Table of Contents


```
%    _      _      _      _      _      _      _      _      _      _
%   (c).-.(c)    (c).-.(c)    (c).-.(c)    (c).-.(c)    (c).-.(c)    (c).-.(c)
%    / ._. \      / ._. \      / ._. \      / ._. \      / ._. \      / ._. \
%  __\( Y )/__  __\( Y )/__  __\( Y )/__  __\( Y )/__  __\( Y )/__  __\( Y )/__
% (_.-/'-'\-._)(_.-/'-'\-._)(_.-/'-'\-._)(_.-/'-'\-._)(_.-/'-'\-._)(_.-/'-'\-._)
%    || O ||      || O ||      || O ||      || O ||      || O ||      || O ||
%  _.' `-' '._  _.' `-' '._  _.' `-' '._  _.' `-' '._  _.' `-' '._  _.' `-' '._
% (.-./`-'\.-.)(.-./`-'\.-.)(.-./`-'\.-.)(.-./`-'\.-.)(.-./`-'\.-.)(.-./`-'\.-.)
%  `-'     `-'  `-'     `-'  `-'     `-'  `-'     `-'  `-'     `-'  `-'     `-'

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Author:  Matt Fletcher                                                      %
%    Class:  ENG101, Fall, 2017                                                 %
%  Helpers:  None                                                               %
%                                                                               %
%  Program:  ENG101 Homework 4                                                  %
% Due Date:  25 Sep 2017                                                        %
%                                                                               %
% Language:  MatLab                                                            %
%      IDE:  MatLab R2017a                                                      %
%                                                                               %
% Purpose:   Perform requested tasks from Homework 4                           %
%                                                                               %
%                                                                               %
%                                                                               %
%   "Undocumented features": None.                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Clear console and all variables.
clear;
clc;
```

Problem 1

```
clear;
% Form Velocity Vector & Angle Vectors
v = (10:2:20)';          % m/s
theta = 50:10:80;     % degrees
g=9.8;                 %m/s^2
```

```
% Preallocate h matrix
h = zeros(length(theta),length(v));
% Two For Loops
for indx = 1:length(theta)
    for jndx = 1:length(v)
       h=v.^2.*sind(theta)/(2*g);
    end
end

fprintf('Solution to problem 1')
%Set array to a table.
array2table(h)
```

```
Solution to problem 1
ans =

  6×4 table

          h1                h2                h3                h4
    _______________    _______________    _______________    _______________

    3.90839001591315    4.41849695808387    4.79435010605055    5.0245293521031
    5.62808162291494    6.36263561964077     6.9038641527128    7.23532226702847
    7.66044443118978    8.66025403784439    9.39692620785908    9.84807753012208
    10.0054784407377    11.3113522126947    12.2735362714894    12.8627951413839
    12.6631836515586    14.3159301441917    15.5336943436038    16.2794751008141
    15.6335600636526    17.6739878323355    19.1774004242022    20.0981174084124
```

Problem 2

Problem 2 with For Loop

```
%Format numbers as long
format long
clear;


fprintf('\n\n\n Leibnitz using for loop \n\n')
%Set number of trials
num_trials=10.^(2:6);

%Start counter
idx=0;
```

```
%Print desired value
fprintf('The desired value is %f. \n', pi/4)


%Initialize sum
sum_leib=0;

%For loop to calculate value of summation.
for idx = 0:num_trials(end)
    %Defined by equation
    leib = ((-1)^idx)/(2*idx+1);

    %Use recursion to add sum for k=idx to the current sum.
    sum_leib=sum_leib+leib;

    %Increase index by 1
    idx=idx+1;

    %Print out results at given values.
    if (idx==100||idx==1000||idx==10000||idx==1000000)
        fprintf('When k equals %d, the summation equals %f \n',idx,
 sum_leib)
    end
end


%While loop to calculate required number of iterations until
%the error is less than the desired amount.
clear;
fprintf('\n\n\n Leibnitz using a while loop \n\n')
%Set error amount
%TODO make error smaller after debugging
error=1e-7;

%Set desired value equal to pi/4
leib_actual=pi/4;

%Set index value
idx=0;

%Set initial value of leib
sum_leib=0;

%While loop
while abs(sum_leib-leib_actual)>error
    leib = ((-1)^idx)/(2*idx+1);
    sum_leib=sum_leib+leib;
    idx=idx+1;
end
%Print result
fprintf('Solution \n')
fprintf('To get within %1.1E of pi/4, %d iterations are required.\n ',
 error, idx)
```

Leibnitz using for loop

The desired value is 0.785398.
When k equals 100, the summation equals 0.782898
When k equals 1000, the summation equals 0.785148
When k equals 10000, the summation equals 0.785373
When k equals 1000000, the summation equals 0.785398

Leibnitz using a while loop

Solution
To get within 1.0E-07 of pi/4, 2500001 iterations are required.

Problem 3

```
%TODO create vectorizing version
clear;

%Leibnitz equation
%sum from 0 to k of (-1)^k/(2k+1)

%Set n to 1000000
n=1000000;

%Form  odd series
oddSeries=1:2:n;

%Form signs
signs=0:(length(oddSeries)-1);

%Form alternating signs
altsigns=(-1).^signs;

%Create pi vector
pivector=4*sum(altsigns./oddSeries);

%Printing pretty crap
fprintf("\n\n\nProblem 3, finding Leibnitz sum with 1000000 iterations
 \n \n \n ")
%Form summation
fprintf ('Sum= %6f\n',pivector)
```

Problem 3, finding Leibnitz sum with 1000000 iterations

```
Sum= 3.141591
```

Published with MATLAB® R2017a