Math 337 Homework 07

Andy Reagan

March 2, 2014

1. Solve the BVP
$$y'' + xy' - 3y = 3x, y(0) = 1, y(2) = 5$$
by the shooting method. Use the modified Euler method with $h = 0.1$ as the IVP solver. Plot your solution.

   Solution: My code and a solution plot follow.

```
% HW07 Problem 01
%
% solve the BVP using the shooting method
% - I define the homogeneous and non-homogeneous in separate files
% - the IC are both Nuemann

% IC, BC
y0 = 1; yf = 5;
h = 0.1;
tvec = 0:h:2;

i=1;
shot1 = andy_ME(@andy_hw07_prb01_ODE,tvec,[y0;0],h,[]);
shot2 = andy_ME(@andy_hw07_prb01_ODEh,tvec,[0;1],h,[]);

theta = (yf-shot1(1,end))/shot2(1,end);

soln = shot1+theta.*shot2;
plot(tvec,soln(1,:));
xlabel('x','FontSize',20);
ylabel('y','FontSize',20);
set(gcf, 'units', 'inches', 'position', [1 1 10 10])
set(gcf,'PaperPositionMode','auto')
print('-depsc2','-zbuffer','-r200',sprintf('andy_hw07_prb01_%02g.eps',i))
system(sprintf('epstopdf andy_hw07_prb01_%02g.eps',i));
```

2. Solve the BVP
$$x^3y''' + xy' - y = -3 + \ln x, y(1) = 1, y'(2) = 1/2, y''(2) = 1/4$$
by "shooting" from the left end point and using the example given in the notes. Use the ME method with $h = 0.02$. Plot your solution

   Solution: My code and a solution plot follow.

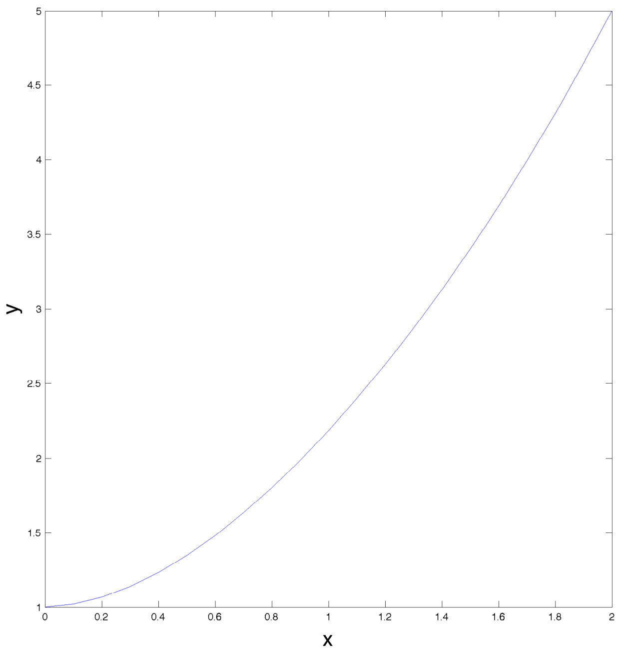

Figure 1: Solution of the BVP with the shooting method.

```
% HW07 Problem 01
%
% solve the BVP using the shooting method
% - I define the homogeneous and non-homogeneous in separate files
% - the IC are both Nuemann

% IC, BC
y0 = 1; yfp = 1/2; yfpp = 1/4;
h = 0.02;
tvec = 1:h:2;

i=1;
% solve the three IVP's
shot1 = andy_ME(@andy_hw07_prb02_ODE,tvec,[y0;0;0],h,[]);
shot2 = andy_ME(@andy_hw07_prb02_ODEh,tvec,[0;1;0],h,[]);
shot3 = andy_ME(@andy_hw07_prb02_ODEh,tvec,[0;0;1],h,[]);

% construct z
z = [shot1(:,end) shot2(:,end) shot3(:,end)];
% take just the bottom two
z = z(2:3,:);

% solve for theta,psi. call them both theta
theta = z(:,2:3)\[yfp-z(1,1);yfpp-z(2,1)];

soln = shot1+theta(1).*shot2+theta(2).*shot3;
plot(tvec,soln);
legend('y','y','ypp')
xlabel('x','FontSize',20);
ylabel('y','FontSize',20);
set(gcf, 'units', 'inches', 'position', [1 1 10 10])
set(gcf,'PaperPositionMode','auto')
print('-depsc2','-zbuffer','-r200',sprintf('andy_hw07_prb02_%02g.eps',i))
system(sprintf('epstopdf andy_hw07_prb02_%02g.eps',i));
```

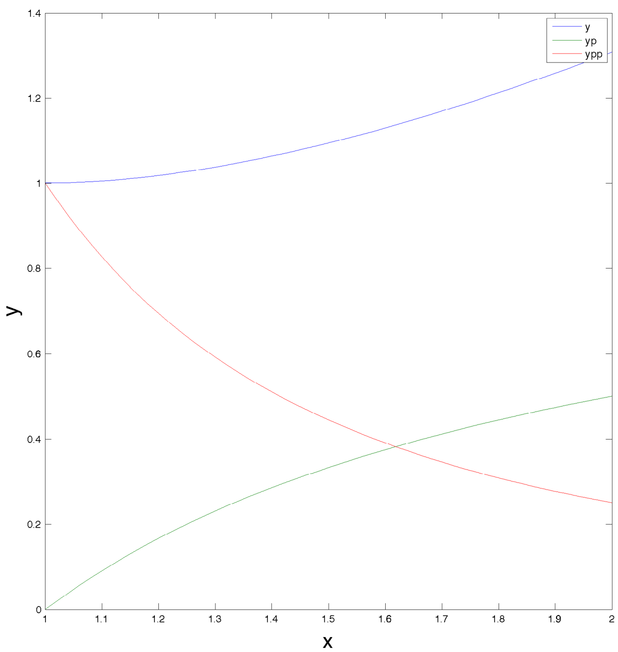

Figure 2: Solution of the BVP with the shooting method. Here yp denotes y' and ypp denotes y''.

3. Solve problem 2, in reverse.

   Solution: My code and a solution plot follow. I found it easiest to write a ME function that goes backwards.

```
% HW07 Problem 01
%
% solve the BVP using the shooting method
% - I define the homogeneous and non-homogeneous in separate files
% - the IC are both Nuemann

% IC, BC
y0 = 1; yfp = 1/2; yfpp = 1/4;
h = 0.02;
tvec = 1:h:2;

i=1;
% solve the three IVP's
shot1 = andy_MEr(@andy_hw07_prb02_ODE,tvec,[0;yfp;yfpp],h,[]);
shot2 = andy_MEr(@andy_hw07_prb02_ODEh,tvec,[1;0;0],h,[]);

% solve for theta,psi. call them both theta
theta = (y0-shot1(1,1))/shot2(1,1);

soln = shot1+theta.*shot2;
plot(tvec,soln);
legend('y','yp','ypp')
xlabel('x','FontSize',20);
ylabel('y','FontSize',20);
set(gcf, 'units', 'inches', 'position', [1 1 10 10])
set(gcf,'PaperPositionMode','auto')
print('-depsc2','-zbuffer','-r200',sprintf('andy_hw07_prb03_%02g.eps',i))
system(sprintf('epstopdf␣andy_hw07_prb03_%02g.eps',i));
```

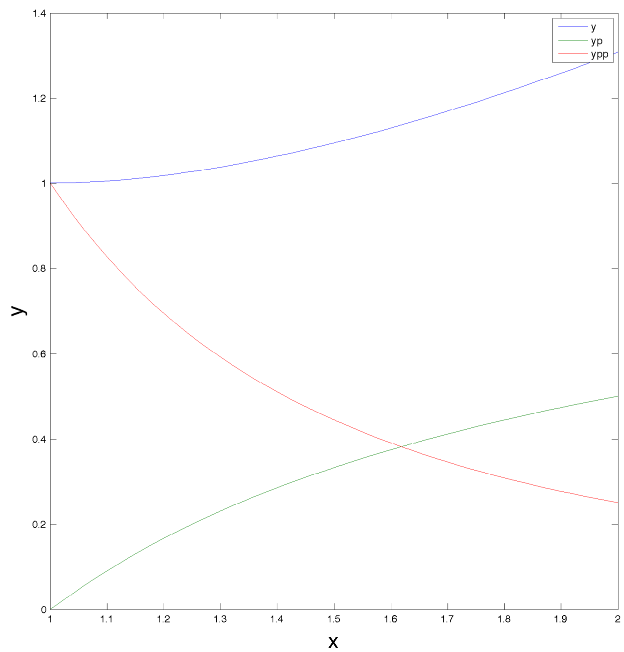

Figure 3: Solution of the BVP with the shooting method. Here yp denotes y' and ypp denotes y''.

4. Solve the BVP

$$y'' = 30^2(y - 1 + 2x), y(0) = 1, y(1.62) = -2.24$$

by the usual (i.e., not multiple) shooting method. Use any appropriate IVP-solving method and any reasonable value for h. Plot your numerical solution along with the exact solution, found in the notes.

Solution: My code and a solution plot follow.

```
% HW07 Problem 04
%
% solve the BVP using the shooting method
% - I define the homogeneous and non-homogeneous in separate files
% - the IC are both Nuemann

% IC, BC
y0 = 1; yf = -2.24;
h = 0.0001;
tvec = 0:h:1.62;

i=1;
shot1 = andy_ME(@andy_hw07_prb04_ODE,tvec,[y0;0],h,[]);
shot2 = andy_ME(@andy_hw07_prb04_ODEh,tvec,[0;1],h,[]);

theta = (yf-shot1(1,end))/shot2(1,end);

soln = shot1+theta.*shot2;
plot(tvec,soln(1,:));
xlabel('x','FontSize',20);
ylabel('y','FontSize',20);
```

```
set(gcf, 'units', 'inches', 'position', [1 1 10 10])
set(gcf,'PaperPositionMode','auto')
print('-depsc2','-zbuffer','-r200',sprintf('andy_hw07_prb04_%02g.eps',i))
system(sprintf('epstopdf andy_hw07_prb04_%02g.eps',i));
```

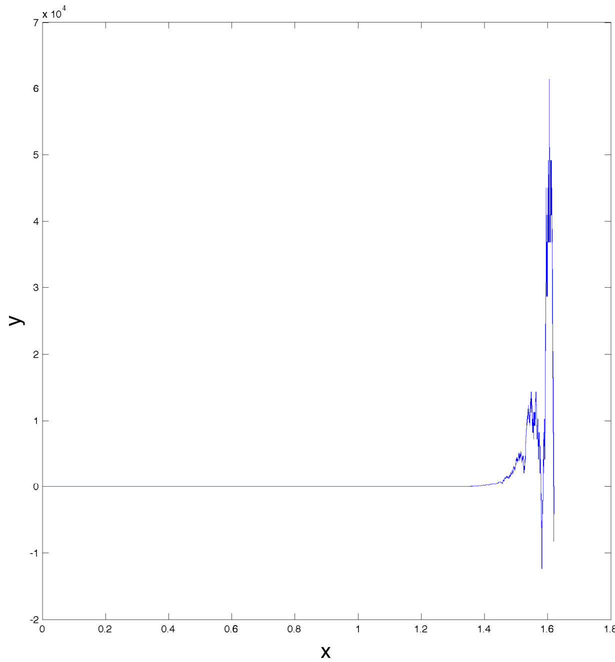

Figure 4: Solution of the BVP with the (regular) shooting method.

5. Solve the nonlinear BVP

$$y'' = \frac{y^2}{2+x}, y(0) = 1, y(2) = 1$$

following the outline given in Sec. 7.5. Find (and, of course, plot) the solutions of this BVP corresponding to both $\overline{\theta}$ and $\overline{\overline{\theta}}$.

Solution: My code and a solution plot follow.

```
% HW07 Problem 05
%
% solve the nonlinear BVP using the shooting method
% shoot iteratively
% use the secand method for solving

% IC, BC
y0 = 1; yf = 1;
h = 0.02;
tvec = 0:h:2;
tol = 10^(-3);

% initialize theta and f
thetacell = {[-2,-1],[2,1]};
for j=1:2
```

```
    err = 1;
    i=2;
    theta = thetacell{j};
    % disp(theta);
    shot1 = andy_ME(@andy_hw07_prb05_ODE,tvec,[y0;theta(1)],h,[]);
    shot2 = andy_ME(@andy_hw07_prb05_ODE,tvec,[y0;theta(2)],h,[]);
    f = [shot1(1,end),shot2(1,end)];
    while err > tol
        % generate new theta
        theta = [theta theta(end)-f(end)/((f(end)-f(end-1))/(theta(end)-theta(end-1)))
            ];
        % solve the ODE
        shot = andy_ME(@andy_hw07_prb05_ODE,tvec,[y0;theta(end)],h,[]);
        % f is y(2) at that theta
        f = [f shot(1,end)-yf];
        % count the iterations
        i=i+1;
        % error is just f
        err = abs(f(end));
    end
    fprintf('took␣%g␣iterations␣for␣theta␣%g␣\n',i,j);
    % disp(theta);

    soln = shot;
    figure;
    plot(tvec,soln(1,:));
    xlabel('x','FontSize',20);
    ylabel('y','FontSize',20);
    set(gcf, 'units', 'inches', 'position', [1 1 10 10])
    set(gcf,'PaperPositionMode','auto')
    print('-depsc2','-zbuffer','-r200',sprintf('andy_hw07_prb05_%02g.eps',j))
    system(sprintf('epstopdf␣andy_hw07_prb05_%02g.eps;␣\\rm␣andy_hw07_prb05_%02g.eps',
        j,j));
end
```

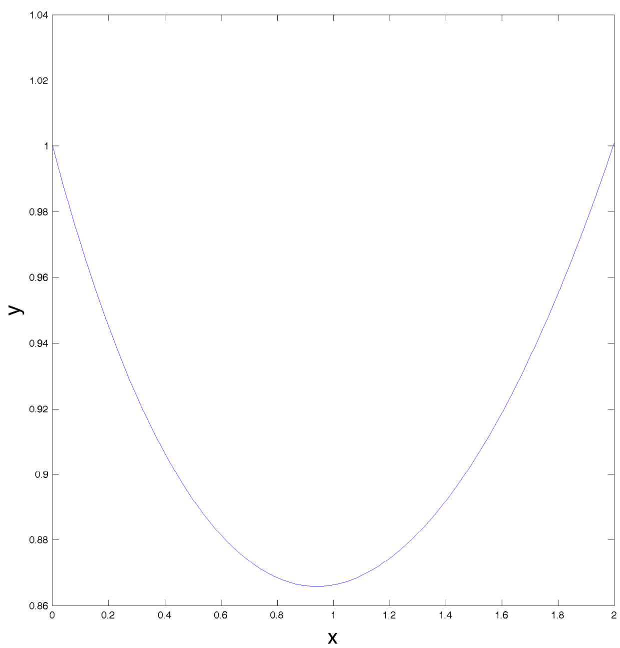

Figure 5: Solution of the BVP with the (regular) shooting method corresponding to $\overline{\theta}$.

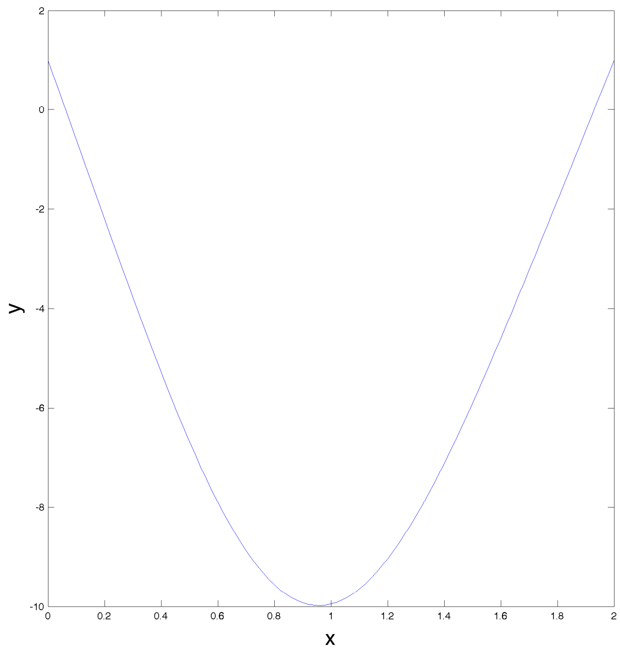

Figure 6: Solution of the BVP with the (regular) shooting method corresponding to $\bar{\bar{\theta}}$.

6. Solve the eigenvalue problem

$$y'' + (2\text{sech}^2 x - \lambda^2)y = 0, x \in (-\infty, \infty), y(|x| \to \infty) \to 0.$$

As is explained in the notes, solve the IVP on the interval $[-R, R]$ for some reasonable large R (say $R = 10$). Choose $y(-R) = \exp[-cR]$ and $y'(-R) = \lambda v(-R)$, where the constant c is of order one (so you may just take $c = 1$). The follow the algorithm outlined in the notes.

Solution: My code and a solution plot follow.

```
% HW07 Problem 06
%
% solve the eigenvalue problem using the shooting method

% IC, BC
R = 10;
y0 = exp(-R);
h = 0.02;
tvec = -R:h:R;

% initialize lambda and f
lambdas = 0.98:.001:1.02;
f = zeros(1,length(lambdas));
% now shoot for all lambda
for i=1:length(lambdas)
    % set lambda
    lambda = lambdas(i);
    yf = lambda*y0;
    % shoot
    shot = andy_ME(@andy_hw07_prb06_ODE,tvec,[y0;yf],h,[lambda]);
    % save shot
```

```
    f(i) = shot(1,end);
    if i==21
        figure; plot(1:length(shot(1,:)),shot(1,:));
    end
end

% find where f crosses 0
for i=1:length(f)-1
    if f(i+1)*f(i) < 0
        disp(i);
        lambdastar = lambdas(i);
        disp(lambdastar);
    end
end


% soln = shot;
% plot(tvec,soln(1,:));
% xlabel('x','FontSize',20);
% ylabel('y','FontSize',20);
% set(gcf, 'units', 'inches', 'position', [1 1 10 10])
% set(gcf,'PaperPositionMode','auto')
% print('-depsc2','-zbuffer','-r200',sprintf('andy_hw07_prb06_%02g.eps',j))
% system(sprintf('epstopdf andy_hw07_prb06_%02g.eps; \\rm andy_hw07_prb06_%02g.eps',j)
    );
```

Appendix 1: ODE Functions

```
function dy = andy_hw07_prb01_ODE(t,y,params)
% [y';v']
% where y' = v, v' = -xv+3y-3x

dy = [y(2);-t*y(2)+3*y(1)-3*t];
```

```
function dy = andy_hw07_prb01_ODEh(t,y,params)
% [y';v']
% where y' = v, v' = -xv+3y

dy = [y(2);-t*y(2)+3*y(1)];
```

```
function dy = andy_hw07_prb02_ODE(t,y,params)
% [y1';y2';y3']
% where y1 = y, y2 = y', y3 = y''

dy = [y(2);y(3);(y(1)-t*y(2)-3+log(t))/t^3];
```

```
function dy = andy_hw07_prb02_ODEh(t,y,params)
% [y1';y2';y3']
% where y1 = y, y2 = y', y3 = y''

dy = [y(2);y(3);(y(1)-t*y(2))/t^3];
```

```
function dy = andy_hw07_prb04_ODE(t,y,params)
% [y';v']
% where y' = v, v' = -xv+3y-3x

dy = [y(2);30^2*(y(1)-1+2*t)];
```

```
function dy = andy_hw07_prb04_ODEh(t,y,params)
% [y';v']
% where y' = v, v' = -xv+3y

dy = [y(2);30^2*(y(1))];
```

```
function dy = andy_hw07_prb05_ODE(t,y,params)
% [y';v']
% where y' = v, v' = -xv+3y-3x

dy = [y(2);y(1)^2/(2+t)];
```

```
function dy = andy_hw07_prb06_ODE(t,y,params)
% [y';v']
% where y' = v, v' = (lambda^2 - 2sech^2 (x))y

dy = [y(2);y(1)*(params(1)^2-2*sech(t)^2)];
```

Appendix 2: Numerical Methods

```
function yvec = andy_ME(func,tspan,y0,h,params)
% modified Euler

yvec = [];
yvec = [yvec y0];
t = tspan(1);
for i=2:length(tspan)
    k1 = func(t,yvec(:,i-1),params);
    k2 = func(t+h,yvec(:,i-1)+h*k1,params);
    yvec = [yvec yvec(:,i-1)+h/2*(k1+k2)];
    t=t+h;
end
```

```
function yvec = andy_MEr(func,tspan,y0,h,params)
% modified Euler

yvec = [];
yvec = [yvec y0];
t = tspan(end);
for i=2:length(tspan)
    k1 = func(t,yvec(:,i-1),params);
    k2 = func(t-h,yvec(:,i-1)-h*k1,params);
    yvec = [yvec yvec(:,i-1)-h/2*(k1+k2)];
    t=t-h;
end
% IMPORTANT
yvec=fliplr(yvec);
```