## Practice Questions for Exam 1 on Wed Feb 15 6-8 PM in Williams 103

**1) A polynomial P is given by**

****

**Write a MATLAB program to calculate and print the roots of the polynomial P**

**(the values of x where the polynomial is zero). Call the function newton to**

**calculate the roots. Scan the x axis from 0 to 5 in steps of .1 to look for**

**the roots. Use 1e-7 as the accuracy factor.**

**Do not write the function newton.**

**The output of this program should look like this:**

**x = 1.38462**

**x = 2.57143**

**x = 3.37500**

**x = 4.00000**

**Answer**

**accuracy = 1e-7;**

**f = @(x) 728\*x^4 - 8249\*x^3 + 33660\*x^2 - 57996\*x + 34992;**

**fp = @(x) 2912\*x^3 - 24747\*x^2 + 67320\*x - 57996;**

**stepsize = .1;**

**for(grid\_point = 0:stepsize:5)**

**left\_end\_point = grid\_point;**

**right\_end\_point = grid\_point + stepsize;**

**function\_left = f(left\_end\_point);**

**function\_right = f(right\_end\_point);**

**if(function\_left \* function\_right < 0 || function\_right == 0)**

**guess = (left\_end\_point + right\_end\_point)/2;**

**x = newton(f, fp, guess, accuracy);**

**fprintf('x = %.5f\n', x);**

**end**

**end**

**2) A long range rifle is fired with muzzle velocity u toward a target. The firing angle  that will permit the rifle to hit the target is given by the equation**



**where h = 70 m, d = 5500 m, and g = 9.81 m/s2 .**

**Write a MATLAB program as follows:**

**1) u will go from 240 m/s to 300 m/s in steps of 30 m/s .**

**2) For each value of u, call the MATLAB function fzero to calculate both**

**values of  that will enable the bullet to hit the target. Scan the **

**axis from 0º to 90º in steps of 1º to look for solutions for  .**

**The output of this program should look like this:**

**u=240 theta=36.20945**

**u=240 theta=54.51973**

**u=270 theta=24.78090**

**u=270 theta=65.94828**

**u=300 theta=19.23816**

**u=300 theta=71.49102**

**Answer**

**h = 70;**

**d=5500;**

**g = 9.81;**

**for(u = 240:30:300)**

**f = @(theta) u\*cos(theta)\*(u\*sin(theta)/g+sqrt(u^2\*sin(theta)^2/g^2-2\*h/g)) - d;**

**stepsize = 1\*pi/180;**

**for(grid\_point = 0:stepsize:90\*pi/180)**

**left\_end\_point = grid\_point;**

**right\_end\_point = grid\_point + stepsize;**

**function\_left = f(left\_end\_point);**

**function\_right = f(right\_end\_point);**

**if(function\_left \* function\_right < 0 || function\_right == 0)**

**guess = (left\_end\_point + right\_end\_point)/2;**

**theta = fzero(f, guess);**

**fprintf('u=%d theta=%.5f\n', u, theta\*180/pi);**

**end**

**end**

**fprintf('\n');**

**end**

## 3)

**R2**

**R2y**

**2**

**R2x**

## R1

**R1y**

**R3y**

**R3y**

**R3**

**R3x**

**R2x**

**R1x**

**3**

**1**

**R4**

**In the four bar linkage shown above, R1=3.24, R2=3.78, R3=2.56, R4=4.15 .**

**Write the MATLAB statements to do the following:**

**1) 3 will go from 84° to 804° in steps of 1° .**

**2) For each value of 3 , call the function newton2 to calculate 1 and 2 . Use 75° and 30° as the initial guesses for 1 and 2 and 1e-7 as the accuracy factor.**

**Use the variables t1, t2 and t3 for 1, 2 and 3 .**

**Do not write the plot statements or the pause statements.**

**Do not write the function newton2.**

**Answer**

**R1 = 3.24;**

**R2 = 3.78;**

**R3 = 2.56;**

**R4 = 4.15;**

**guess1 = 75\*pi/180;**

**guess2 = 30\*pi/180;**

**accuracy = 1e-7;**

**for(t3 = 84\*pi/180:1\*pi/180:804\*pi/180)**

**f1 = @(t1,t2) R1\*cos(t1)+R2\*cos(t2)+R3\*cos(t3)-R4;**

**f2 = @(t1,t2) R1\*sin(t1)-R2\*sin(t2)-R3\*sin(t3);**

**df1d1 = @(t1,t2) -R1\*sin(t1);**

**df1d2 = @(t1,t2) -R2\*sin(t2);**

**df2d1 = @(t1,t2) R1\*cos(t1);**

**df2d2 = @(t1,t2) -R2\*cos(t2);**

**[t1 t2] = newton2(f1,f2,df1d1,df1d2,df2d1,df2d2,guess1,guess2,accuracy);**

**end**

## 4)

**R2**

**R2y**

**2**

**R2x**

## R1

**R1y**

**R3y**

**R3y**

**R3**

**R3x**

**R2x**

**R1x**

**3**

**1**

**R4**

**Define the arrays needed to plot the four bar linkage shown above, and write the plot statement to plot R1, R2, R3 and R4 in blue, green, red and black.**

**Name the arrays line1x, line1y, line2x, line2y, line3x, line3y, line4x and line4y. Use the variables t1, t2 and t3 for 1, 2 and 3 .**

**Do not write any other statements for the graph except the plot statement.**

**Answer**

**line1x = [ 0 R1\*cos(t1) ];**

**line1y = [ 0 R1\*sin(t1) ];**

**line2x = [ R1\*cos(t1) R1\*cos(t1)+R2\*cos(t2) ];**

**line2y = [ R1\*sin(t1) R3\*sin(t3) ];**

**line3x = [ R1\*cos(t1)+R2\*cos(t2) R4 ];**

**line3y = [ R3\*sin(t3) 0 ];**

**line4x = [ 0 R4 ];**

**line4y = [ 0 0 ];**

**plot(line1x,line1y,'r',line2x,line2y,'g',line3x,line3y,'b',line4x,line4y,'k');**

**5) Consider the following system of 3 nonlinear equations for x, y and z:**

**cos(2\*x) = z - 2\*y\*sin(x)**

**cos(2\*x+c-h) = z + y\*(sin(h-x) - sin(c+x))**

**cos(2\*x+g-k) = z + y\*(sin(k-x) - sin(g+x))**

**where**

**c = /9**

**g = 2\*/9**

**h = cot-1(0.8+cot(c))**

**k = cot-1(0.8+cot(g))**

**Write a MATLAB program as follows:**

**Call the function newton3 to calculate x, y and z. Use 2, 0, 4 as the initial guesses for x, y, z and 1e-7 as the accuracy factor.**

**Do not write the function newton3.**

**The output of this program should look like this:**

**x=2.04833 y=0.52105 z=0.34797**

**Answer**

**c = pi/9;**

**g = 2\*pi/9;**

**h = acot(0.8 + cot(c));**

**k = acot(0.8 + cot(g));**

**guess1 = 2;**

**guess2 = 0;**

**guess3 = 4;**

**accuracy = 1e-7;**

**f1 = @(x,y,z) cos(2\*x) - z + 2\*y\*sin(x);**

**f2 = @(x,y,z) cos(2\*x+c-h) - z - y\*(sin(h-x) - sin(c+x));**

**f3 = @(x,y,z) cos(2\*x+g-k) - z - y\*(sin(k-x) - sin(g+x));**

**df1d1 = @(x,y,z) -2\*sin(2\*x)+2\*y\*cos(x);**

**df1d2 = @(x,y,z) 2\*sin(x);**

**df1d3 = @(x,y,z) -1;**

**df2d1 = @(x,y,z) -2\*sin(2\*x+c-h) - y\*(-cos(h-x) - cos(c+x));**

**df2d2 = @(x,y,z) -(sin(h-x) - sin(c+x));**

**df2d3 = @(x,y,z) -1;**

**df3d1 = @(x,y,z) -2\*sin(2\*x+g-k) - y\*(-cos(k-x) - cos(g+x));**

**df3d2 = @(x,y,z) -(sin(k-x) - sin(g+x));**

**df3d3 = @(x,y,z) -1;**

**[x y z] = newton3(f1,f2,f3,df1d1,df1d2,df1d3,df2d1,df2d2,df2d3,df3d1,df3d2,df3d3,...**

**guess1,guess2,guess3,accuracy);**

**fprintf('x=%.5f y=%.5f z=%.5f\n',x,y,z);**

**6) A drag racer starts from rest at time t=0. The differential equation for**

**the racer’s velocity v is:**

****

**where c = .0016 and v is in ft/sec .**

**Write a MATLAB program as follows:**

**1) t will go from 0 to 15 sec in steps of .001 sec .**

**2) Calculate v for each value of t. Use 1e-7 as the accuracy factors.**

**Then convert v to mph (the conversion factor to go from ft/sec to mph**

**is 60/88 .**

**3) Plot v versus t in blue. Just write the plot statement. Do not write any**

**other statements for the graph.**

**Answer**

**c = .0016;**

**t = 0:.001:15;**

**f = @(t,v) (3\*t^2+6\*t+150) / sqrt(t^2+4\*t+20) - c\*v^2;**

**u0 = 0;**

**options = odeset('AbsTol',1e-7,'RelTol',1e-7);**

**[t v] = ode45(f,t,u0,options);**

**v = v\*60/88;**

**plot(t,v,'b');**

**7) Gilpin’s system for modeling the behavior of three interacting species is given by the following differential equations:**

****

**where A = .0012, B = .011, C = .0014, D = .006, E = .0004, k = .45, and x, y and z are the populations of the three species.**

**Write a MATLAB program as follows:**

**1) t will go from 0 to 80 sec in steps of .001 sec .**

**2) Calculate x, y and z for each value of t. Use 1e-7 as the accuracy**

**factors and 1000, 500 and 600 as the initial values of x, y and z.**

**3) Plot x, y and z versus t using the colors blue, green and red.**

**Just write the plot statement. Do not write any other statements for the**

**graph.**

**This program has a function defined in a separate MATLAB file. Name this**

**function prog7f.**

**Write both the main program and the function.**

**Answer**

**% main program**

**t = 0:.001:80;**

**u0 = [1000 500 600];**

**options = odeset('AbsTol',1e-7,'RelTol',1e-7);**

**[t u] = ode45('prog7f',t,u0,options);**

**plot(t,u(:,1),'b',t,u(:,2),'g',t,u(:,3),'r');**

**% function prog7f**

**function f = prog7f(t,uf)**

**A = .0012;**

**B = .011;**

**C = .0014;**

**D = .006;**

**E = .0004;**

**k = .45;**

**x = uf(1);**

**y = uf(2);**

**z = uf(3);**

**f = zeros(3,1);**

**f(1) = x - A\*x^2 - A\*k\*x\*y - B\*x\*z;**

**f(2) = y - C\*k\*x\*y - A\*y^2 - A\*y\*z;**

**f(3) = -z + D\*x\*z + E\*y\*z;**

**8) In a damped spring-mass oscillator driven by a horizontal time-dependent**

**force, the mass is initially stationary with the spring stretched a distance of .5 m from its equilibrium position. The differential equation for the oscillator is**

****

**where the mass m = 3.6 kg, the damping coefficient c = 2.8 N·s/m, the spring constant k = 19 N/m, x is the displacement of the mass from equilibrium, t is the time, and the mass’s velocity v is the first derivative of x:**

****

**Write a MATLAB program as follows:**

**1) t will go from 0 to 20 sec in steps of .001 sec.**

**2) Calculate the displacement x and velocity v for each value of t. Use 1e-7**

**as the accuracy factors.**

**3) Plot x in blue and v in red versus t and the t axis in black.**

**Just write the plot statement. Do not write any other statements for the**

**graph.**

**This program has a function defined in a separate MATLAB file. Name this**

**function prog8f.**

**Write both the main program and the function.**

**Answer**

**% main program**

**t = 0:.001:20;**

**u0 = [.5 0];**

**options = odeset('AbsTol',1e-7,'RelTol',1e-7);**

**[t u] = ode45('prog8f',t,u0,options);**

**line1x = [0 20];**

**line1y = [0 0];**

**plot(t,u(:,1),'b',t,u(:,2),'r',line1x,line1y,'k');**

**% function prog8f**

**function f = prog8f(t,uf)**

**m = 3.6;**

**c = 2.8;**

**k = 19;**

**x = uf(1);**

**v = uf(2);**

**f = zeros(2,1);**

**f(1) = v;**

**f(2) = 1/m\*(32\*sin(3\*t)\*cos(5\*t) - c\*v - k\*x);**

**9)**



**In the mass-spring system shown above, the masses m1, m2 and m3 are .8, .6**

**and .5, the spring constants k1, k2, k3 and k4 are 4.3, 5.1, 4.6 and 5.4, and x1, x2 and x3 are the displacements of m1, m2 and m3 from their equilibrium positions.**

**Write a MATLAB program as follows:**

**1) t will go from 0 to 8 sec in steps of .001 sec.**

**2) Calculate the displacements and velocities of the masses for each**

**value of t. Use 1e-7 as the accuracy factors, .7, .2 and .4 as the**

**initial values of x1, x2 and x3 , and 0 as the initial values of the**

**velocities.**

**3) Plot x1, x2 and x3 versus t using the colors blue, red and green and**

**the t axis in black.**

**4) In a separate figure, plot the velocities v1, v2 and v3 versus t using**

**the colors blue, red and green and the t axis in black.**

**Just write the figure statements and the plot statements. Do not write any**

**other statements for the graphs.**

**This program has a function defined in a separate MATLAB file. Name this**

**function prog9f.**

**Write both the main program and the function.**

**Equations**

****

**Answer is on next page.**

**Question 9 Answer**

**% main program**

**t=0:.001:8;**

**u0 = [.7 0 .2 0 .4 0];**

**options = odeset('AbsTol',1e-7,'RelTol',1e-7);**

**[t u] = ode45('prog9f',t,u0,options);**

**line1x = [0 8];**

**line1y = [0 0];**

**figure(1);**

**plot(t,u(:,1),'b',t,u(:,3),'r',t,u(:,5),'g',line1x,line1y,'k');**

**figure(2);**

**plot(t,u(:,2),'b',t,u(:,4),'r',t,u(:,6),'g',line1x,line1y,'k');**

**% function prog9f**

**function f = prog9f(t,uf)**

**m1 = .8;**

**m2 = .6;**

**m3 = .5;**

**k1 = 4.3;**

**k2 = 5.1;**

**k3 = 4.6;**

**k4 = 5.4;**

**x1 = uf(1);**

**v1 = uf(2);**

**x2 = uf(3);**

**v2 = uf(4);**

**x3 = uf(5);**

**v3 = uf(6);**

**f = zeros(6,1);**

**f(1) = v1;**

**f(2) = 1/m1\*( -k1\*x1 + k2\*(x2-x1) );**

**f(3) = v2;**

**f(4) = 1/m2\*( -k2\*(x2-x1) + k3\*(x3-x2) );**

**f(5) = v3;**

**f(6) = 1/m3\*( -k3\*(x3-x2) - k4\*x3 );**