Homework #0

Table of Contents

First Last
Math 365, Spring 2016

Introduction

The purpose of this homework is to become familar with the Matlab publish command and to see how to turn in homework problems.

```
function hmwk0()
```

Every assignment will include a function like this which will serve as a "main" function for your homework.

```
% You will then call all of your homework problem "functions" like
 this.
hmwk_problem(@prob1,'prob1');
hmwk_problem(@prob2,'prob2');
hmwk_problem(@prob3,'prob3');
hmwk_problem(@prob4,'prob4');
hmwk_problem(@prob5,'prob5');
hmwk_problem(@prob6,'prob6');
hmwk_problem(@prob7,'prob7');
hmwk_problem(@prob8,'prob8');


end

function hmwk_problem(prob,msg)
```

This function should be included in every assignment

```
try
    prob()
    fprintf('%s : Success!\n',msg);
catch me
    fprintf('%s : Something went wrong.\n',msg);
    fprintf('%s\n',me.message);
end
fprintf('\n');
```

```
end
```

Problem #1 : Surface area of a torus

In this problem, we compute the surface area of a torus whose inner radius is 3.21 and whose outer radius is 3.56. The result is saved to the file 'torus.out'

```
function prob1()

%  Your work goes here

end
```

```
prob1 : Success!
```

Problem #2 : Smartphones

```
function prob2()

% Your work goes here

end
```

```
prob2 : Success!
```

Problem #3 : Vectorizing calculations

```
function prob3()
```

Matlab makes it very easy to manipulate vectors and arrays. Many commands that would normally require a "for" loop can be "vectorized", creating code that is generally faster and more compact than non-vectorized equivalents.

```
% Create an array of 11 equally spaced points in [-1,1].  Note that
% there are many ways to create an array in Matlab.  This is the
% simplest way!
x = [1, 2, 3, 4, 5];     % x_i = i,   i = 1,2,3,4,5.
y = [7,6,5,4,3];     % y_j = 5-j, j = -2,-1,0,1,2.

% We can use a "loop" to create a vector z = x + y
z = zeros(size(x));
for i = 1:5,
    z(i) = x(i) + y(i);
end

% Vectorize this loop!
z = x + y;

% Store your z array to a file.
write_file(z,'z.out');

end
```

```
prob3 : Success!
```

Problem #4 : Plot the graph of a function

```
function prob4()
```

Create function handles for two functions and construct a third composite function.

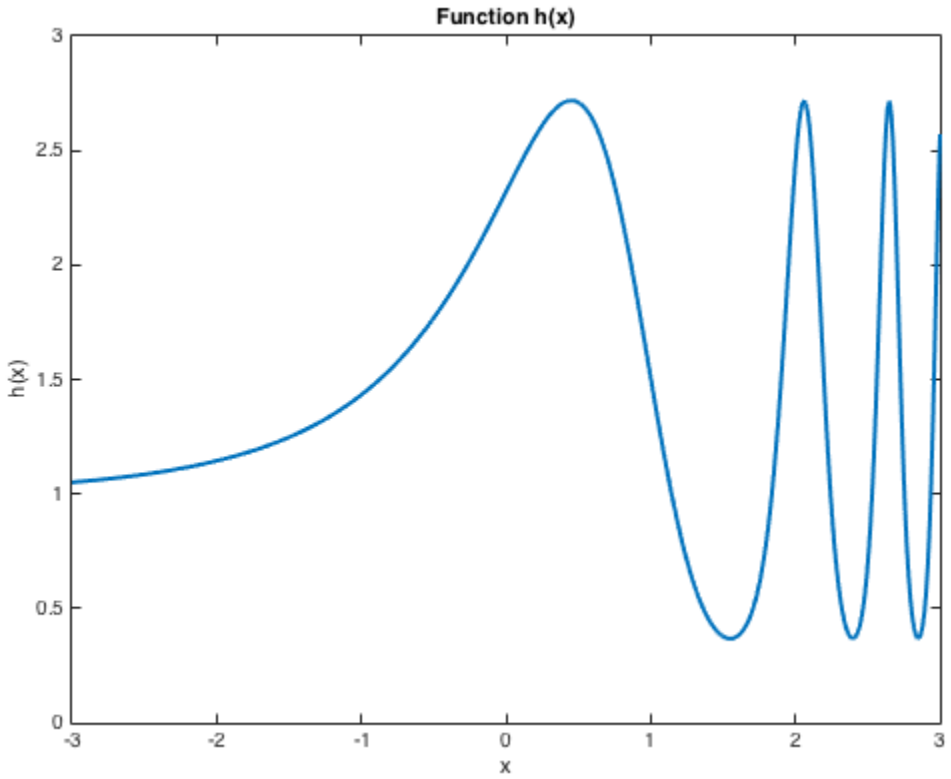
```
% Anonymous function handles
f = @(x) sin(x);
g = @(x) exp(x);
h = @(x) g(f(g(x)));

% Construct  a vector of equally spaced points
x = linspace(-3,3,500);

% Plot the results
plot(x,h(x),'linewidth',2);
title('Function h(x)');
xlabel('x');
ylabel('h(x)');

end
```

```
prob4 : Success!
```



Problem #5 : Continued fraction approx. to pi

```
function prob5()
```

Use a continued fraction to approximate π. Make sure that your code doesn't run off of the edge of the page.

Publish allows you to insert mathematical expressions (although they don't show up so well in PDF).

$$\pi \approx d_1 + \cfrac{1}{d_2 + \cfrac{1}{d_3 + \cfrac{1}{\ddots}}}$$

```
d = [3, 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, ...
    14, 2, 1, 1, 2, 2, 2, 2, 1, 84];

x = d(1) + 1/(d(2) + 1/(d(3) + 1/(d(4) + 1/...
    (d(5) + 1/(d(6) + 1/(d(7) + 1/(d(8) + 1/...
    (d(9) + 1/(d(10) + 1/(d(11) + 1/(d(12) + 1/...
    (d(13) + 1/(d(14) + 1/(d(15)))))))))))));

pi_approx = x;

fprintf('%15s %24.16f\n','pi (approx)',pi_approx);
fprintf('%15s %24.16f\n','pi (exact)',pi);
fprintf('%15s %24.4e\n','Error',abs(pi-pi_approx));

end
```

```
    pi (approx)       3.1415926535897931
     pi (exact)       3.1415926535897931
          Error                0.0000e+00
prob5 : Success!
```

Problem #6 - Continued fraction approx. to e

```
function prob6()

% Your work goes here

end
```

```
prob6 : Success!
```

Problem #7 : Loading data from a file

```
function prob7()
```

Load data from a file and report the mean and standard deviation.

```
h = load('heights.dat');

fprintf('%12s %12d\n','Count',length(h));
```

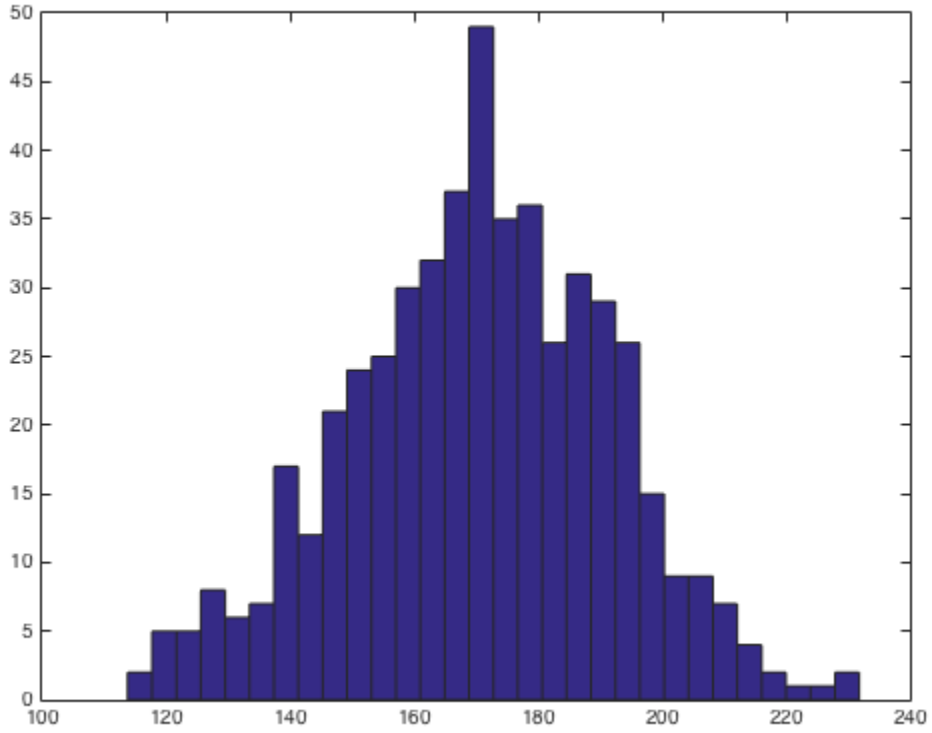
```
fprintf('%12s %12.2f\n','Min',min(h));

% Add remaining computations here

% Create a histogram of the data.
hist(h,30);

end
```

```
       Count          513
         Min       113.81
prob7 : Success!
```



Problem #8 : How to succeed in Math 365

```
function prob8()
```

Publish allows you to create lists, use different font styles, and include preformatted code

How to succeed in Math 365

- *Always* start your homework early
- *Don't* spend too much time googling for answers
- *Read* the homework tips!

Steps for getting help on homework problems.

1. Read the Matlab tutorials available on the course website
2. Read lecture notes and demo codes on the online website.
3. Use Matlab online "help" system for help on Matlab commands.
4. Read the Course textbook
5. Email the professor for help, if you can't find answers in the above.
6. Do not spend too much time with Prof. Google or Dr. YouTube. This is likely going to be a waste of time! Spend more time thinking about what you have learned in class, and debugging your own code.

Include sample code that you don't want run by "formatting" the code like this. Use exactly three spaces between the percent sign and your code.

```
curly = 4*pi;
larry = sin(curly);
moe = tan(curly + larry);
```

There are lots of helpful hints for publishing by issuing the command

```
>> doc publishing markup
```

```
end
```

```
prob8 : Success!
```

Published with MATLAB® R2015a