EGR 103L – Fall 2017

Structured Programming I

Ian Hanus (ih52)
Lab Section 1B, Tuesday 8:30-11:20 AM
1 October 2017

I understand and have adhered to all the tenets of the Duke Community Standard in completing every part of this assignment. I understand that a violation of any part of the Standard on any part of this assignment can result in failure of this assignment, failure of this course, and/or suspension from Duke University.

Contents

List of Figures

1 Chapra Problem 2.10

The code and the graph are in the appendices.

2 Chapra Problem 2.22

The equations for the butterfly curve[1, p. 52] are:

$$x = \sin(t)(e^{\cos(t)} - 2\cos(4t) - \sin^5(\frac{t}{12}))$$
$$y = \cos(t)(e^{\cos(t)} - 2\cos(4t) - \sin^5(\frac{t}{12}))$$

These equations generate a butterfly curve because of the sinusoidal nature of the sine and cosine graphs. The two long phrases in parentheses in the equation above match, so they simply determine the magnitude of the $\sin(t)$ and $\cos(t)$ graphs that distinguish x from y. The changing comparison of sin and cos at the different points make the butterfly appear.

3 Palm Problem 3.12

The equation for the length of the fence is:

$$\text{Length} = \text{Area} - \frac{1}{2}(\frac{\text{Width}}{\sqrt{2}})^2$$

Grading diary:

```
Trial  1: Not enough input arguments.
Trial  2: Not enough input arguments.
Trial  3: Length: 7.50e-01
Trial  4: Not enough input arguments.
Trial  5: Incorrect entry or entries.
Trial  6: Not enough input arguments.
Trial  7: Incorrect entry or entries.
Trial  8: Impossible Area
Trial  9: Length: 2.00e+00
Trial 10: Incorrect entry or entries.
Trial 11: Incorrect entry or entries.
Trial 12: Incorrect entry or entries.
```

4 Pick a card!

```
52 out of 52 correct
Grade: 10.0 out of 10
```

5 What card is that?

```
52 out of 52 correct
Grade: 10.0 out of 10
```

6 Is that a three of a kind?

```
You found 52 and there are 52 TOK
22100 out of 22100 correct assessments
Grade: 20.0 out of 20
Test of ~TOK with printout:
Your Cards: 6S JS 5H
Nothing...

ans =

     0

Test of ~TOK with no printout:

ans =

     0

Test of ~TOK with no printout argument:

ans =

     0

Test of TOK with printout:
Your Cards: 6S 6S 6S
Three of a Kind!

ans =

     4

Test of TOK with no printout:

ans =

     4

Test of TOK with no printout argument:

ans =

     4

```

A Codes

A.1 RunChapra02p10.m

```
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [ih52]
%% Initialize the workspace
clear; format short e
%% Define anonymous equation - fix this
yfun = @(alpha, beta, xi) beta.*exp(-alpha.*xi).*sin(beta*xi).*(0.012.*xi.^4-0.15.*xi.^3+0.075.*xi.^2+2.
%% Make plots
x = 0:pi/40:pi/2;
figure(1); clf
plot(x, yfun(2, 5, x), 'ms--', 'LineWidth', 2,   'MarkerSize',  16, 'MarkerFaceColor', 'y')
hold on
plot(x, yfun(4,5,x), 'ok-', 'LineWidth', 1.5, 'MarkerSize', 12, 'MarkerFaceColor', 'g')
plot(x, yfun(2,10,x), 'dr-.', 'LineWidth', 1, 'MarkerSize', 8, 'MarkerfaceColor', 'w', 'MarkerEdgeColor'
hold off
grid on
legend('(2,5)', '(4,5)', '(2,10)', 'location', 'best')
print -depsc Chapra02p10Plot1
```

A.2 RunButterfly.m

```
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [ih52]
%% Initialize workspace
clear; format short e
%% Define variable
t = linspace(0,24*pi,10000);
[x,y] = Butterfly(t);
%% Plot butterfly
figure(1); clf
subplot(2,1,1)
plot(t,x,'g-',t,y,'m-')
hold on
subplot(2,1,2)
plot(x,y,'r-')
axis square
hold off
%% Print butterfly
print -depsc RunButterfly
```

A.3 Butterfly.m

```
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [ih52]
function [x,y] = Butterfly(t)
x = sin(t).*(exp(cos(t))-2.*cos(4.*t)-(sin(t./12)).^5);
y = cos(t).*(exp(cos(t))-2.*cos(4.*t)-(sin(t./12)).^5);
```

A.4 FenceLength.m

```
 % I have adhered to all the tenets of the
 % Duke Community Standard in creating this code.
 % Signed: [ih52]
%% Create function
function [L] = FenceLength(W,A);
%% Check for conditions
if W<=0
    error('Incorrect entry or entries.')
elseif A<=0
    error('Incorrect entry or entries.')
elseif numel(W)~=1
    error('Incorrect entry or entries.')
elseif numel(A)~=1
    error('Incorrect entry or entries.')
elseif nargin ~= 2
    error('Incorrect entry or entries.')
elseif A.^2 < W.^2./4
    error('Impossible Area')
%% Calculate Length
else
    L = (A - .5*(W./sqrt(2)).^2)./W;
end
% Total Length = W + 2((A - .5*(W./sqrt(2)).^2)./W)+2(sqrt(w^2/2))
```

A.5 BuildCard.m

```
 % I have adhered to all the tenets of the
 % Duke Community Standard in creating this code.
 % Signed: [ih52]
function CardString = BuildCard(CardNum)

CardVal = mod(CardNum-1,13)+1;
CardSuit = ceil(CardNum./13);
%% Get card value - this works already!
Vals  = 'A23456789XJQK';
CardString(1) = Vals(CardVal);
%% Get card suit - you have to do all this!
Suits = 'SHDC';
CardString(2) = Suits(CardSuit);
fprintf('%s%s\n','CardString(1)','Cardstring(2)')
end
```

A.6 CardInfo.m

```
1   % I have adhered to all the tenets of the
2   % Duke Community Standard in creating this code.
3   % Signed: [ih52]
4   function [Val, Suit] = CardInfo(C)
5   %% Get value from the first character
6   if C(1)=='A'
7       Val=1;
8   elseif C(1)=='X'
9       Val=10;
10  elseif C(1)=='J'
11      Val=11;
12  elseif C(1)=='Q'
13      Val=12;
14  elseif C(1)=='K'
15      Val=13;
16  % add code for jack, queen, and king
17  else % the card must be a 2 through a 9 if we're here
18      Val= str2num(C(1));
19  end
20
21  %% If the above tree doesn't assign a value, make it 0 by default
22  if isempty(Val)
23      Val=0;
24  end
25
26  %% Get suit fromt the second character
27  switch C(2)
28      case 'S'
29          Suit=1;
30      case 'H'
31          Suit=2;
32      case 'D'
33          Suit=3;
34      case'C'
35          Suit=4;
36      otherwise
37          Suit=0;
38  end
```

A.7 CheckTOK.m

```
 % I have adhered to all the tenets of the
 % Duke Community Standard in creating this code.
 % Signed: [ih52]
function [Val] = CheckTOK(C,p)
if nargin == 1
    p = 0;
end
if p==1
    fprintf('Your Cards: %s %s %s\n', C(1,:),C(2,:),C(3,:));
end
if p==1
    if C(1)~=C(2)|C(1)~=C(3);
        fprintf('Nothing...\n');
        Val = 0;
    else
        fprintf('Three of a Kind!\n');
        Val = 4;
    end
else
    if C(1)==C(2) & C(1)==C(3)
        Val = 4;
    else
        Val = 0;
    end
end
```

B Figures

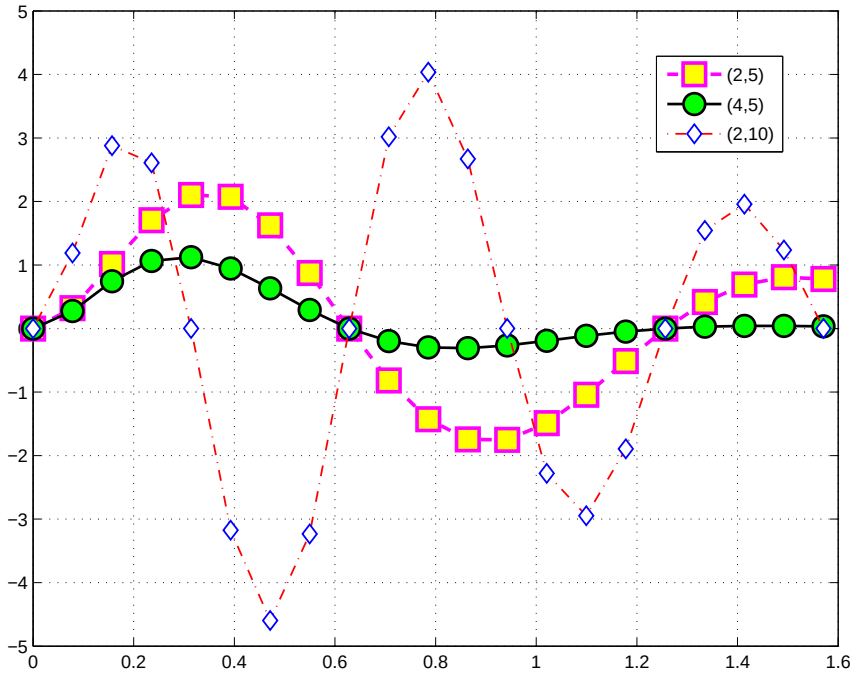

Figure 1: Plot of Chapra Problem 2.10

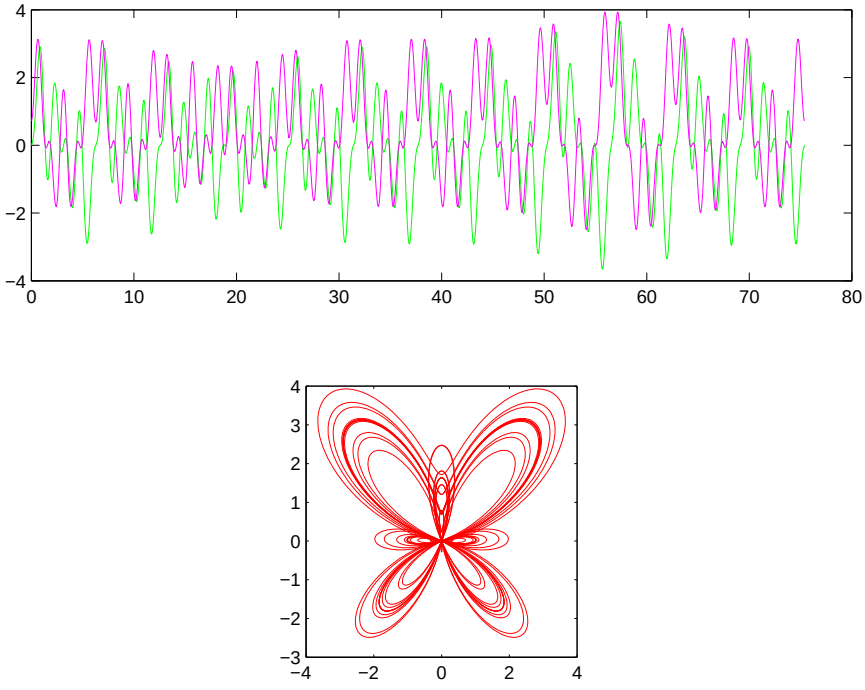

Figure 2: Plot of Butterfly Equations

References

[1] Chapra, Steven C., *Applied Numerical Methods with MATLAB for Engineering and Scientists*. McGraw-Hill, New York, 4th Edition, 2018.

[2] Palm, William J., *Introduction to MATLAB for Engineers*. McGraw-Hill, New York, 3rd Edition, 2011.