Matlab Review

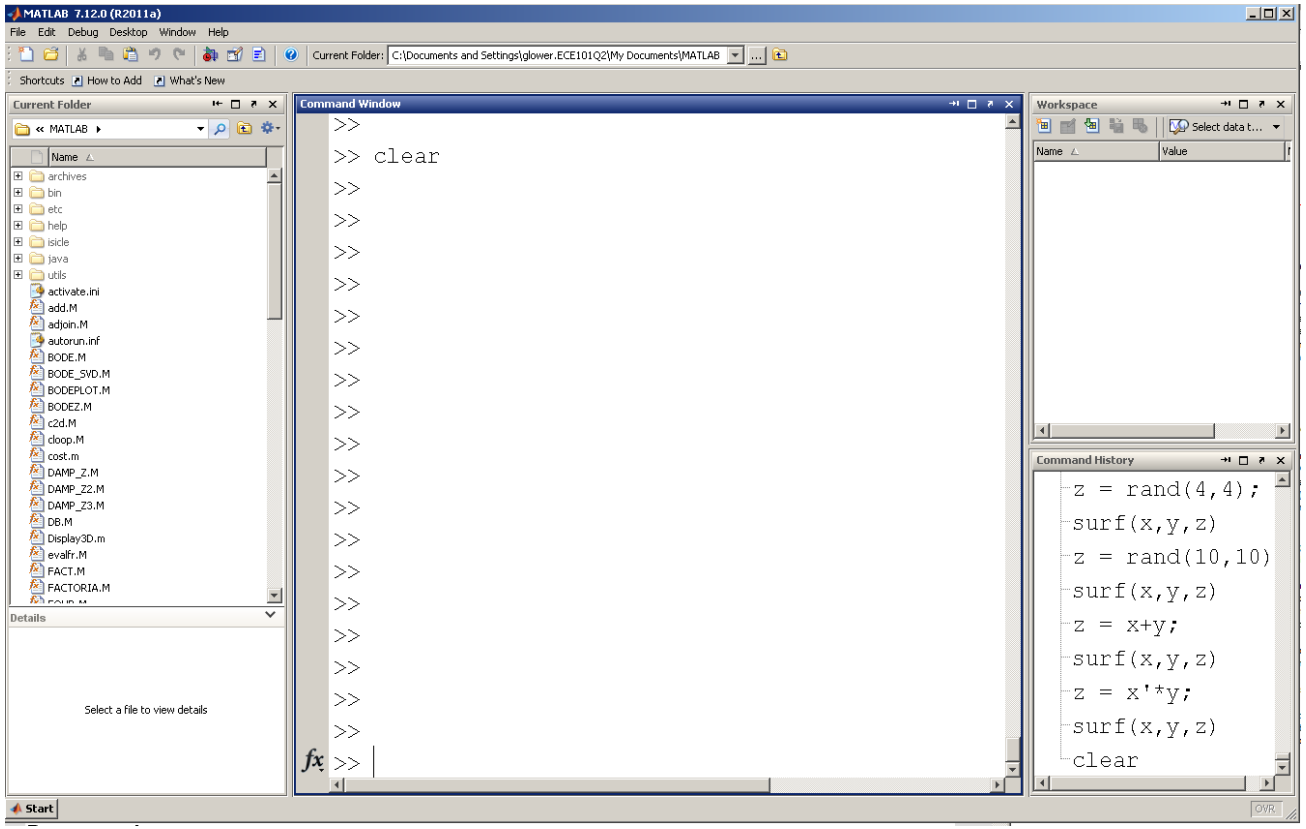
Lecture #1
ECE 761: Robotics

Class taught at North Dakota State University
Department of Electrical and Computer Engineering

Please visit www.BisonAcademy.com for corresponding lecture notes, homework sets, and solutions.

Becoming familiar with MATLAB

- The console
- The editor
- The graphics windows
- The help menu
- Saving your data (diary)



Matlab can be used like a calculator: it adds, subtracts, multiplies, and divides

```
x = 17/3
    5.6667

y = (3+4)*5
    35
```

Some special numbers are pre-defined in Matlab

```
e = exp(1)
    2.7183

pi
    3.1416

i
        0 + 1.0000i

j
        0 + 1.0000i
```

You can choose to or not to display the results of an operation

```
x = 2*pi
    6.2832

x = 2*pi;
```

You can change how numbers are displayed

```
format short
pi

    3.1416

format long
pi

   3.141592653589793

format longe
pi^30

    8.212893304027486e+014
```

At its heart, Matlab is a matrix language: it is designed to to matrix operations.

To input a matrix, the following symbols are used

```
[ start of matrix
] end of matrix
, next element
; next row
```

Example:

```
A = [1,2,3]
    1     2     3

B = [1,2,3;4,5,6]
     1     2     3
     4     5     6

C = A'
     1
     2
     3

D = zeros(1,3)
     0     0     0
```

Random Numbers: Uniform distribution from (0, 1)

```
rand(2,4)

    0.8147    0.1270    0.6324    0.2785
    0.9058    0.9134    0.0975    0.5469
```

Normal distrubution: N(0,1)

```
randn(2,4)

    3.5784   -1.3499    0.7254    0.7147
    2.7694    3.0349   -0.0631   -0.2050
```

for loops:

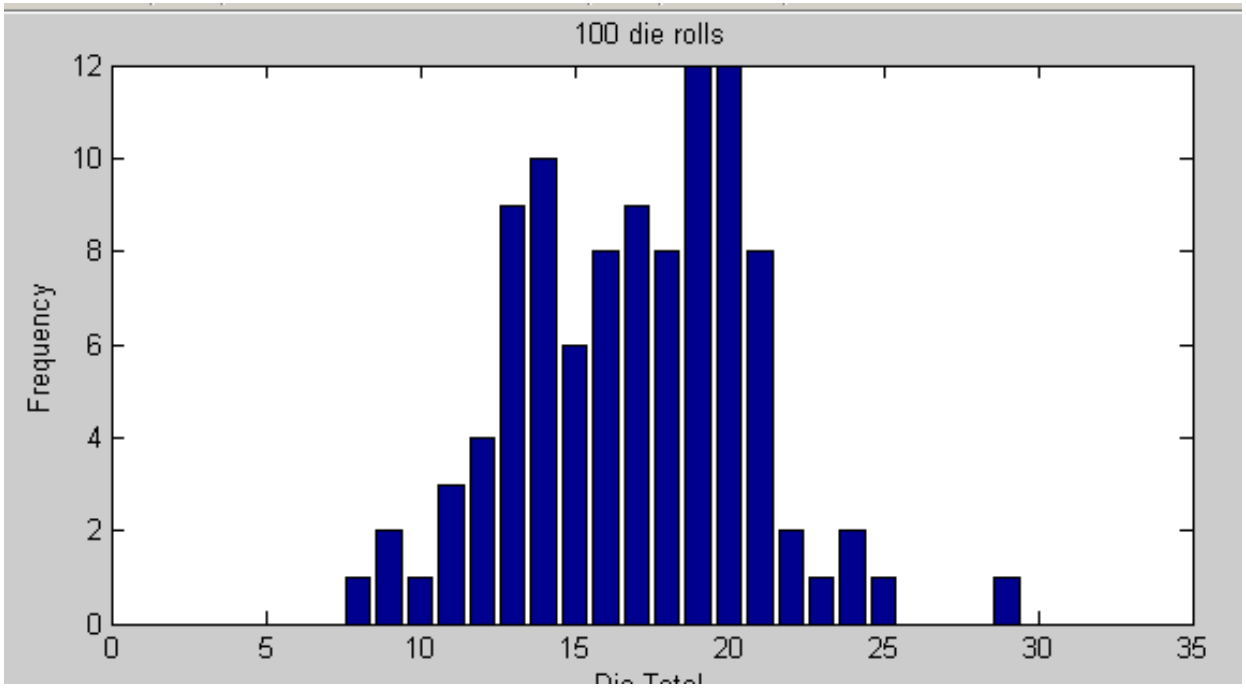
```
x = zeros(1,5);
for i=1:5
   x(i) = i*i;
   end
x

x =  1     4     9    16    25
```

Example: Roll five 6-sided dice

```
>> x = [1:5]

     1     2     3     4     5

>> rand(1,5)

    0.7298    0.8908    0.9823    0.7690    0.5814

>> dice = ceil( 6*rand(1,5) )

     3     6     2     4     4
dice = ceil( 6*rand(1,5) )

     1     4     3     1     3

sum( ceil( 6*rand(1,5) ) )

     8
sum( ceil( 6*rand(1,5) ) )

    16
```

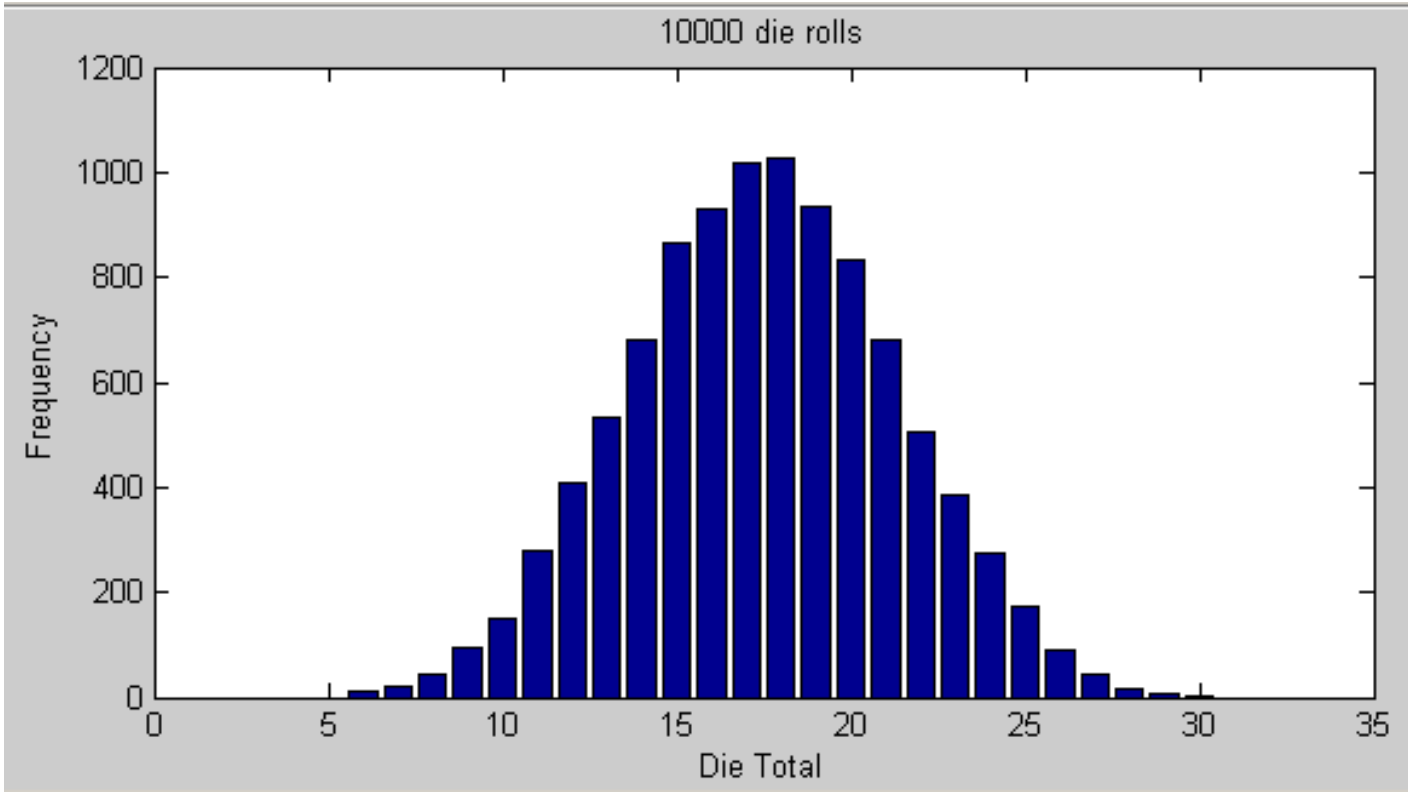
Roll 5d6 one-hundred times and record how many rolls you get for each total:

```
X = zeros(30,1);
for i=1:100
   D = sum( ceil( 6*rand(1,5) ) );
   X(D) = X(D) + 1;
   end
bar(X)
xlabel('Die Total');
ylabel('Frequency');
title('100 die rolls')
```



Roll 5d6 10,000 times and record the frequency of each outcome:

```
X = zeros(30,1);
for i=1:10000
    D = sum( ceil( 6*rand(1,5) ) );
    X(D) = X(D) + 1;
    end
bar(X)
```



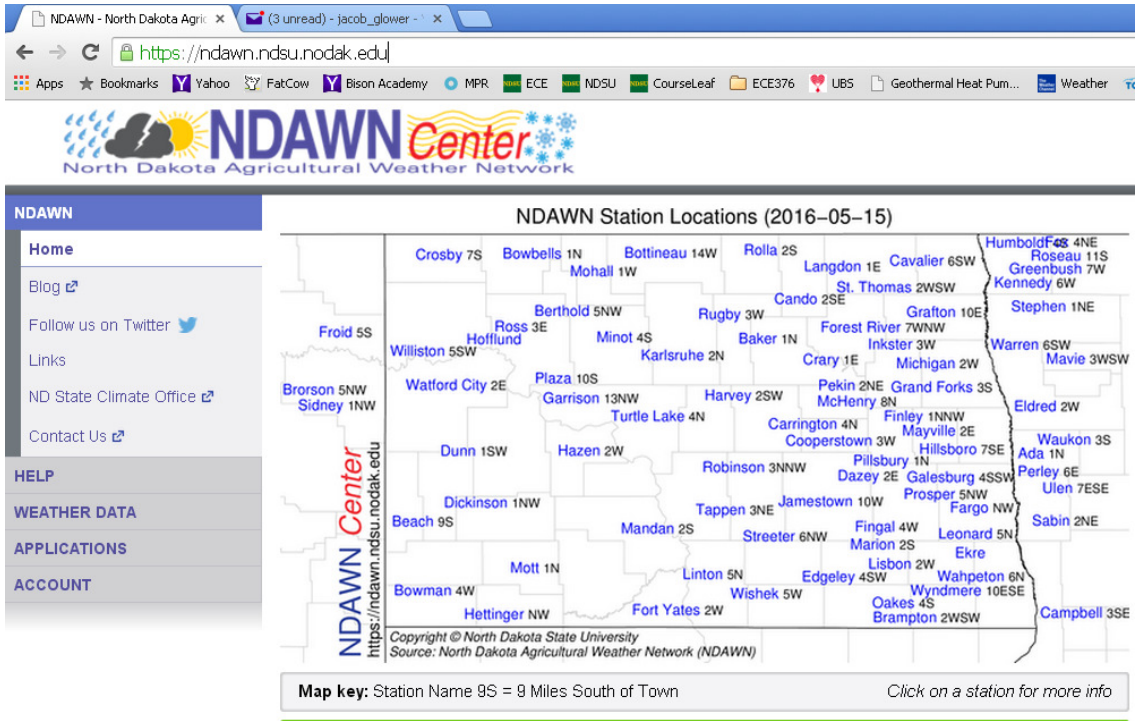
Numerical Integration:

The simplest (and least accurate) is Euler integration

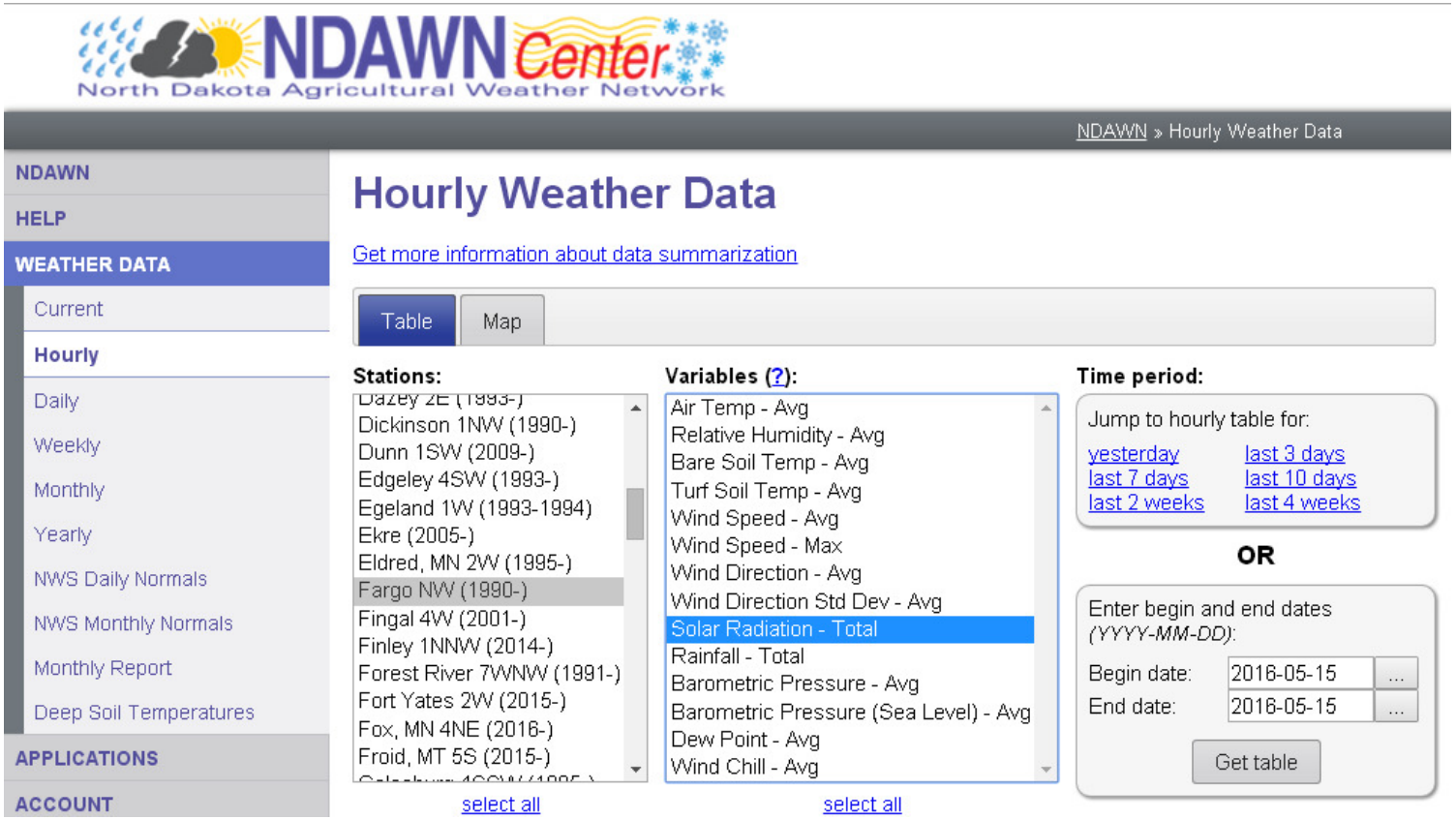
Area = Width * height

Example: Determine how much energy a 1.5m2 solar panel will generate in Fargo, ND over the past two weeks. Assume the efficiency of the solar panel is 20%

Solution: Get solar data from NDAWN: https://ndawn.ndsu.nodak.edu/

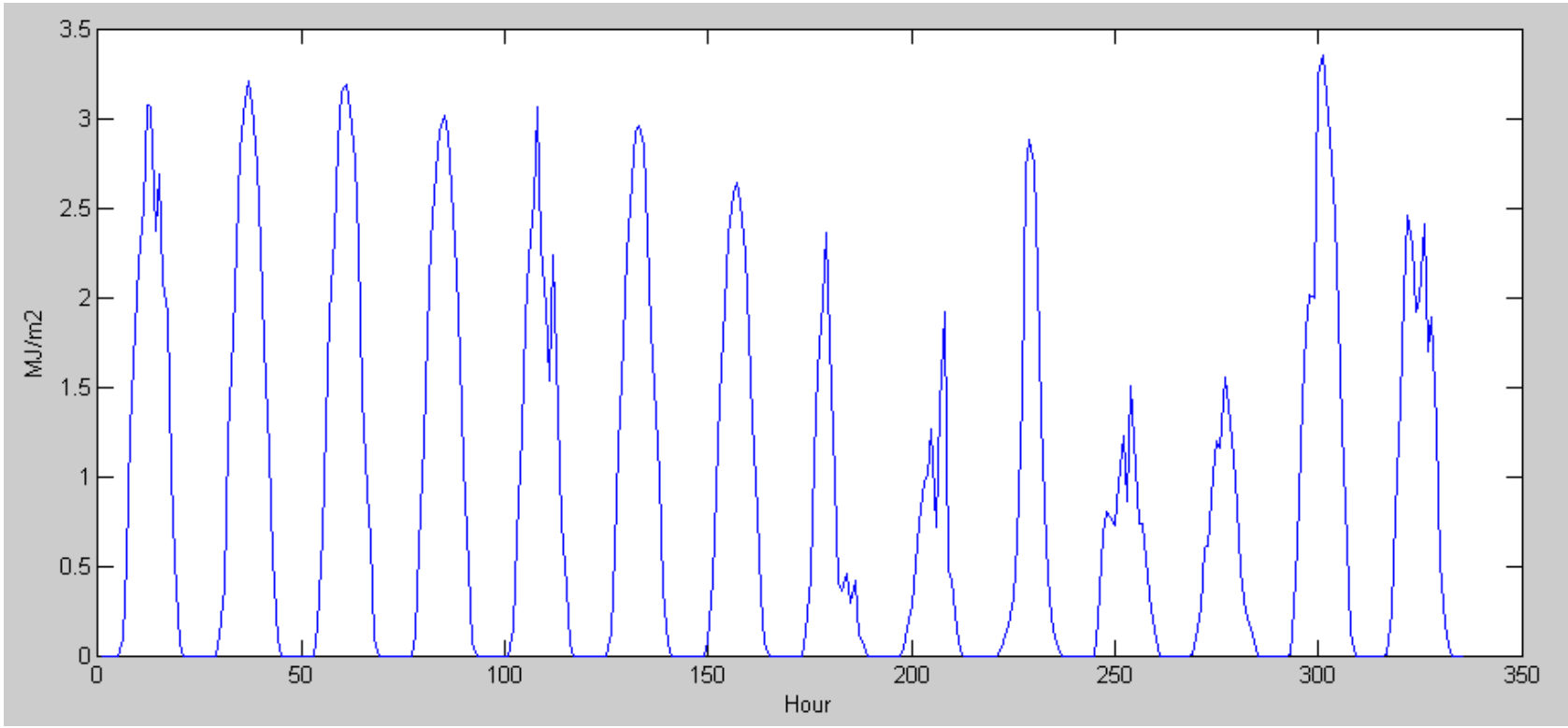


Select Weather Data - Hourly - Fargo - Solar Total



Export to a CVS file and copy the data to the clip board. From Matlab

```
Sun = [
      paste in the data
    ];

h = [1:length(Sun)]';
plot(h,Sun);
```



This is hourly data. To convert to Joules, integrate

- height =data * 1,000,000 (MJ total over an hour)
- width = 1 hour
- Area = Width * height = Joules

```
MJ = sum(Sun)

MJ =  280.2130
```

To convert that to kWh

$$1MJ = 0.2778\ kWh$$

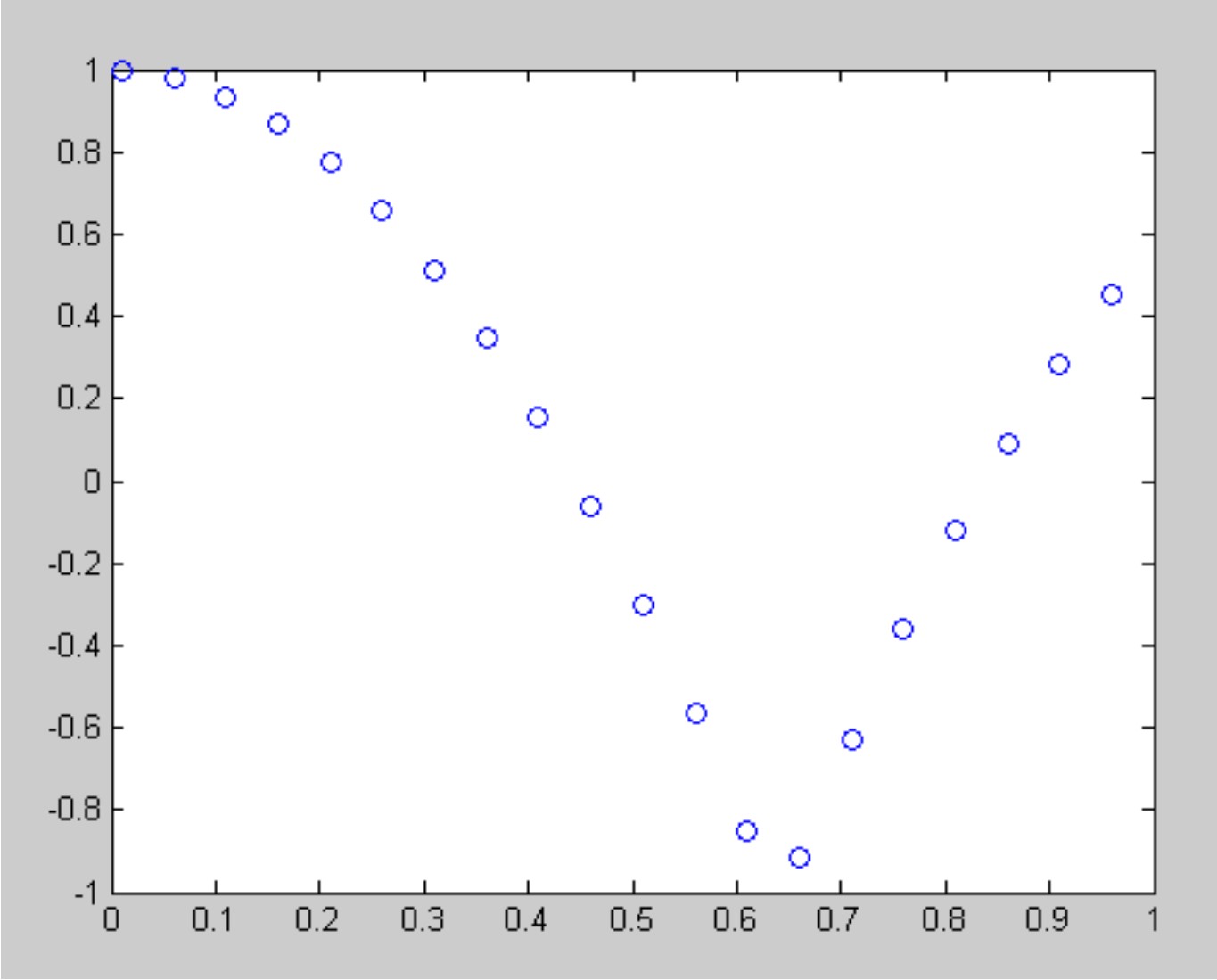
```
kWh = MJ * 0.2778

kWh =   77.8432
```

At 20% efficiency, a solar panel would generate 15.5kWh over this 2 week span. This is worth about $1.55

Bouncing Ball

Matlab is also surprisingly good at animation. For example, simulate a bouncing ball

```
x = 0;
y = 1;

dx = 1;
dy = 0;

ddx = 0;
ddy = 0;

dt = 0.01;

for i=1:1000
   ddy = -9.8;

   dx = dx + ddx*dt;
   dy = dy + ddy*dt;

   x = x + dx*dt;
   y = y + dy*dt;

   if (x > 1)
      dx = -abs(dx);
      end
   if (x < -1)
      dx = abs(dx);
      end
   if (y < -1)
      dy = abs(dy);
      end

   hold off
   plot([-1,1],[-1,1],'.');
   hold on
   plot(x,y,'o')
   pause(0.01);
   end
```



Bouncing Ball

Matlab Commands

Analysis

- sqrt(x) square root of x
- log(x) log base e
- log10(x) log base 10
- exp(x) e^x
- exp10(x) 10^x
- abs(x) |x|
- round(x) round to the nearest integer
- floor(x) round down (integer value of x)
- ceil(x) round up to the next integer
- real(x) real part of a complex number
- imag(x) imaginary part of a complex number
- abs(x) absolute value of x, magnitude of a complex number
- angle(x) angle of a complex number (answer in radians)
- unwrap(x) remove the discontinuity at pi (180 degrees) for a vector of angles

Polynomials

- poly(x)
- roots(x)
- conv(x,y)

Trig Functions

- sin(x) sin(x) where x is in radians
- cos(x) cos()
- tan(x) tan()
- asin(x) arcsin(x)
- acos(x) arccos(x)
- atan(x) arctan(x)
- atan2(y,x) angle to a point (x,y)

Probability and Statistics

- factorial(x) (x-1)!
- gamma(x) x!
- rand(n,m) create an nxm matrix of random numbers between 0 and 1
- randn(n,m) create an nxm matrix of random numbers with a normal distribution
- sum(x) sum the columns of x
- prod(x) multiply the columns of x
- sort(x) sort the columns of x from smallest to largest
- length(x) return the dimensions of x
- mean(x) mean (average) of the columns of x
- std() standard deviation of the columns of x

Display Functions

- plot(x) plot x vs sample number
- plot(x,y) plot x vs. y
- semilogx(x,y) log(x) vs y
- semilogy(x,y) x vs log(y)
- loglog(x,y) log(x) vs log(y)
- mesh(x) 3d plot where the height is the value at x(a,b)
- contour(x) contour plot
- bar(x,y) draw a bar graph
- xlabel('time') label the x axis with the word 'time'
- ylabel() label the y axis
- title() put a title on the plot
- grid() draw the grid lines

Useful Commands

- hold on — don't erase the current graph
- hold off — do erase the current graph
- diary — create a text file to save whatever goes to the screen
- linepace(a, b, n) — create a 1xn array starting at a, increment by b
- logspace(a,b,n) — create a 1xn array starting at 10^a going to 10^b, spaced logarithmically
- subplot() — create several plots on the same screen
- disp('hello') — display the message *hello*

Utilities

- format — set the display format
- zeros(n,m) — create an nxm matrix of zeros
- eye(n,m) — create an nxm matrix with ones on the diagonal
- ones(n,m) — create an nxm matrix of ones
- help — help using different functions
- pause(x) — pause x seconds (can be a fraction). Show the graph as well
- clock — the present time
- etime — the difference between to times
- tic — start a stopwatch
- toc — the number of seconds since tic