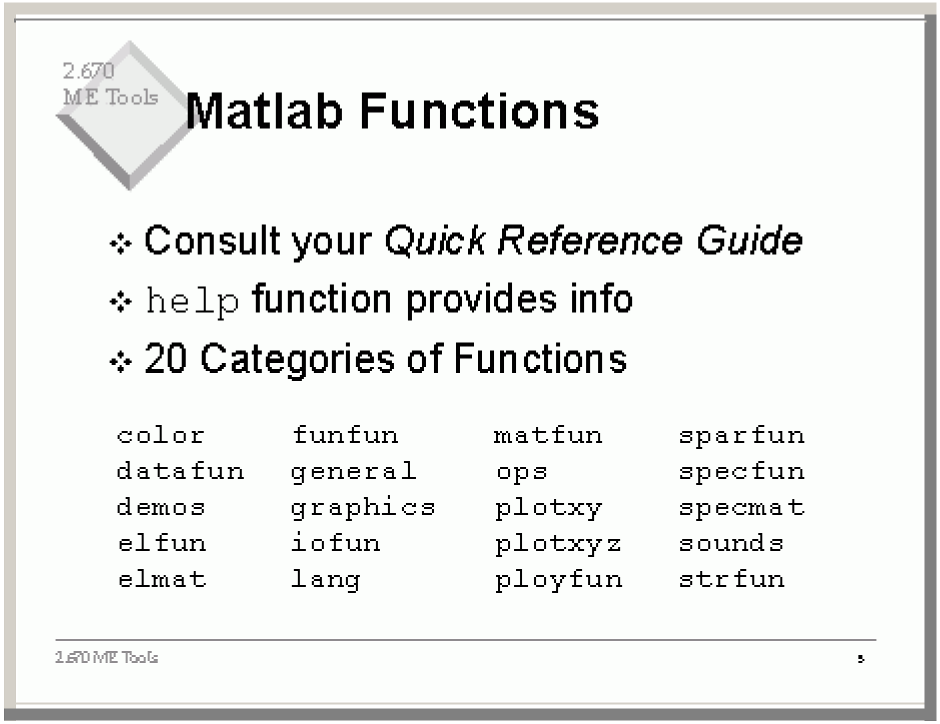


Notes:

In matlab, you can do all the UNIX commands.

There is an idea of the matlab workspace. As you define variables and formulas, they are stored, overwritten, expanded, deleted. But there is a state of the matlab environment at any point.

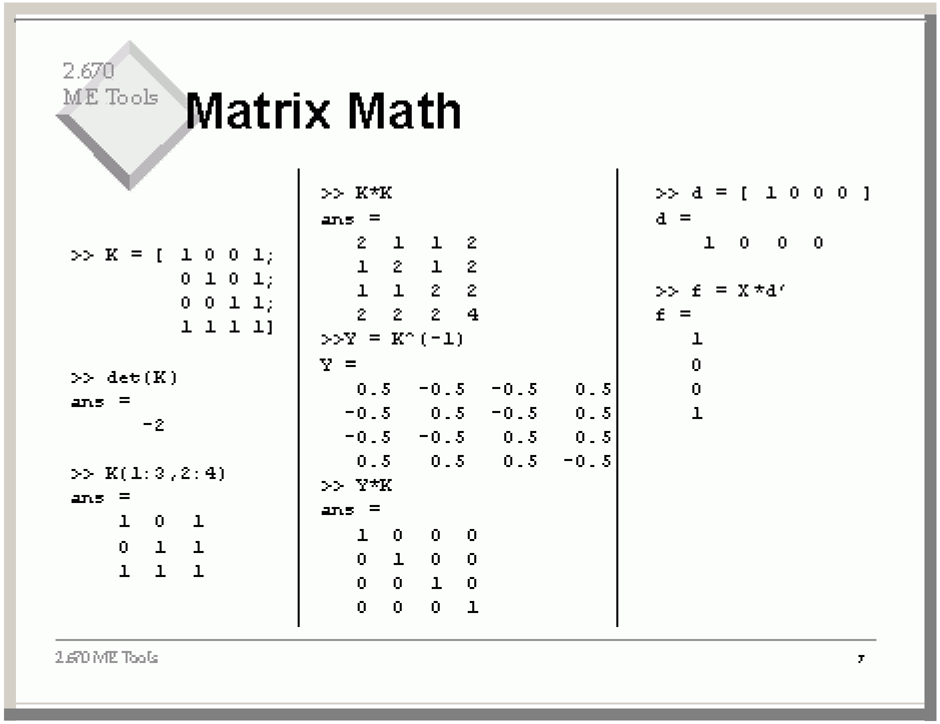
You may choose to leave matlab because of time pressure. You can store the state of matlab with a save command. Then quiot matlab, and leave. Later, after starting matlab, just enter load, and the file matlab.mat will be loaded, which restores the state of matlab to what it was when you did the save.



Notes:

There are many matlab functions..

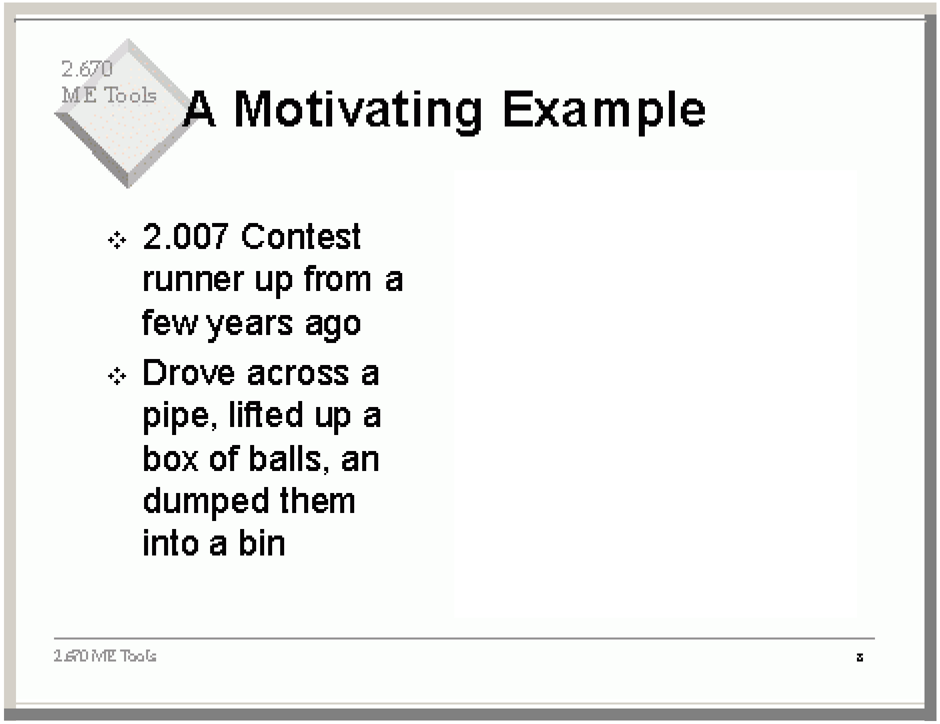
Matrix Math



Notes:

---

lets try some matrix math.



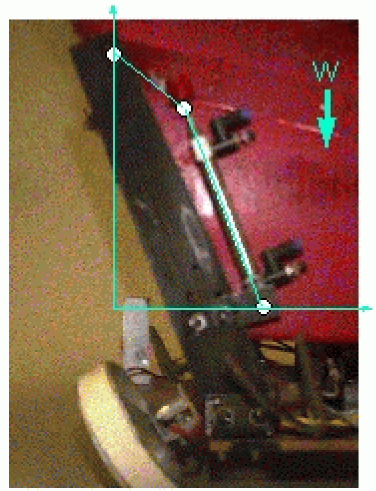
Notes:

Now lets see how we might use this.

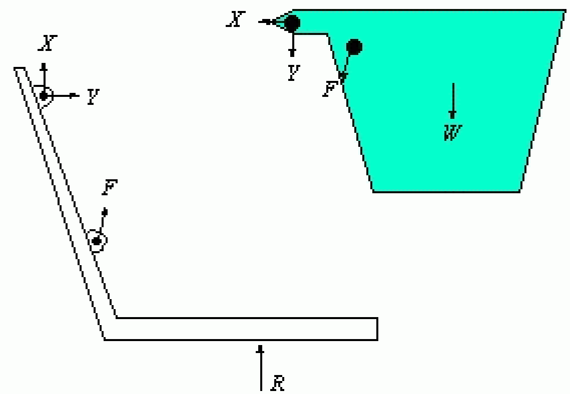
2.670
ME Tools

Examine the Lift Linkage

- How much weight in the bin can the air cylinders lift?
- An air cylinder can exert ~2 lbs force at 50 psig.



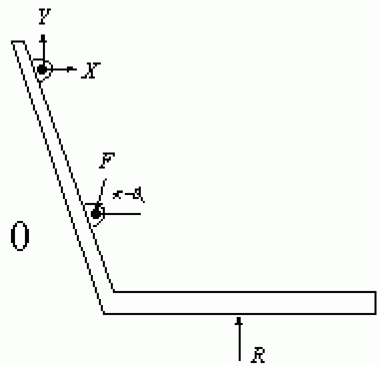
Model a System of Forces



Support Forces

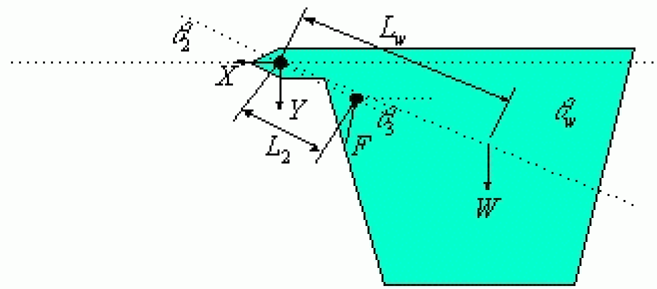
$$Y - F\sin(\pi - \theta_3) + R = 0$$

$$X - F\cos(\pi - \theta_3) = 0$$



2.670
ME Tools

Forces on the Bin



$$-Y + F\sin\theta_3 - W = 0$$

$$L_w \cos\theta_w W - L_2 \cos\theta_2 F \sin(\pi - \theta_3) - L_2 \sin\theta_2 F \cos(\pi - \theta_3) = 0$$

2.670
ME Tools

System of Equations

This defines a system of equations

$$\begin{bmatrix} 1 & 0 & \cos\theta_3 & 0 \\ 0 & -1 & \sin\theta_3 & 0 \\ 0 & 0 & L_2\sin(\theta_3-\theta_2) & 0 \\ 0 & 1 & -\sin\theta_3 & 1 \end{bmatrix} \begin{Bmatrix} X \\ Y \\ F \\ R \end{Bmatrix} = \begin{Bmatrix} 0 \\ W \\ L_w\cos\theta_w W \\ 0 \end{Bmatrix}$$

Or in matrix/vector notation

$$[B]\vec{F} = \vec{R}$$

Note that

$\theta_2 = 70^\circ$ $L_2 = 1.5"$

$\theta_3 = 110^\circ$

$\theta_w = 62^\circ$ $L_w = 6.5"$

2.670
ME Tools

What Force for 10lbs Weight?

```
>> rad2 = 70*pi/180;
>> rad3 = 110*pi/180;
>> radw = 62*pi/180;
>> L2 = 1.5;
>> Lw = 6.5;
>> W = 10;
>> Bmat = ...
  [ 1 0 cos(rad3) 0 ;...
    0 -1 sin(rad3) 0 ;...
    0 0 L2*sin(rad3-rad2) 0 ;...
    0 1 -sin(rad3) 1 ];
>> Rvec = ...
   [ 0 ; W ; Lw*cos(radw)*W ; 0 ];
>> Fvec = Bmat^(-1)*Rvec
```

```
Fvec =

   10.8247
   19.7406
   31.6493
   10.0000
```

So the two pistons need a combined force of 32 lbs to lift a bag of sugar in the bin.

What about other Weights?

- To try many weights, we need an m-file to call many times
- Call the m-file `force.m`

```
function Fvector = force(W)
% returns the reaction and
% piston forces for a given
% load, at the initial position.
r2 = 70*pi/180;
r3 = 110*pi/180;
rw = 60*pi/180;
L2 = 1.5;
Lw = 6.5;
Bmatrix = ...
  [ 1 0 cos(r3) 0 ;...
    0 -1 sin(r3) 0 ;...
    0 0 L2*sin(r3-r2) 0 ;...
    0 1 -sin(r3) 1 ];
Rvector = ...
 [ 0 ; W ; Lw*cos(rw)*W ; 0 ];
Fvector = Bmatrix^(-1)*Rvector;
```

Call the m-file

```
>> F=force(7)
F =

    7.5773
   13.8184
   22.1545
    7.0000
```

```
>> F=force(6)
F =

    6.4948
   11.8444
   18.9896
    6.0000
```

X force on the pin
Y force on the pin
F piston force
R force holding up structure

2.670
ME Tools

Expand the m-file

- Read into the m-file `rad2, rad3, radw` as a vector `ang`
- Try
  `ang(1) = 70*pi/180`
  `ang(2) = 70*pi/180`
  `ang(3) = 62*pi/180`
- What happened? Why?

```
function Fvector = force(W,ang)
% returns the reaction and
% piston forces for a given
% load W, at a given angular
% position ang = [a1 a2 a3].
r2 = ang(1);
r3 = ang(2);
rw = ang(3);
L2 = 1.5;
Lw = 6.5;
Bmatrix = ...
  [ 1 0 cos(r3) 0 ;...
    0 -1 sin(r3) 0 ;...
    0 0 L2*sin(r3-r2) 0 ;...
    0 1 -sin(r3) 1 ];
Rvector = ...
 [ 0 ; W ; Lw*cos(radw)*W ; 0 ];
Fvector = Bmatrix^(-1)*Rvector;
```

The Matrix Determinant

```
function Fvector = force(W,ang)
r2 = ang(1);
r3 = ang(2);
rw = ang(3);
L2 = 1.5;
Lw = 6.5;
Bmatrix = ...
  [ 1 0 cos(r3) 0 ;...
    0 -1 sin(r3) 0 ;...
    0 0 L2*sin(r3-r2) 0 ;...
    0 1 -sin(r3) 1 ];
disp(`The determinant is `);
disp(det(Bmatrix));
Rvector = ...
 [ 0 ; W ; Lw*cos(w)*W ; 0 ];
Fvector = Bmatrix^(-1)*Rvector;
```

```
>> force(10,[70*pi/180,...
   70*pi/180,60*pi/180])
The determinant is
     0
Warning: Matrix is singular to
working precision.

f =

   NaN
   NaN
   NaN
   NaN
```

- B is singular
- So even infinately large forces cannot support the bin at these angles.

Plotting Data in Matlab

2.670
ME Tools

Matlab plots data, not functions

- If you have a vector pair of data, plot it

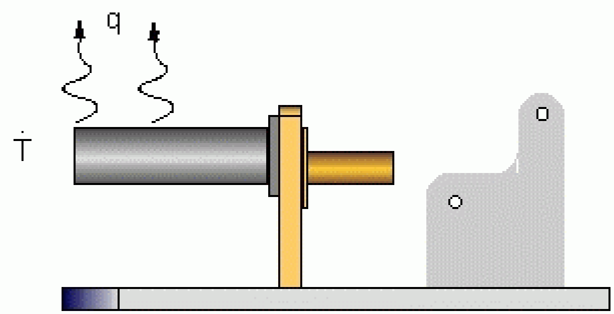
```
>> load('data.dat')
>> plot(data(:,1),...
        data(:,2))
```

- If you have a function to plot, first you must generate data that represents it

```
>> Xvec = 0:0.01:4;
>> Yvec = sin(Xvec);
>> plot(Xvec,Yvec)
```

A Motivating Example

- How long until you can grab the machine after blowing out the flame?



2.670
ME Tools

Cool Time Data

Measure Temperature at regular time intervals

t (min)	T (F)	t (min)	T (F)	t (min)	T (F)	t (min)	T (F)	t (min)	T (F)	t (min)	T (F)
0.17	653	4.50	243	8.67	154.4	12.83	118.4	17.00	96.8	21.17	87.8
0.33	619.4	4.67	240.8	8.83	152.6	13.00	114.8	17.17	96.8	21.33	87.8
0.50	590	4.83	233.6	9.00	152.6	13.17	113	17.33	96.8	21.50	87.8
0.67	559.4	5.00	230	9.17	149	13.33	111.2	17.50	95	21.67	86
0.83	532.4	5.17	224.6	9.33	147.2	13.50	113	17.67	95	21.83	86
1.00	505.4	5.33	222.8	9.50	145.4	13.67	113	17.83	95	22.00	86
1.17	483.8	5.50	219.2	9.67	143.6	13.83	111.2	18.00	93.2	22.17	86
1.33	460.4	5.67	213.8	9.83	141.8	14.00	109.4	18.17	93.2	22.33	84.2
1.50	438.8	5.83	210.2	10.00	140	14.17	109.4	18.33	93.2	22.50	84.2
1.67	422.6	6.00	206.6	10.17	140	14.33	107.6	18.50	93.2	22.67	84.2
1.83	406.4	6.17	199.4	10.33	138.2	14.50	107.6	18.67	91.4	22.83	84.2
2.00	390.2	6.33	197.6	10.50	134.6	14.67	105.8	18.83	93.2	23.00	84.2
2.17	368.6	6.50	195.8	10.67	136.4	14.83	105.8	19.00	91.4	23.17	84.2
2.33	361.4	6.67	190.4	10.83	134.6	15.00	104	19.17	91.4	23.33	82.4
2.50	350.6	6.83	186.8	11.00	131	15.17	104	19.33	91.4	23.50	82.4
2.67	338	7.00	181.4	11.17	131	15.33	104	19.50	91.4	23.67	82.4
2.83	327.2	7.17	176	11.33	125.6	15.50	102.2	19.67	91.4	23.83	82.4
3.00	316.4	7.33	174.2	11.50	125.6	15.67	102.2	19.83	89.6	24.00	82.4
3.17	307.4	7.50	174.2	11.67	125.6	15.83	100.4	20.00	89.6	24.17	82.4
3.33	302	7.67	172.4	11.83	125.6	16.00	100.4	20.17	89.6	24.33	82.4
3.50	294.8	7.83	168.8	12.00	123.8	16.17	100.4	20.33	87.8	24.50	82.4
3.67	284	8.00	165.2	12.17	123.8	16.33	100.4	20.50	86	24.67	80.6
3.83	275	8.17	163.4	12.33	122	16.50	98.6	20.67	87.8	24.83	80.6
4.00	266	8.33	158	12.50	122	16.67	98.6	20.83	87.8	25.00	80.6
4.17	260.6	8.50	156.2	12.67	120.2	16.83	96.8	21.00	87.8	25.17	80.6

Plotting Data

Copy the temperature data file from the 2.670 locker

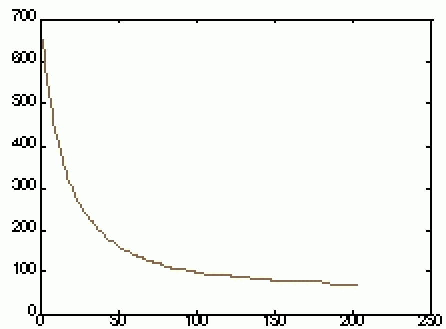
```
athena% cp /mit/2.670/Computers/matlab/temp.dat ~/matlab
```

Read data from a file into matlab

```
>> load(`temp.dat´);
>> tvec=temp(:,1);
>> TmpDatavec=temp(:,2);
```

Now plot the measured data

```
>> plot(tvec,TmpDatavec)
```



Hmmm....

Looks exponential.

2.670
ME Tools

Why an Exponential Look?

The plot looks like an exponential decay...

Convection Cooling ⇨ $q = hA\left(T(t) - T_\infty\right)$

Lumped Mass ⇨ $-q = mc_p \dfrac{dT}{dt}$

$$\frac{dT}{dt} = \frac{-hA}{mc_p}\left(T(t) - T_\infty\right)$$

2.670
ME Tools

Convective Cooling

Transform variable $D = T - T_\infty$ $\dot{D} = \dot{T}$

So $\dfrac{dD}{dt} = \dfrac{-hA}{mc_p} D \Rightarrow \dfrac{-hA}{mc_p} dt = \dfrac{dD}{D}$

Integrate $\displaystyle\int_0^t \frac{-hA}{mc_p} dt = \int_{D_0}^{D} \frac{dD}{D} \Rightarrow \frac{-hAt}{mc_p} = \ln\left(\frac{T - T_\infty}{T_0 - T_\infty}\right)$

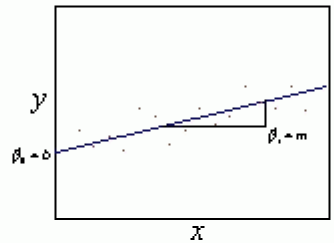
So

$$\boxed{T = T_\infty + \left(T_0 - T_\infty\right) e^{\left(\frac{-hA}{mc_p}\right)t}}$$

Curve Fitting

- Matlab only fits polynomials to data….

```
>> polyfit(Xvec,Yvec,1)
```



fits a least squares best line

$$y_{predicted} = \beta_0 + \beta_1 x$$

to a dataset

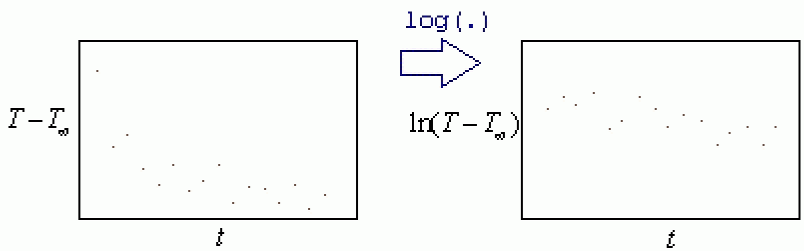
$$\left[\vec{x}, \vec{y}_{measured}\right]$$

2.670
ME Tools

Transform Data to a Linear Form

- Our data is exponential, not linear

  So take the `log(.)` of the data



$T - T_\infty \sim K_0 e^{K_1 t}$

$\ln(T - T_\infty) \sim \beta_0 + \beta_1 t$

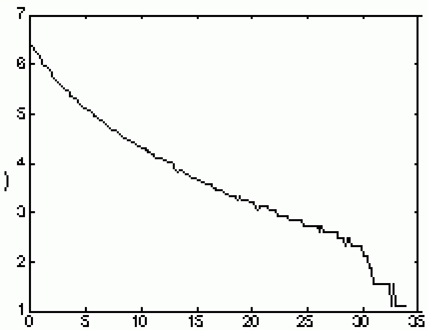
2.670
ME Tools

Plot the Ln Data

Just to convince ourselves, lets plot the `log(.)` data.

Ambient Temperature is 65°F

```
>> Tinf = 65;
>> DeltaDatavec = ...
          TmpDatavec - Tinf;
>> lnDeltaDatavec = ...
          log(DeltaDatavec);
>> plot(tvec,lnDeltaDatavec)
```



Is it linear?.....

Find the Least Squares Fit

Find the coefficients that least squares best fit a line (a linear polynomial):

```
>> beta=polyfit(tvec,lnDeltaDatavec,1)
beta =

    -0.1281    5.7895
```

A_1 ↑ $\qquad$ A_0 ↑

Which tells us $\ln\left(T(t) - 65\right) = 5.8 - .13t$

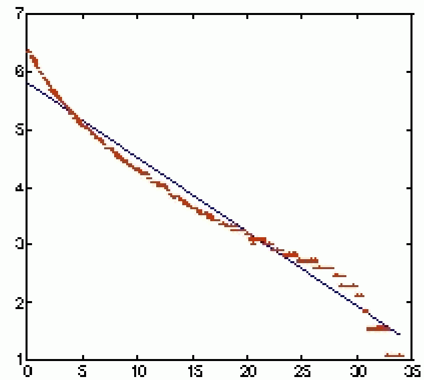
or $T(t) = 65 + 330e^{-.13t}$

Do you believe this least squares best fit model?

Plot both Data and Line

Lets check by visualizing the data.

```
>> Fitvec = ...
   beta(1)*tvec ...
   + beta(2);
>> plot(tvec,...
   lnDeltaDatavec,...
   'r.',...
   tvec,...
   Fitvec,'b')
```

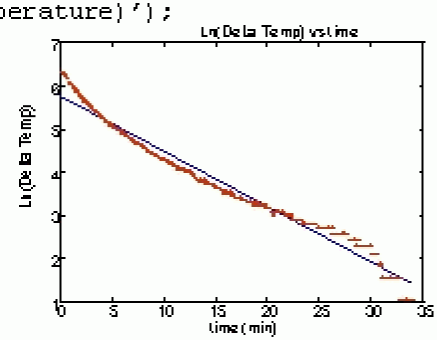


Do you believe the model?

Formatting Plots

Add titles, axis labels, and legends with plot formatting commands

```
>> title('Ln(Delta Temp) vs. time');
>> xlabel('time (min)');
>> ylabel('Ln(Delta Temperature)');
```

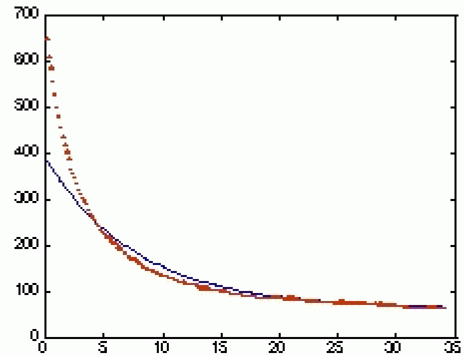


Plot the Exponential Equation

The least squares best fit expoential model to the data is then

$$T = 65 + 330e^{-.13t}$$

How long until we can grab our engine?



Do you believe
- the model?
- the time prediction?