```
>> prob1
------------ 11.1.4 a -------------
          t           y      y true
 0.00000000  1.00000000  1.00000000
 0.15707963  1.05248506  1.05248562
 0.31415927  1.07905470  1.07905555
 0.47123890  1.07905469  1.07905555
 0.62831853  1.05248505  1.05248562
 0.78539816  1.00000000  1.00000000
------------ 11.1.4 b -------------
          t           y      y true
 0.00000000  0.00000000  0.00000000
 0.15707963 -0.06061198 -0.06062540
 0.31415927 -0.09117479 -0.09119581
 0.47123890 -0.08959214 -0.08961338
 0.62831853 -0.05748564 -0.05749950
 0.78539816 -0.00000000 -0.00000000
>> prob3
------------------ 11.3.2 a ------------------
          t           y      y true
 0.00000000  1.00000000  1.00000000
 0.15707963  1.05260081  1.05248562
 0.31415927  1.07922974  1.07905555
 0.47123890  1.07922974  1.07905555
 0.62831853  1.05260081  1.05248562
 0.78539816  1.00000000  1.00000000
------------------ 11.3.4 b ------------------
          t           y      y true
 0.00000000  0.00000000  0.00000000
 0.15707963 -0.06141845 -0.06062540
 0.31415927 -0.09240491 -0.09119581
 0.47123890 -0.09080499 -0.08961338
 0.62831853 -0.05825827 -0.05749950
 0.78539816  0.00000000 -0.00000000
>> prob2
------------------ 11.2.4 a ------------------
it=  0, z=-0.1666667, phi=  7.4895e-02
it=  1, z=-0.2503499, phi= -3.1564e-04
it=  2, z=-0.2500003, phi= -6.3594e-09
  t           y      y true
1.0  0.50000000  0.50000000
1.1  0.47619050  0.47619048
1.2  0.45454550  0.45454545
1.3  0.43478266  0.43478261
1.4  0.41666671  0.41666667
1.5  0.40000004  0.40000000
1.6  0.38461542  0.38461538
1.7  0.37037040  0.37037037
1.8  0.35714287  0.35714286
1.9  0.34482759  0.34482759
2.0  0.33333333  0.33333333
------------------ 11.2.4 b ------------------
it=  0, z= 0.5000000, phi=  1.5795e+02
it=  1, z= 0.4532380, phi=  6.5615e+01
it=  2, z= 0.3924824, phi=  2.7730e+01
it=  3, z= 0.3109207, phi=  1.1655e+01
it=  4, z= 0.2047668, phi=  4.5499e+00
it=  5, z= 0.0913159, phi=  1.4008e+00
it=  6, z= 0.0182313, phi=  2.3299e-01
it=  7, z= 0.0007496, phi=  8.8524e-03
it=  8, z= 0.0000319, phi=  1.3755e-05
  t           y      y true
1.0  2.00000000  2.00000000
1.1  2.00910296  2.00909091
1.2  2.03335064  2.03333333
1.3  2.06924973  2.06923077
1.4  2.11430448  2.11428571
1.5  2.16668438  2.16666667
1.6  2.22501639  2.22500000
```



```
1.7  2.28825041  2.28823529
1.8  2.35556967  2.35555556
1.9  2.42632936  2.42631579
2.0  2.50001376  2.50000000
>> prob4
------------------ 11.4.4 a ------------------
it=0, norm(F)=4.414365e-01
it=1, norm(F)=1.181519e-03
it=2, norm(F)=5.957781e-09
  t           y      y true
1.0  0.50000000  0.50000000
1.1  0.47619724  0.47619048
1.2  0.45455634  0.45454545
1.3  0.43479561  0.43478261
1.4  0.41668027  0.41666667
1.5  0.40001304  0.40000000
1.6  0.38462696  0.38461538
1.7  0.37037978  0.37037037
1.8  0.35714955  0.35714286
1.9  0.34483111  0.34482759
2.0  0.33333333  0.33333333
------------------ 11.4.4 b ------------------
it=0, norm(F)=1.041378e+01
it=1, norm(F)=3.781993e+00
it=2, norm(F)=5.202054e-02
it=3, norm(F)=1.235002e-05
  t           y      y true
1.0  2.00000000  2.00000000
1.1  2.00923194  2.00909091
1.2  2.03351586  2.03333333
1.3  2.06940818  2.06923077
1.4  2.11443873  2.11428571
1.5  2.16678973  2.16666667
1.6  2.22509387  2.22500000
1.7  2.28830301  2.28823529
1.8  2.35560014  2.35555556
1.9  2.42633877  2.42631579
2.0  2.50000000  2.50000000
>>
```

Feb 25, 11 4:01 **prob1.m** Page 1/1

```
% HW 3 Problem 1

% 11.1.4 a
f = @(t,y,yp) -y;
a=0; b=pi/4; alpha=1; beta=1;
ytrue=@(t) cos(t) + (sqrt(2)-1)*sin(t);
fprintf('------------ 11.1.4 a ------------\n');
n=5; h=(b-a)/n;
y = linshoot(a,b,n,alpha,beta,f);
fprintf('%11s %11s %11s\n','t','y','y true');
for i=1:n+1,
 ti=a+(i-1)*h;
 fprintf('%11.8f %11.8f %11.8f\n',ti,y(i),ytrue(ti));
end;

% 11.1.2 b
f = @(t,y,yp) -4*y+cos(t);
a=0; b=pi/4; alpha=0; beta=0;
ytrue=@(t) -cos(2*t)/3 -(sqrt(2)/6)*sin(2*t) + (1/3)*cos(t);
fprintf('------------ 11.1.4 b ------------\n');
n=5; h=(b-a)/n;
y = linshoot(a,b,n,alpha,beta,f);
fprintf('%11s %11s %11s\n','t','y','y true');
for i=1:n+1,
 ti=a+(i-1)*h;
 fprintf('%11.8f %11.8f %11.8f\n',ti,y(i),ytrue(ti));
end;
```

Feb 25, 11 4:20 **prob2.m** Page 1/1

```
% Problem 2

tol = 1e-4; maxit=100;

% 11.2.4 a
f   = @(t,y,yp) y^3-y*yp;
fy  = @(t,y,yp) 3*y^2 - yp;
fyp = @(t,y,yp) -y;
ytrue = @(t) 1/(t+1);
a=1; b=2; alpha=1/2; beta=1/3;
fprintf('------------------ 11.2.4 a ------------------\n');
n=10; h=(b-a)/n;
y = nonlinshoot(a,b,n,alpha,beta,f,fy,fyp,tol,maxit);
fprintf('%3s %11s %11s\n','t','y','y true');
for i=1:n+1,
 ti=a+(i-1)*h;
 fprintf('%3.1f %11.8f %11.8f\n',ti,y(i),ytrue(ti));
end;

% 11.2.4 b
f   = @(t,y,yp) 2*y^3 -6*y -2*t^3;
fy  = @(t,y,yp) 6*y^2 -6;
fyp = @(t,y,yp) 0*t;
ytrue = @(t) t + 1/t;
a=1; b=2; alpha=2; beta=5/2;
fprintf('------------------ 11.2.4 b ------------------\n');
n=10; h=(b-a)/n;
y = nonlinshoot(a,b,n,alpha,beta,f,fy,fyp,tol,maxit);
fprintf('%3s %11s %11s\n','t','y','y true');
for i=1:n+1,
 ti=a+(i-1)*h;
 fprintf('%3.1f %11.8f %11.8f\n',ti,y(i),ytrue(ti));
end;
```

Feb 25, 11 4:00 **prob3.m** Page 1/1

```
% HW 3 Problem 3

% 11.3.4 a
a=0; b=pi/4; alpha=1; beta=1;
p = @(t) zeros(size(t)); q = @(t) -ones(size(t)); r = @(t) zeros(size(t));
ytrue=@(t) cos(t) + (sqrt(2)-1)*sin(t);
n=4; h=(b-a)/(n+1);
y = linfd(a,b,n,alpha,beta,p,q,r);
fprintf('------------------ 11.3.2 a ------------------\n');
fprintf('%11s %11s %11s\n','t','y','y true');
for i=1:n+2,
 ti=a+(i-1)*h;
 fprintf('%11.8f %11.8f %11.8f\n',ti,y(i),ytrue(ti));
end;

% 11.3.4.b
a=0; b=pi/4; alpha=0; beta=0;
p = @(t) zeros(size(t)); q = @(t) -4*ones(size(t)); r = @(t) cos(t);
ytrue=@(t) -cos(2*t)/3 -(sqrt(2)/6)*sin(2*t) + (1/3)*cos(t);
n=4; h=(b-a)/(n+1);
y = linfd(a,b,n,alpha,beta,p,q,r);
fprintf('------------------ 11.3.4 b ------------------\n');
fprintf('%11s %11s %11s\n','t','y','y true');
for i=1:n+2,
 ti=a+(i-1)*h;
 fprintf('%11.8f %11.8f %11.8f\n',ti,y(i),ytrue(ti));
end;
```

Feb 25, 11 4:20 **prob4.m** Page 1/1

```
% Problem 4

tol = 1e-4; maxit=100;

% 11.4.4 a
f   = @(t,y,yp) y^3-y*yp;
fy  = @(t,y,yp) 3*y^2 - yp;
fyp = @(t,y,yp) -y;
ytrue = @(t) 1/(t+1);
a=1; b=2; alpha=1/2; beta=1/3;
fprintf('------------------ 11.4.4 a ------------------\n');
n=9; h=(b-a)/(n+1);
y = nonlinfd(a,b,n,alpha,beta,f,fy,fyp,tol,maxit);
fprintf('%3s %11s %11s\n','t','y','y true');
for i=1:n+2,
 ti=a+(i-1)*h;
 fprintf('%3.1f %11.8f %11.8f\n',ti,y(i),ytrue(ti));
end;

% 11.4.4 b
f   = @(t,y,yp) 2*y^3 -6*y -2*t^3;
fy  = @(t,y,yp) 6*y^2 -6;
fyp = @(t,y,yp) 0*t;
ytrue = @(t) t + 1/t;
a=1; b=2; alpha=2; beta=5/2;
fprintf('------------------ 11.4.4 b ------------------\n');
n=9; h=(b-a)/(n+1);
y = nonlinfd(a,b,n,alpha,beta,f,fy,fyp,tol,maxit);
fprintf('%3s %11s %11s\n','t','y','y true');
for i=1:n+2,
 ti=a+(i-1)*h;
 fprintf('%3.1f %11.8f %11.8f\n',ti,y(i),ytrue(ti));
end;
```

```
% Linear shooting method
function y = linshoot(a,b,n,alpha,beta,f)
% notes method
F = @(t,y) [y(3);y(4);f(t,y(1),y(3));f(t,y(2),y(4))];
y0 = [alpha;alpha;0;1];
yt = rk4(a,b,n,y0,F);
lambda = (beta-yt(2,n+1))/(yt(1,n+1)-yt(2,n+1));
y =lambda*yt(1,:) + (1-lambda)*yt(2,:);
```



```
% non linear shooting method (uses Newton's method and variational
% equation)
function y = nonlinshoot(a,b,n,alpha,beta,f,fy,fyp,tol,maxit)

% initial guess = slope of line satisfying b.c.
z = (beta-alpha)/(b-a);


% setup system of equations to solve for DE and variational
% equation at same time. Our vector of solutions contains
% y,y',v,v'
F = @(t,y) [ y(2)
             f(t,y(1),y(2))
             y(4)
             fy(t,y(1),y(2))*y(3) + fyp(t,y(1),y(2))*y(4) ];
% solve once to check tolerance
y0 = [alpha;z;0;1]; uy = rk4(a,b,n,y0,F); y=uy(1,:);
nit=0;
fprintf('it=%3d, z=%10.7f, phi=%12.4e\n',nit,z,y(end)-beta);
while (abs(y(end)-beta)>tol && nit<=maxit)
 % update z
 z = z - (y(end)-beta) / uy(3,end);
 % solve for new value of z
 y0 = [alpha;z;0;1]; uy = rk4(a,b,n,y0,F); y=uy(1,:);
 nit=nit+1;
 fprintf('it=%3d, z=%10.7f, phi=%12.4e\n',nit,z,y(end)-beta);
end;
if (nit>maxit)
 fprintf('max number of iterations %d reached and error = %d\n',...
         maxit,(y(end)-beta));
end;
```

```
% function y = linfd(a,b,n,alpha,beta,p,q,r)
% Solves the linear BVP
%  y''(x) = p(x) y' + q(x) y(x) + r(x)
%  y(a)=alpha;  y(b)=beta
function y = linfd(a,b,n,alpha,beta,p,q,r)
% setup grid
h=(b-a)/(n+1);  z = a + (1:n)'*h;
% evaluate at grid points
ps = p(z); qs = q(z); rs=r(z);
% setup matrix of linear system
L = (1/h^2)*spdiags(ones(n,1)*[1,-2,1],-1:1,n,n);
D = (1/2/h)*spdiags([[-ps(2:n);0],[0;ps(1:n-1)]],[-1,1],n,n);
Q = spdiags(qs,0,n,n);
A = L-D-Q;
% setup RHS of linear system
b = rs;
b(1) = b(1) - alpha/h^2 - ps(1)*alpha/2/h;
b(n) = b(n) - beta/h^2  + ps(n)*beta/2/h;
y = [alpha;A\b;beta];
```



```
function y = nonlinfd(a,b,n,alpha,beta,f,fy,fyp,tol,maxit)
% setup grid
h=(b-a)/(n+1);  t = a + (0:n+1)'*h;

% initial guess is line satisfying bc
% note: y has size n+2 (simplifies code)
y = alpha + (beta-alpha)*(t-a)/(b-a);

% discrete Laplacian (for later)
L = (1/h^2)*spdiags(ones(n,1)*[1,-2,1],-1:1,n,n);


% evaluate non-linear map F
F=zeros(n,1);
for i=2:n+1,
 F(i-1) = (y(i-1)-2*y(i)+y(i+1))/h^2 ...
      - f(t(i-1),y(i),(y(i+1)-y(i-1))/2/h);
end;

nit=0;
fprintf('it=%d, norm(F)=%e\n',nit,norm(F));
while (norm(F)>tol ^^ nit≤maxit)
 % build Jacobian
 Q  = zeros(n,3);
 % diagonal
 for i=2:n+1,Q(i-1,2) =  fy (t(i),y(i),(y(i+1)-y(i-1))/2/h); end;
 % super diagonal
 for i=2:n,  Q(i  ,3) =  fyp(t(i),y(i),(y(i+1)-y(i-1))/2/h)/2/h;end;
 % sub diagonal
 for i=3:n+1,Q(i-2,1) = -fyp(t(i),y(i),(y(i+1)-y(i-1))/2/h)/2/h;end;
 DF = L - spdiags(Q,-1:1,n,n);

 % we only update part of y that is not a boundary node
 y(2:n+1) = y(2:n+1) - DF\F;

 % evaluate non-linear map F
 F=zeros(n,1);
 for i=2:n+1,
  F(i-1) = (y(i-1)-2*y(i)+y(i+1))/h^2 ...
       - f(t(i),y(i),(y(i+1)-y(i-1))/2/h);
 end;
 nit=nit+1;
 fprintf('it=%d, norm(F)=%e\n',nit,norm(F));
end;
```

```
% function y = rk4(a,b,n,y0,f)
%
%  Runge-Kutta method for systems
%
%    y'  = f(t,y),  t in [a,b]
%    y(a) = y0
%
%  Inputs
%   a,b    time interval
%   n      number of subintervals
%   y0     initial condition (vector of length m)
%   f      function handle defining the problem
%            f = @(t,y) ....
%          where t = time, y = vector of length m
%          and output is a vector of length m
%
% Outputs:
%  y       iterate history (m by n+1 matrix)
function y = rk4(a,b,n,y0,f)
 y=zeros(length(y0),n+1);
 y(:,1) = y0;
 h=(b-a)/n;
 for i=1:n,
  ti=a+(i-1)*h;
  F1 = h*f(ti,y(:,i));
  F2 = h*f(ti+h/2,y(:,i)+F1/2);
  F3 = h*f(ti+h/2,y(:,i)+F2/2);
  F4 = h*f(ti+h  ,y(:,i)+F3);
  y(:,i+1) = y(:,i) + (F1 + 2*F2 + 2*F3 + F4)/6;
 end;
```