ECOR 2606 Midterm Fall 2011 B & C

Sample Solutions and Marking Scheme

Authorized memoranda: Closed book. Supplied formula sheets. Authorized calculators.
Answer both questions in the booklet provided.   Marks as shown.
Show your work.
Put your name and student number on this sheet and include it in the book.

1. You own a factory which manufactures bicycles. Every month, you sell 100 bikes at $700 each. You discover that if you lower the price by $10 each, you sell two more bikes. The total revenue therefore is *R(x) = (700 - 10x) (100 + 2x)*, where *x* is the number of $10 discounts you make. You want to find the price in order to maximize your monthly revenue.

   a) Write this as a root finding problem and then write the Matlab code to find the optimal price and output the results using *fprintf*. (4 marks)

   Calculations to get derivative:
   (700-10x)(100+2x) = 0
   70000 – 1000x + 1400x -20x^2 = 0
   -20x^2 +400x +70000 = 0
   Derivative is:
   -40x + 400 (set this to 0 to find the maximum of the original function)
   Matlab:

```
r = @(x) -20*x^2 +400*x +70000    % or (700-10*x)*(100+2*x)); .* optional
rp = @(x) -40*x  +400
a = fzero(rp, [-10 15]) % (comma between values is fine; range must include 10;
         % or alternatively: a = roots([-40 400]))
fprintf('Optimal price is $%6.2f, giving revenue of $%8.2f.\n', 700-10*a, f(a))
         % (6.2, 8.2 optional)
```

   (1 mark for function; 1 mark for derivative; 1 mark for fzero/roots; 1 mark for fprintf)

   b) Write the Matlab code to find the maximum value of *R(x)* directly and output the results using *fprintf*. (3 marks)

```
r = @(x) (700-10*x)*(100+2*x) % (or -20*x^2 +400*x+70000) reuse from above
g = @(x) –r(x) % (or could write out the negative of the above)
a = fminbnd(g, -10, 15)   % range must include 10
fprintf('Optimal price is $%6.2f, giving revenue of $%8.2f.\n', 700-10*a, f(a))
         % (6.2, 8.2 optional)
```

   (1 mark for function; 1 mark for fminbnd; 1 mark for fprintf)

c) Perform three iterations on *R(x)* using the Golden Section algorithm, by completing the table below. Start with x_L = -10 and x_u = 15. (5 marks)

x_L	x_2	x_1	x_U	$R(x_2)$	$R(x_1)$	**Emax**
-10	-0.4509	5.4509	15	-69815.595	-71586.105	12.50
-0.4509	5.4509	9.0983	15	-71586.105	-71983.739	7.7254
5.4509	9.0983	11.3525	15	-71983.739	-71963.412	4.7746

Deduct 1 mark if students have not shown their work (on the exam or in the booklet, somewhere).

(2 marks for the x's, 2 marks for R(x)'s, 1 mark for Emax's, -1 mark if work not shown.)

d) What is your best guess for optimal price and the resultant revenue? (3 marks)

$(x_L+x_U)/2$ = (5.4509+15)/2 = 10.2255
or
$(x_L+x_1)/2$ as x_1 will be the next x_U = (5.4509+11.3525)/2 = 8.4017
(2 marks for optimal price.)

Giving revenue of: $71998.98
or
$71948.91

(2 marks for optimal price; 1 mark for resultant revenue.)

(15 marks)

2. A simplified model of the suspension of a car consists of a mass *m*, a spring with stiffness *k*, and a damper with damping coefficient, *c*. A bumpy road can be modeled by a sinusoidal up-and-down motion of the wheel $y = Y \sin(\omega t)$. From the solution of the equation of motion for this model, the steady-state up-and-down motion of the car is given by $x = X \sin(\omega t - \varphi)$. The ratio between amplitude *X* and amplitude *Y* is given by:

$$X / Y = \sqrt{\left(\frac{m\,c\omega^2}{k(k - m\omega^2) + (\omega\,c)^2}\right)} \qquad \text{....(1)}$$

Assuming m = 2000 kg, k = 500,000 N/m and $c = 28 \times 10^3$ Ns/m, we want to determine the frequency $\boldsymbol{\omega}$ (omega) for which $X / Y = 0.2$.

a) Write Matlab code which will solve for the required $\boldsymbol{\omega}$ using *fzero* and produce nicely formatted results. (5 marks)

```
m = 2000; k = 500000; c = 28e3; XDY = 0.2;
g = @(omega) sqrt(m*c*omega.^2/(k*(k-m*omega.^2) + (omega*c).^2)) - XDY
    % (dot operators optional)
a = fzero(g, [0 20])  % range should include 12.43
fprintf('The required frequency is %5.3f rad/sec.\n',a)
    % (5.3 optional)
```

(1 mark for constants; 2 marks for function; 1 mark for fzero; 1 mark for fprintf)

(continued on next page)

b) Write the Matlab code which will produce the required ***ω*** using *roots* and then output the **one** required root. (7 marks)

Calculations:
Square both sides and multiply by denominator:
(XDY^2)*(k*(k-m*omega^2)+(omega*c)^2)) = m*c*omega^2
Move RHS to left and rewrite in quadratic form:
(XDY^2*(-m*k+c^2)-m*c)*omega^2 + XDY^2*k^2 = 0
Matlab:

```
m = 2000; k = 500000; c = 28e3; XDY = 0.2; % as before
poly = [XDY^2*(-m*k+c^2)-m*c 0 XDY^2*k^2];
omegavec = roots(poly);
% need to find the positive one
if (omegavec(1)>0)
  omegart = omegavec(1);
else
  omegart = omegavec(2);
end
fprintf('The required frequency is %5.3f rad/sec.\n',omegart)
    % (5.3 optional – continued on next page)
% or (replacing the above from "if" on)
for i =1:length(omegavec);
   if omegavec(i)  > 0
      fprintf('The required frequency is %f rad/s.\n', omegavec(i))
   end
end
```

(3 marks for polynomial; 1 mark for roots; 2 marks for determining positive root; 1 mark for fprintf.)

c) I am doing a bisection search and after 11 iterations my E_{MAX} is 0.0004883. What was the original search width and how many more iterations would I need to do to get $E_{MAX} < 0.000001$? (3 marks)

Current E_{MAX}*2^11 = 0.0004883*2^11 = 1.00004 (or 1).
(Note: E_{MAX} after the first iteration is 0.0004883*2^10, so original search width is twice that, thus the 2^11.)
Original search width was 1 (or 1.00004).

To get < 0.000001, we need to find n such that 0.0004883 < 2^n*0.000001.
In other words, 2^n > 0.0004883/0.000001 = 488.3, or 2^n = 512, n = 9.
Can also take log2(488.3) = 8.9316 and thus 9.
Thus, 9 more iterations are needed to get $E_{MAX} < 0.000001$. (Total of 20 iterations.)

(1 mark for original search width; 2 marks for number of additional iterations.)

(15 marks)