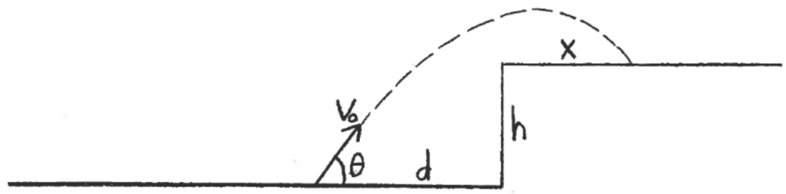
Practice Questions for Exam 1 on Thurs Feb 27, 6-8 PM, in Williams 102

1)



An artillery shell is fired at a horizontal distance d from the bottom of a cliff of height h, with a muzzle velocity v_0 at an angle θ with respect to the horizontal, toward a target at a horizontal distance x from the top of the cliff. The firing angle θ that will permit the shell to hit the target is given by the equation

$$\frac{V_0 \cos\theta}{g}\left(V_0 \sin\theta + \sqrt{V_0^2 \sin^2\theta - 2gh}\right) = d + x$$

where v0 = 185 m/s, d = 650 m, x = 1700 m, and g = 9.81 m/s^2 .

The height, y, of the shell when it passes over the top of the cliff, and the maximum height of the shell with respect to the ground, y_{max}, are given by

$$y = d\tan\theta - \frac{gd^2}{2V_0^2\cos^2\theta}$$

$$y_{max} = \frac{V_0^2 \sin^2\theta}{2g}$$

Write a MATLAB program as follows:

1) h will go from 350 m to 550 m in steps of 100 m.
2) For each value of h, call the function newton to calculate both values of θ that will enable the shell to hit the target. Then use these calculated values of θ to calculate y and ymax according to the above formulas. Scan the θ axis from 30° to 90° in steps of 1° to look for solutions for θ . Use 1e-7 as the accuracy factor.
   Print h, θ (in degrees), y and y_{max}.

Do not write the function newton.

The output of this program should look like this:

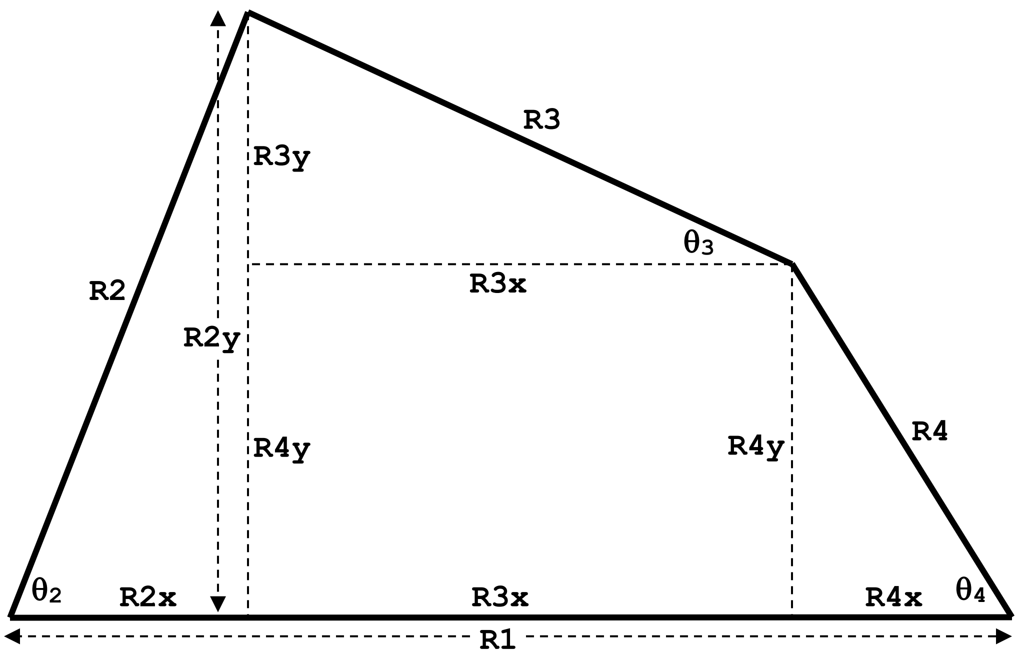
```
h=350   theta=31.45765   y= 314.44573   ymax= 475.07933
h=350   theta=67.01349   y=1135.25444   ymax=1478.37096

h=450   theta=34.50636   y= 357.67243   ymax= 559.81004
h=450   theta=66.33395   y=1107.32878   ymax=1463.32618

h=550   theta=37.63518   y= 404.65061   ymax= 650.43352
h=550   theta=65.53737   y=1075.65163   ymax=1445.26751
```

Answer is on the next page

Problem 1 Answer

```
d = 650;
x = 1700;
g = 9.81;
v0 = 185;
accuracy = 1e-7;
for(h = 350:100:550)
  f=@(theta)  v0*cos(theta)/g * ( v0*sin(theta) + ...
              sqrt(v0^2*sin(theta)^2-2*g*h) ) - d - x;
  fp=@(theta) -v0*sin(theta)/g * ( v0*sin(theta) + ...
              sqrt(v0^2*sin(theta)^2-2*g*h) ) + ...
              v0*cos(theta)/g * ( v0*cos(theta) + ...
              1/(2*sqrt(v0^2*sin(theta)^2-2*g*h))  * ...
              2*v0^2*sin(theta)*cos(theta) );
  stepsize = 1*pi/180;
  for(left_end_point = 30*pi/180:stepsize:90*pi/180)
    right_end_point = left_end_point + stepsize;
    function_left = f(left_end_point);
    function_right = f(right_end_point);
    if(function_left * function_right < 0)
       guess = (left_end_point + right_end_point)/2;
       theta = newton(f, fp, guess, accuracy);
       y = d*tan(theta) - g*d^2/(2*v0^2*cos(theta)^2);
       ymax = v0^2*sin(theta)^2/(2*g);
       fprintf('h=%d theta=%.5f y=%10.5f ymax=%10.5f\n',h,theta*180/pi,y,ymax);
    end
    if(function_left == 0)
       theta = left_end_point;
       y = d*tan(theta) - g*d^2/(2*v0^2*cos(theta)^2);
       ymax = v0^2*sin(theta)^2/(2*g);
       fprintf('h=%d theta=%.5f y=%10.5f ymax=%10.5f\n',h,theta*180/pi,y,ymax);
    end
    fprintf('\n');
  end
```

2)



In the four bar linkage shown above, R4 is the driver, R2 is the follower, R3 is the connector, and R1 is the frame, where R1=4.14, R2=3.25, R3=3.77, R4=2.57 .

Write the MATLAB statements to do the following:

1) θ_4 will go from 84° to 804° in steps of 1° .

2) For each value of θ_4 , <u>call the function newton2</u> to calculate θ_2 and θ_3 . Use 75° and 30° as the initial guesses for θ_2 and θ_3 and 1e-7 as the accuracy factor.

Use the variables t2, t3 and t4 for θ_2, θ_3 and θ_4 .

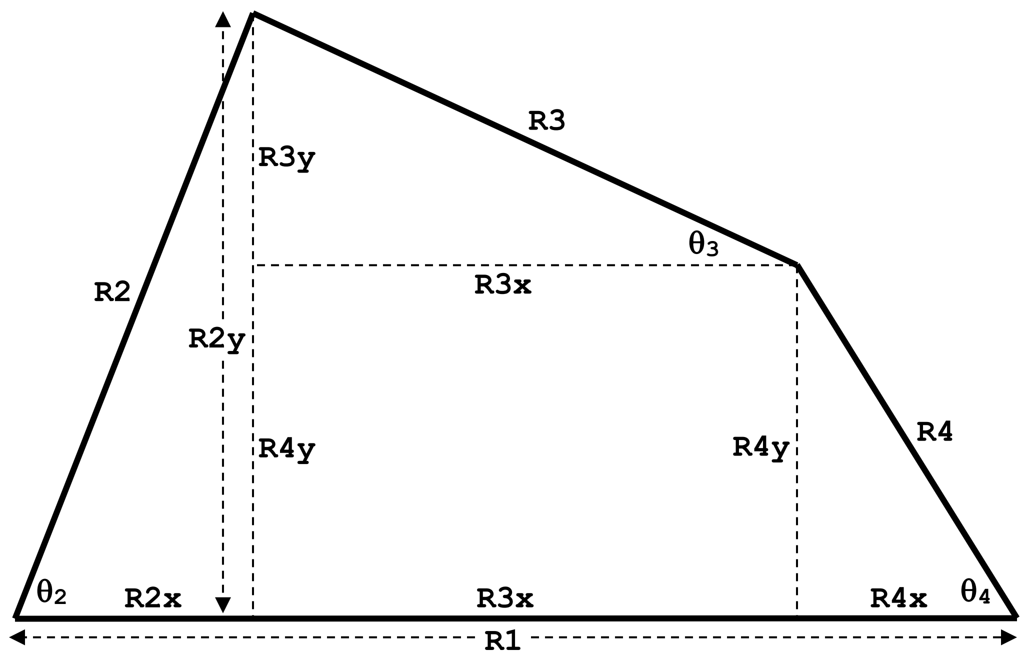
<u>Do not write the plot statements or the pause statements</u>.

<u>Do not write the function newton2</u>.

<u>Answer</u>

```
R1 = 4.14;
R2 = 3.25;
R3 = 3.77;
R4 = 2.57;
guess2 = 75*pi/180;
guess3 = 30*pi/180;
accuracy = 1e-7;
for(t4 = 84*pi/180 : 1*pi/180 : 804*pi/180)
   f1 = @(t2,t3) R2*cos(t2) + R3*cos(t3) + R4*cos(t4) - R1;
   f2 = @(t2,t3) R3*sin(t3) + R4*sin(t4) - R2*sin(t2);
   df1dt2 = @(t2,t3) -R2*sin(t2);
   df1dt3 = @(t2,t3) -R3*sin(t3);
   df2dt2 = @(t2,t3) -R2*cos(t2);
   df2dt3 = @(t2,t3)  R3*cos(t3);
   [t2  t3] = newton2(f1,f2,df1dt2,df1dt3,df2dt2,df2dt3,guess2,guess3,accuracy);
end
```

3)



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 black, red, green and blue.

Name the arrays line1x, line1y, line2x, line2y, line3x, line3y, line4x and line4y. Use the variables t2, t3 and t4 for θ_2, θ_3 and θ_4 .

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

Answer

```
R2x = R2*cos(t2);
R2y = R2*sin(t2);
R3x = R3*cos(t3);
R3y = R3*sin(t3);    % not used in this particular problem
R4x = R4*cos(t4);    % not used in this particular problem
R4y = R4*sin(t4);
line1x = [ 0  R1 ];
line1y = [ 0  0  ];
line2x = [ 0  R2x ];
line2y = [ 0  R2y ];
line3x = [ R2x  R2x+R3x ];
line3y = [ R2y  R4y ];
line4x = [ R2x+R3x  R1 ];
line4y = [ R4y      0  ];
plot(line1x,line1y,'k',line2x,line2y,'r',line3x,line3y,'g',line4x,line4y,'b');
```

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

$$xy = z^2 + 1$$

$$xyz + y^2 = x^2 + 2$$

$$e^x + z = e^y + 3$$

Write a MATLAB program as follows:

Call the function newton3 to calculate x, y and z. Use 2, 1.5, 1 as the initial guesses for x, y, z and 1e-7 as the accuracy factor. Print x, y and z.

The output of this program should look like this:

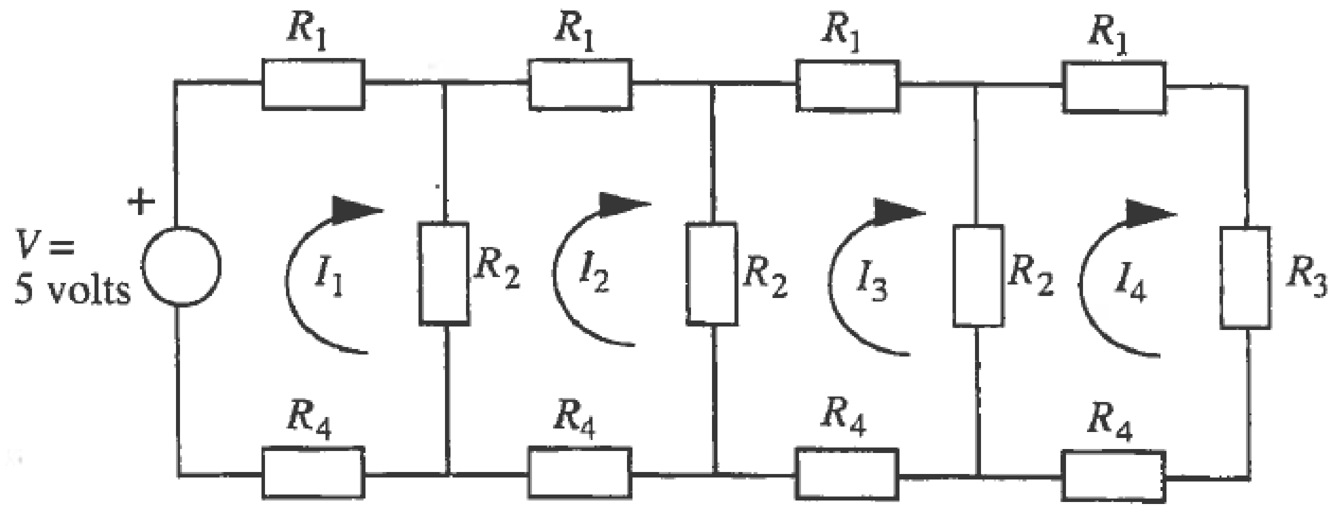
```
x=1.77767  y=1.42396  z=1.23747
```

Do not write the function newton3.

Answer

```
guess1 = 2;
guess2 = 1.5;
guess3 = 1;
accuracy = 1e-7;
f1 = @(x,y,z) x*y - z^2 - 1;
f2 = @(x,y,z) x*y*z + y^2 - x^2 - 2;
f3 = @(x,y,z) exp(x) + z - exp(y) - 3;
df1dx = @(x,y,z)  y;
df1dy = @(x,y,z)  x;
df1dz = @(x,y,z) -2*z;
df2dx = @(x,y,z)  y*z - 2*x;
df2dy = @(x,y,z)  x*z + 2*y;
df2dz = @(x,y,z)  x*y;
df3dx = @(x,y,z)  exp(x);
df3dy = @(x,y,z) -exp(y);
df3dz = @(x,y,z)  1;
[x,y,z]=newton3(f1,f2,f3,df1dx,df1dy,df1dz,df2dx,df2dy,df2dz, ...
                df3dx,df3dy,df3dz,guess1,guess2,guess3,accuracy);
fprintf('x=%.5f  y=%.5f  z=%.5f\n',x,y,z);
```

5)



In the electrical network shown above, the equations for the currents I_1 - I_4 are:

$$(R_1+R_2+R_4)I_1 - R_2I_2 = V$$

$$(R_1+2R_2+R_4)I_2 - R_2I_1 - R_2I_3 = 0$$

$$(R_1+2R_2+R_4)I_3 - R_2I_2 - R_2I_4 = 0$$

$$(R_1+R_2+R_3+R_4)I_4 - R_2I_3 = 0$$

where $R_1=0.3$, $R_2=0.6$, $R_3=0.2$, $R_4=0.5$, and $V=5$. Write a MATLAB program to calculate and print the currents I_1 - I_4 .

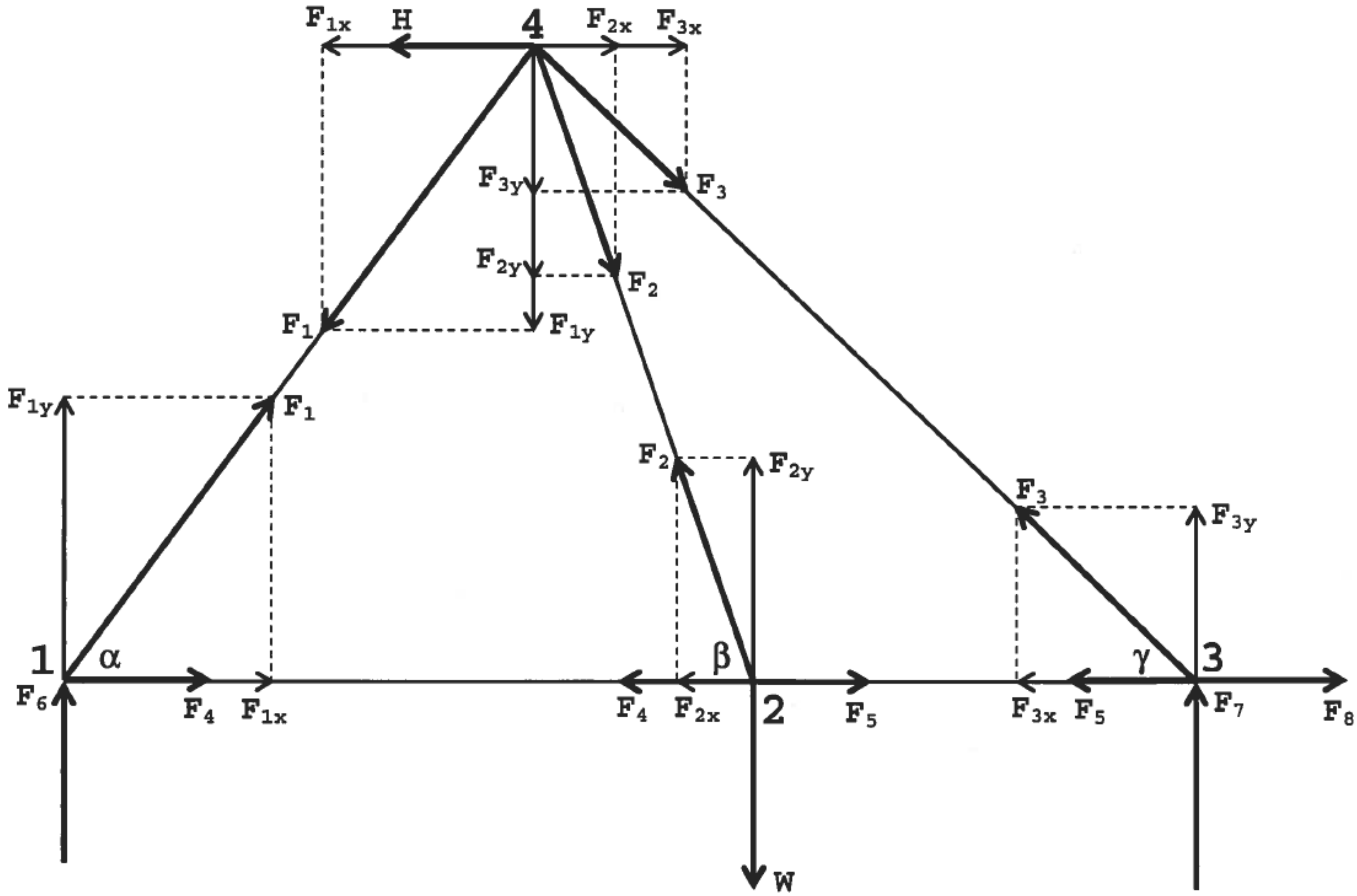
The output of this program should look like this:

```
I =

    4.1678
    1.3914
    0.4703
    0.1764
```

Answer

```
R1 = .3;
R2 = .6;
R3 = .2;
R4 = .5;
V = 5;
d = [ V
      0
      0
      0 ];
a = [ R1+R2+R4     -R2            0             0
     -R2            R1+2*R2+R4   -R2            0
      0            -R2            R1+2*R2+R4   -R2
      0             0            -R2            R1+R2+R3+R4 ];
b = inv(a);
I = b*d;
I
```

6)



In the truss shown above, α=62°, β=73°, γ=38°, W=300, and H=350. Write a MATLAB program to calculate and print the unknown forces F_1 – F_8 .

The output of this program should look like this:

```
F =

 -401.5167
  313.7075
   88.5522
  188.5007
  280.2199
  354.5182
  -54.5182
  350.0000
```

The steps in writing down the equations and defining the arrays are shown on the next page, and the program is shown on the page after that.

Problem 6 Steps

Step 1

For each node, write down the following two equations:

1) The sum of the x components of the forces acting on the node is 0
2) The sum of the y components of the forces acting on the node is 0

In these equations, whenever you get an external force (W or H) on the left of the equals sign, move it to the right. This method, called the method of joints, will give you 8 equations for the 8 unknown forces F_1-F_8:

$$F_{1x} + F_4 = 0$$

$$F_{1y} + F_6 = 0$$

$$-F_{2x} - F_4 + F_5 = 0$$

$$F_{2y} = W$$

$$-F_{3x} - F_5 + F_8 = 0$$

$$F_{3y} + F_7 = 0$$

$$-F_{1x} + F_{2x} + F_{3x} = H$$

$$-F_{1y} - F_{2y} - F_{3y} = 0$$

Step 2

Put the equations in final form by substituting what the x and y components of the forces are equal to:

$$F_1\cos(\alpha) + F_4 = 0$$

$$F_1\sin(\alpha) + F_6 = 0$$

$$-F_2\cos(\beta) - F_4 + F_5 = 0$$

$$F_2\sin(\beta) = W$$

$$-F_3\cos(\gamma) - F_5 + F_8 = 0$$

$$F_3\sin(\gamma) + F_7 = 0$$

$$-F_1\cos(\alpha) + F_2\cos(\beta) + F_3\cos(\gamma) = H$$

$$-F_1\sin(\alpha) - F_2\sin(\beta) - F_3\sin(\gamma) = 0$$

Step 3

Define the arrays d and a. The array d is the column on the right of the equals signs in the equations. The array a contains the coefficients of the unknown forces in the equations. The first column of the array a contains the coefficients of F1, the second column contains the coefficients of F2, etc. The first equation gives the first row of the array a, the second equation gives the second row, etc. If a force does not appear in an equation, its coefficient is 0 .

```
d=[ 0        a=[ cos(alpha)    0             0              1   0   0   0   0
    0            sin(alpha)    0             0              0   0   1   0   0
    0            0            -cos(beta)     0             -1   1   0   0   0
    W            0             sin(beta)     0              0   0   0   0   0
    0            0             0            -cos(gamma)     0  -1   0   0   1
    0            0             0             sin(gamma)     0   0   0   1   0
    H           -cos(alpha)    cos(beta)     cos(gamma)     0   0   0   0   0
    0 ];        -sin(alpha)   -sin(beta)    -sin(gamma)     0   0   0   0   0 ];
```

Problem 6 Answer

```
alpha = 62*pi/180;
beta = 73*pi/180;
gamma = 38*pi/180;
W = 300;
H = 350;
d = [ 0
      0
      0
      W
      0
      0
      H
      0 ];
a = [ cos(alpha)    0             0              1    0    0    0    0
      sin(alpha)    0             0              0    0    1    0    0
      0            -cos(beta)     0             -1    1    0    0    0
      0             sin(beta)     0              0    0    0    0    0
      0             0            -cos(gamma)     0   -1    0    0    1
      0             0             sin(gamma)     0    0    0    1    0
     -cos(alpha)    cos(beta)     cos(gamma)     0    0    0    0    0
     -sin(alpha)   -sin(beta)    -sin(gamma)     0    0    0    0    0 ];
b = inv(a);
F = b*d;
F
```

7) The angular velocity ω of a particle moving in a circle is 3 rad/s at t=0. The differential equation for ω is

$$\frac{d\omega}{dt} + k\omega^2 = \sqrt{2t^2+3t+16} \cdot \ln\left(\sqrt{5t^2+120}\right)$$

where k=.0547 .

Write a MATLAB program to do the following:

1) t will go from 0 to 4 sec in steps of .001 sec .

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

3) Plot ω versus t in blue.
   Just write the plot statement. Do not write any other statements for the graph.

Answer

```
k = .0547;
t = 0:.001:4;
w0 = 3;
f = @(t,w) sqrt(2*t^2+3*t+16) * log(sqrt(5*t^2+120)) - k*w^2;
options = odeset('RelTol',1e-7,'AbsTol',1e-7);
[t  w] = ode45(f,t,w0,options);
plot(t,w,'b');
```

8) The differential equations for the Predator-Prey Problem are:

$$\frac{dH}{dt} = K_1 H - CHL - S_1 H$$

$$\frac{dL}{dt} = -K_2 L + DHL - S_2 L$$

where H is the hare population, L is the lynx population, and K_1, K_2, C, D, S_1, and S_2 are 2, 10, .0012, .0019, .63, and .57, respectively.

Write a MATLAB program as follows:

1) t will go from 0 to 9 sec in steps of .001 sec .

2) Calculate H and L for each value of t. Use 1e-7 as the accuracy factors and 4000 and 200 as the initial values of H and L.

3) Plot H and L versus t using the colors blue and red. <u>Just write the plot statement. Do not write any other statements for the graph.</u>

This program has a function defined in a separate MATLAB file. Name this function prog8f.

<u>Write both the main program and the function</u>.

<u>Answer</u>

```
% main program
t = 0:.001:9;
u0 = [4000  200];
options = odeset('RelTol',1e-7,'AbsTol',1e-7);
[t  u] = ode45('prog8f',t,u0,options);
plot(t,u(:,1),'b',t,u(:,2),'r');

% function prog8f
function f = prog8f(t,uf)
K1 = 2;
K2 = 10;
C = .0012;
D = .0019;
S1 = .63;
S2 = .57;
H = uf(1);
L = uf(2);
f = zeros(2,1);
f(1) =  K1*H - C*H*L - S1*H;
f(2) = -K2*L + D*H*L - S2*L;
```

9) Find the inverse of the 2x2 matrix A:

$$A = \begin{bmatrix} 3 & -2 \\ 2 & 1 \end{bmatrix}$$

<u>**Answer**</u>

Put a 2x2 unit matrix next to the matrix A:

```
3  -2    1  0
2   1    0  1
```

Then do the following row operations:

1) Put a 1 in the diagonal element of row 1 by dividing every element on row 1 by the diagonal element 3. Row 2 remains unchanged in this step.

```
1  -2/3    1/3  0
2   1      0    1
```

2) Put a 0 everywhere else in the column where you just put the 1 (put a 0 in the first element of row 2) by doing the following:
   a) Let c be the first element of row 2 (here c is 2).
   b) Substract c times row 1 from row 2 and put the result in row 2 (replace row 2 with the result of this operation). Row 1 remains unchanged in this step.

```
1  -2/3     1/3  0
0   7/3    -2/3  1
```

3) Put a 1 in the diagonal element of row 2 by dividing every element on row 2 by the diagonal element 7/3: Row 1 remains unchanged in this step.

```
1  -2/3     1/3  0
0   1      -2/7  3/7
```

4) Put a 0 everywhere else in the column where you just put the 1 (put a 0 in the second element of row 1) by doing the following:
   a) Let c be the second element of row 1 (here c is -2/3).
   b) Substract c times row 2 from row 1 and put the result in row 1 (replace row 1 with the result of this operation). Row 2 remains unchanged in this step.

```
1   0       1/7  2/7
0   1      -2/7  3/7
```

Once the matrix on the left has become a unit matrix, the matrix on the right is now the inverse of A:

$$A^{-1} = \begin{bmatrix} 1/7 & 2/7 \\ -2/7 & 3/7 \end{bmatrix}$$