Problem 1

```
function [x,t] = BVP_shooting(f,a,b,alpha,beta,h,TOL,MaxIters)
    % Rewrite y'' = f(t,y,y') using x_0 = t,x_1 = y,x_2 = y'
    % x_0' = 1
    % x_1' = x_2
    % x_2' = f(x_0, x_1, x_2)

    % Function phi using RK4
    steps = round((b-a)/h);
    F = @(X) [1, X(3), f(X(1), X(2), X(3))];
    function out = phi(z)
        X = [0, alpha, z];
        for i = 1:steps
            F1 = h*F(X);
            F2 = h*F(X + F1/2);
            F3 = h*F(X + F2/2);
            F4 = h*F(X + F3);
            X = X + (F1 + 2*F2 + 2*F3 + F4)/6;
        end
        out = X(2) - beta;
    end
    % Initial Guesses for x'
    z1 = 0;
    z2 = 1;
    phi_z1 = phi(z1);
    phi_z2 = phi(z2);
    % Secant Method
    for i = 1:MaxIters
        if phi_z2-phi_z1 == 0
            error("Divide by zero");
        end
        z3 = z2 - phi_z2*(z2-z1)/(phi_z2-phi_z1);
        z1 = z2;
        z2 = z3;
        phi_z1 = phi_z2;
        phi_z2 = phi(z2);
        if abs(phi_z2) < TOL
            break
        end
    end
    % Use RK4 with z2 as initial velocity
    x = zeros(steps+1,1);
    t = linspace(a,b,steps+1)';
    x(1) = alpha;
    X = [0, alpha, z2];
    for i = 2:steps+1
        F1 = h*F(X);
        F2 = h*F(X + F1/2);
        F3 = h*F(X + F2/2);
        F4 = h*F(X + F3);
        X = X + (F1 + 2*F2 + 2*F3 + F4)/6;
        x(i) = X(2);
    end

end
```

```
>> [x,t]=BVP_shooting(@(t, y, yp) exp(t) + y * cos(t) - (t + 1) * yp,0,1,1,3,0.1,0.0001,10)

x =

1.0000
1.1865
1.3752
1.5668
1.7617
1.9602
2.1623
2.3678
2.5764
2.7874
3.0000


t =

0
0.1000
0.2000
0.3000
0.4000
0.5000
0.6000
0.7000
0.8000
0.9000
1.0000

>> [x,t]=BVP_shooting(@(t, y, yp) exp(t) + y * cos(t) - (t + 1) * yp,0,1,1,3,0.05,0.0001,10)

x =

1.0000
1.0930
1.1865
1.2805
1.3752
1.4706
1.5668
1.6638
1.7617
1.8605
1.9602
2.0608
2.1623
2.2647
2.3678
2.4718
2.5764
2.6816
2.7874
2.8935
3.0000


t =

0
0.0500
0.1000
0.1500
0.2000
0.2500
0.3000
0.3500
0.4000
0.4500
0.5000
0.5500
0.6000
0.6500
0.7000
0.7500
0.8000
0.8500
0.9000
0.9500
1.0000
```

Problem 2

```
function [x,t] = BVP_finitediff(u,v,w,a,b,alpha,beta,h)
    n = round((b-a)/h)-1;
    % Construct tridiagonal matrix, A
    A = zeros(n,n);
    for i=1:n-1
        A(i,i) = 2+h^2*v(a+i*h);            %d_i;
        A(i+1, i) = -1 - h*w(a+(i+1)*h)/2; %a_i
        A(i, i+1) = -1 + h*w(a+i*h)/2;      %c_i
    end
    A(n,n) = 2+h^2*v(n*h);
    % Construct RHS vector, B
    B = zeros(n,1);
    for i=1:n
        B(i) = -h^2*u(a+i*h);
    end
    B(1) = B(1) - (-1 - h*w(a+1*h)/2)*alpha;
    B(n) = B(n) - (-1 - h*w(a)/2)*beta;
    % Return answer
    t = linspace(a,b,n+2)';
    x = [alpha;A\B;beta];
end
```

We can see that with finite differences there is a jump between y_n and y_{n+1} that gets smaller as $h \to 0$. This jump does not exist with the shooting method.

```
>> [x,t] = BVP_finitediff(@(t)exp(t),@(t)cos(t),@(t) -(t+1),0,1,1,3,0.1)

x =

    1.0000
    1.1273
    1.2624
    1.4054
    1.5563
    1.7147
    1.8804
    2.0528
    2.2312
    2.4148
    3.0000


t =

         0
    0.1000
    0.2000
    0.3000
    0.4000
    0.5000
    0.6000
    0.7000
    0.8000
    0.9000
    1.0000

>> [x,t] = BVP_finitediff(@(t)exp(t),@(t)cos(t),@(t) -(t+1),0,1,1,3,0.05)

x =

    1.0000
    1.0769
    1.1551
    1.2345
    1.3154
    1.3975
    1.4812
    1.5662
    1.6527
    1.7406
    1.8299
    1.9206
    2.0127
    2.1060
    2.2006
    2.2963
    2.3931
    2.4909
    2.5896
    2.6890
    3.0000


t =

         0
    0.0500
    0.1000
    0.1500
    0.2000
    0.2500
    0.3000
    0.3500
    0.4000
    0.4500
    0.5000
    0.5500
    0.6000
    0.6500
    0.7000
    0.7500
    0.8000
    0.8500
    0.9000
    0.9500
    1.0000
```