HW2 EMA 471 Intermediate Problem Solving for Engineers

Spring 2016
Engineering Mechanics Department
University of Wisconsin, Madison

Instructor: Professor Robert J. Witt

By

Nasser M. Abbasi

December 30, 2019

Contents

0.1 Problem 1

PROBLEM DESCRIPTION

(1) (10 pts) Solve the non-linear boundary value problem:

$$\dddot{y} + 10\ddot{y} - 5y^3 + ty = t^2$$

over the interval, $0 \le t \le 2$, subject to: $y(0) = 0$, $y(2) = -1$, $\ddot{y}(2) = 0$. Use the `bvp4c` utility.

SOLUTION

The first step is to convert the system to state space.

$$y''' + 10y'' - 5y^3 + ty = t^2$$

Let $x_1 = y, x_2 = y', x_3 = y''$, therefore

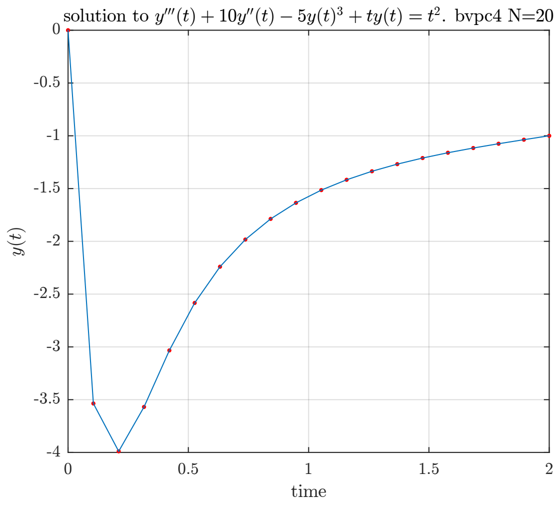
$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= x_3\\ \dot{x}_3 &= -10y'' + 5y^3 - ty + t^2\\ &= -10x_3 + 5x_1^3 - tx_1 + t^2\end{aligned}$$

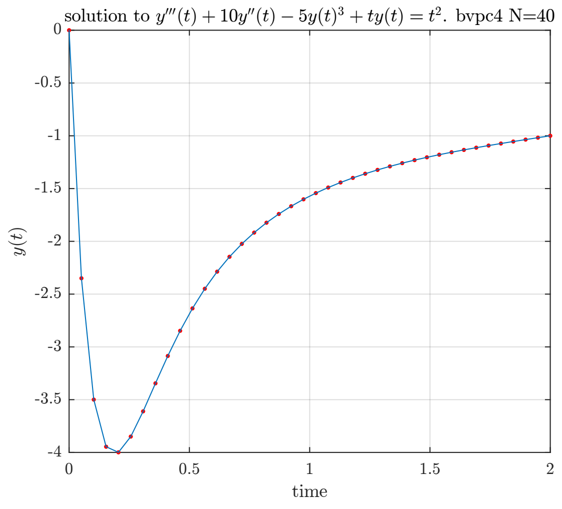
The above $\dot{x}$ vector is what returned back in the RHS call used by bvp4c. The following shows the solution obtained and the code used. One difficulty with this problem was to guess the correct initial solution to use. If the wrong guess was used, then Matlab gives an error

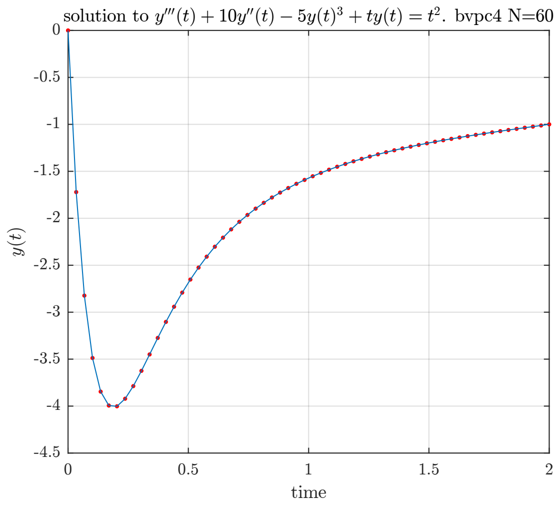
```
Unable to solve the collocation equations -- a singular Jacobian encountered.
```

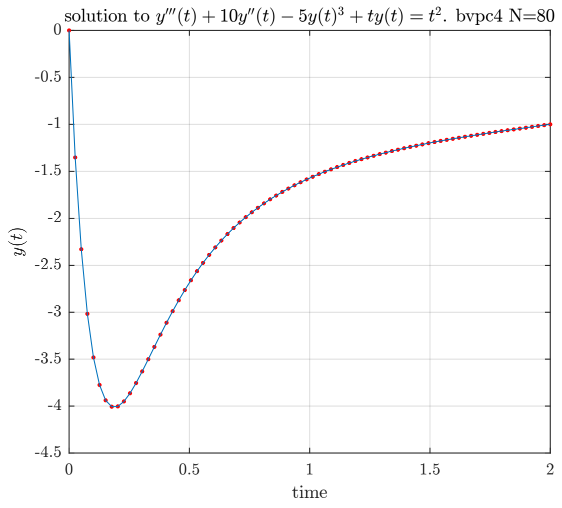
0.1.1 Output

Four plots were generated, with different number of grid points, using $N = 20, 40, 60, 80$ grid point, to see how the solution improves with more grid points added. At $N = 80$, the solution was smooth. Here are the results









0.1.2 Source code

```
function nma_EMA471_HW2_prob_1
% Solves y'''+10 y''-5 y^3 + t y= t^2
% over 0<=t<=2, with y(0)=0,y(2)=-1,y''(2)=0 using bvp4c
%
% see HW2, EMA 471
% by Nasser M. Abbasi
%
clc; close all;

reset(0);
set(groot,'defaulttextinterpreter','Latex');
set(groot, 'defaultAxesTickLabelInterpreter','Latex');
set(groot, 'defaultLegendInterpreter','Latex');

%solve on different grids  20,40,60,80 points
for i=20:20:80
    make_test(i);
```

```
end

end

function make_test(N)
%N is the number of grid points.
%solves the problem using bvp4c

%Important, must use the following guess initial solution [-1 0 0]
%else this error
% Unable to solve the collocation equations -- a singular
%Jacobian encountered. will be generated (Matlab 2015a)

x_bvp4c  = linspace(0,2,N);
solinit1 = bvpinit(x_bvp4c,[-1 0 0]);
sol      = bvp4c(@rhs,@bc,solinit1);

 %evaluate at our grid point, to compare with FDM
y_bvp4c = deval(sol,x_bvp4c);
y_bvp4c = y_bvp4c(1,:)';

figure();
plot(x_bvp4c,y_bvp4c,'r.',x_bvp4c,y_bvp4c);
xlabel('time'); ylabel('$y(t)$');
title(sprintf( ...
 'solution to $y''''''(t)+10 y''''(t)-5 y(t)^3 + t y(t)= t^2$. bvpc4 N=%d',N));

grid;
set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);

end
%-------------------------------------------------------------------------%
function f = rhs(t,x)
%This function sets up the RHS of the state space setup
%for this problem. similar to ode45 RHS

x1 = x(2);
x2 = x(3);
x3 = -10*x(3)+5*x(1)^3-t*x(1)+t^2;
f = [ x1
      x2
      x3 ];
end
%-------------------------------------------------------------------------%
function res = bc(ya,yb)
%This sets up the boundary conditions vector
res = [ ya(1)
        yb(1)+1
        yb(3)
      ];
end
```

0.2 Problem 2

PROBLEM DESCRIPTION

(2) (15 pts) Consider the linear equation:

$$y'''' - y' = e^x$$

Solve this over the interval, $0 \le x \le 1$, subject to: $y(0)=0,\quad y'(0)=-1,\quad y(1)=1,\quad y'(1)=0$. As this is a linear equation, you can employ a finite-difference approximation to compare to the analytical solution as well as the solution generated by `bvp4c`.

SOLUTION

The first step is to convert the system to state space. (I used t below as the independent variable, instead of x as given in the problem statement, to reduce confusion with the x_i used for state space setup).

$$y^{(4)} - y' = e^t$$

Let $x_1 = y, x_2 = y', x_3 = y'', x_4 = y'''$, therefore

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= x_3\\ \dot{x}_3 &= x_4\\ \dot{x}_4 &= y' + e^t\\ &= x_1 + e^t\end{aligned}$$

The above $\dot{x}$ vector is what returned back in the RHS call used by bvp4c. Now we will solve it analytically in order to compare the solutions. The homogenous ODE is $y'''' - y' = 0$, hence the characteristic equation is $\lambda^4 - \lambda = 0$, which has solutions

$$\begin{aligned}\lambda &= 0\\ \lambda &= 1^{\frac{1}{3}}\\ &= \left\{1, -\frac{1}{2} - \frac{\sqrt{3}}{2}i, -\frac{1}{2} + \frac{\sqrt{3}}{2}i\right\}\end{aligned}$$

Therefore the homogenous solution is

$$\begin{aligned}y_h &= Ae^{\lambda_1} + Be^{\lambda_2} + Ce^{\lambda_3} + De^{\lambda_4}\\ &= A + Be^t + Ce^{\left(-\frac{1}{2}-\frac{\sqrt{3}}{2}i\right)t} + De^{\left(-\frac{1}{2}+\frac{\sqrt{3}}{2}i\right)t}\end{aligned}$$

Since the homogenous solution contains e^t solution, and the forcing function is also e^t, we can't guess e^t as particular solution. We try $y_p = cte^t$. Hence

$$\begin{aligned}y_p' &= \left(cte^t + ce^t\right)\\ y_p'' &= \left(cte^t + ce^t\right) + ce^t\\ y_p''' &= \left(\left(cte^t + ce^t\right) + ce^t\right) + ce^t\\ y_p'''' &= \left(\left(\left(cte^t + ce^t\right) + ce^t\right) + ce^t\right) + ce^t\end{aligned}$$

Substituting this in the original ODE we obtain

$$\begin{aligned}\left(\left(\left(cte^t+ce^t\right)+ce^t\right)+ce^t\right)+ce^t-\left(cte^t+ce^t\right)&=e^t\\ 3ce^t&=e^t\end{aligned}$$

Hence $3c=1$ or $c=\frac{1}{3}$, therefore $y_p=\frac{1}{3}te^t$ and the full solution is

$$\begin{aligned}y&=y_h+y_p\\ &=A+Be^t+Ce^{\left(-\frac{1}{2}-\frac{3}{2}i\right)t}+De^{\left(-\frac{1}{2}+\frac{3}{2}i\right)t}+\frac{1}{3}te^t\\ &=A+Be^t+Ce^{\frac{-1}{2}t}e^{\frac{-3}{2}it}+De^{\frac{-1}{2}t}e^{\frac{3}{2}it}+\frac{1}{3}te^t\\ &=A+Be^t+e^{\frac{-1}{2}t}\left(C\cos\left(\frac{\sqrt{3}}{2}t\right)+D\sin\left(\frac{\sqrt{3}}{2}t\right)\right)+\frac{1}{3}te^t\end{aligned}\tag{A}$$

Now we find the constants from initial and boundary conditions. At $y(0)=0$ hence

$$0=A+B+C\tag{1}$$

From $y(1)=1$ we obtain

$$1=A+Be+e^{\frac{-1}{2}}\left(C\cos\left(\frac{\sqrt{3}}{2}\right)+D\sin\left(\frac{\sqrt{3}}{2}\right)\right)+\frac{1}{3}e\tag{2}$$

To apply the other conditions, we need $y'(t)$. Taking derivative of (A) gives

$$y'=Be^t+e^{\frac{-1}{2}t}\left(C\cos\left(\frac{\sqrt{3}}{2}t\right)+D\sin\left(\frac{\sqrt{3}}{2}t\right)\right)+e^{\frac{-1}{2}t}\left(-C\frac{\sqrt{3}}{2}\sin\left(\frac{\sqrt{3}}{2}t\right)+D\frac{\sqrt{3}}{2}\cos\left(\frac{\sqrt{3}}{2}t\right)\right)+\frac{1}{3}e^t+\frac{1}{3}te^t$$

Using $y'(0)=1$ gives

$$0=B+C+D\frac{\sqrt{3}}{2}+\frac{1}{3}\tag{3}$$

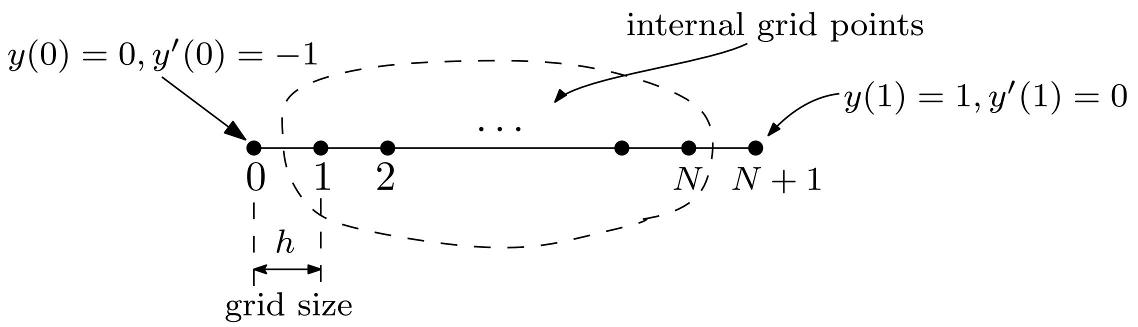
And from $y'(1)=0$ we find

$$0=Be+e^{\frac{-1}{2}}\left(C\cos\left(\frac{\sqrt{3}}{2}\right)+D\sin\left(\frac{\sqrt{3}}{2}\right)\right)+e^{\frac{-1}{2}}\left(-C\frac{\sqrt{3}}{2}\sin\left(\frac{\sqrt{3}}{2}\right)+D\frac{\sqrt{3}}{2}\cos\left(\frac{\sqrt{3}}{2}\right)\right)+\frac{2}{3}e\tag{4}$$

Now we solve Eq (1,2,3,4) for A,B,C,D. With the help of the computer, the above were now solved and the coefficients substituted back into (A) to give the analytical solution below

$$y(t)=\frac{1}{3}te^t-3.956e^t-16.1397e^{\frac{-1}{2}t}\cos\left(\frac{\sqrt{3}}{2}t\right)-6.2898e^{\frac{-1}{2}t}\sin\left(\frac{\sqrt{3}}{2}t\right)+20.096$$

Now that we have the analytical solution, we now need to find a solution using finite differences as well as per problem statement. The first step is to set up the grid. The following diagram shows the grid used



N grid points. $N-2$ internal grid points

Total of N grid points is used. Since the solution is known at $i=0$ and $i=N-1$, the solution at the remaining only $N-2$ points needs to be determined using finite difference scheme. We will now derive the FD equations for grid points $i=1,2,3$. From grid point $i=3$ to $i=N$, the same pattern repeats, and the matrix $A_{N\times N}$ will be filled using an iteration process as shown below.

The differential equation

$$y''''(x)-y'(x)=e^t$$

In finite differences form is

$$\frac{y_{i-2}-4y_{i-1}+6y_i-4y_{i+1}+y_{i+2}}{h^4}-\frac{y_{i+1}-y_{i-1}}{2h}=e^{x_i}$$

Where we used centered difference with $O\left(h^2\right)$ local truncation error for the approximation of $y'(x)$ and 5 points centered difference for the approximation of $y''''(x)$

At $i=1$

$$\frac{y_{-1}-4y_0+6y_1-4y_2+y_3}{h^4}-\frac{y_2-y_0}{2h}=e^{x_1}$$

$$y_{-1}-4y_0+6y_1-4y_2+y_3-\frac{1}{2}h^3\left(y_2-y_0\right)=h^4e^{x_1}$$

To find y_{-1}, since $y'(0)$ is known, using $y'(0)=y_0'=\frac{y_1-y_{-1}}{2h}$ given $y_{-1}=y_1-2hy_0'$ and the above becomes

$$\left(y_1-2hy_0'\right)-4y_0+6y_1-4y_2+y_3-\frac{1}{2}h^3\left(y_2-y_0\right)=h^4e^{x_1}$$

$$y_0\left(-4+\frac{1}{2}h^3\right)+7y_1+y_2\left(-4-\frac{1}{2}h^3\right)+y_3=h^4e^{x_1}+2hy_0'$$

$$7y_1+y_2\left(-4-\frac{1}{2}h^3\right)+y_3=h^4e^{x_1}+2hy_0'+y_0\left(4-\frac{1}{2}h^3\right)$$

At $i=2$

$$\frac{y_0-4y_1+6y_2-4y_3+y_4}{h^4}-\frac{y_3-y_1}{2h}=e^{x_i}$$

$$y_0-4y_1+6y_2-4y_3+y_4-\frac{1}{2}h^3\left(y_3-y_1\right)=h^4e^{x_2}$$

$$y_1\left(-4+\frac{1}{2}h^3\right)+6y_2+y_3\left(-4-\frac{1}{2}h^3\right)+y_4=h^4e^{x_2}-y_0$$

At $i = 3$

$$\frac{y_1 - 4y_2 + 6y_3 - 4y_4 + y_5}{h^4} - \frac{y_4 - y_2}{2h} = e^{x_3}$$

$$y_1 - 4y_2 + 6y_3 - 4y_4 + y_5 - \frac{1}{2}h^3\left(y_4 - y_2\right) = h^4 e^{x_3}$$

$$y_1 + y_2\left(-4 + \frac{1}{2}h^3\right) + 6y_3 + y_4\left(-4 - \frac{1}{2}h^3\right) + y_5 = h^4 e^{x_3}$$

The rest will now be repeated with i being increased by one for each new row, and shifted to the right by one for each row. For example for $i = 4$

$$\frac{y_2 - 4y_3 + 6y_4 - 4y_5 + y_6}{h^4} - \frac{y_5 - y_3}{2h} = e^{x_4}$$

$$y_2 - 4y_3 + 6y_4 - 4y_5 + y_6 - \frac{1}{2}h^3\left(y_5 - y_3\right) = h^4 e^{x_4}$$

$$y_2 + y_3\left(-4 + \frac{1}{2}h^3\right) + 6y_4 + y_5\left(-4 + \frac{1}{2}h^3\right) + y_6 = h^4 e^{x_4}$$

Therefore, the $Ax = b$ system to solve is

$$\boxed{\begin{bmatrix} 7 & \left(-4 - \frac{1}{2}h^3\right) & 1 & 0 & \cdots & 0 & 0 & 0 \\ \left(-4 + \frac{1}{2}h^3\right) & 6 & \left(-4 - \frac{1}{2}h^3\right) & 1 & 0 & \cdots & 0 & 0 \\ 1 & \left(-4 + \frac{1}{2}h^3\right) & 6 & \left(-4 - \frac{1}{2}h^3\right) & 1 & 0 & \cdots & 0 \\ 0 & 1 & \left(-4 + \frac{1}{2}h^3\right) & 6 & \left(-4 + \frac{1}{2}h^3\right) & 1 & 0 & \cdots \\ 0 & 0 & \left(-4 + \frac{1}{2}h^3\right) & 6 & \left(-4 + \frac{1}{2}h^3\right) & 1 & 0 & \cdots \\ & & & & & & & \\ & & & & & & & \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{N-2} \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} h^4 e^h - 2hy_0' + y_0\left(4 - \frac{1}{2}h^3\right) \\ h^4 e^{2h} - y_0 \\ h^4 e^{3h} \\ h^4 e^{4h} \\ \vdots \\ \\ \\ \end{bmatrix}}$$

Now to fill in the last 3 rows. At $i = N$

$$\frac{y_{N-2} - 4y_{N-1} + 6y_N - 4y_{N+1} + y_{N+2}}{h^4} - \frac{y_{N+1} - y_{N-1}}{2h} = e^{x_i}$$

But we do not know y_{N+2} but since we know $y'(1) = 0$, then using

$$y'(1) = y'_{N+1} = \frac{y_{N+2} - y_N}{2h}$$

Hence $y_{N+2} = 2hy'(1) + y_N$ and the above becomes (by also replacing $y_{N+1} = y(1)$, which is known).

$$\frac{y_{N-2} - 4y_{N-1} + 6y_N - 4y_{N+1} + 2hy'(1) + y_N}{h^4} - \frac{y_{N+1} - y_{N-1}}{2h} = e^{x_N}$$

$$y_{N-2} - 4y_{N-1} + 6y_N - 4y(1) + 2hy'(1) + y_N - \frac{1}{2}h^3\left(y(1) - y_{N-1}\right) = h^4 e^{x_N}$$

$$y_{N-2} + y_{N-1}\left(-4 + \frac{1}{2}h^3\right) + 7y_N = h^4 e^{x_N} - 2hy'(1) + \left(\frac{1}{2}h^3 + 4\right)y(1)$$

At $i = N - 1$

$$\frac{y_{N-3} - 4y_{N-2} + 6y_{N-1} - 4y_N + y_{N+1}}{h^4} - \frac{y_N - y_{N-2}}{2h} = e^{x_{N-1}}$$

$$y_{N-3} - 4y_{N-2} + 6y_{N-1} - 4y_N + y\left(1\right) - \frac{1}{2}h^3\left(y_N - y_{N-2}\right) = h^4 e^{x_{N-1}}$$

$$y_{N-3} + y_{N-2}\left(-4 + \frac{1}{2}h^3\right) + 6y_{N-1} + y_N\left(-4 - \frac{1}{2}h^3\right) = h^4 e^{x_{N-1}} - y\left(1\right)$$

At $i = N - 2$

$$\frac{y_{N-4} - 4y_{N-3} + 6y_{N-2} - 4y_{N-1} + y_N}{h^4} - \frac{y_{N-1} - y_{N-3}}{2h} = e^{x_{N-2}}$$

$$y_{N-4} - 4y_{N-3} + 6y_{N-2} - 4y_{N-1} + y_N - \frac{1}{2}h^3\left(y_{N-1} - y_{N-3}\right) = h^4 e^{x_{N-2}}$$

$$y_{N-4} + y_{N-3}\left(-4 + \frac{1}{2}h^3\right) + 6y_{N-2} + y_{N-1}\left(-4 - \frac{1}{2}h^3\right) + y_N = h^4 e^{x_{N-2}}$$

The $Ax = b$ system becomes

$$\begin{bmatrix}
7 & \left(-4-\frac{1}{2}h^3\right) & 1 & 0 & \cdots & 0 & 0 & 0\\
\left(-4+\frac{1}{2}h^3\right) & 6 & \left(-4-\frac{1}{2}h^3\right) & 1 & 0 & \cdots & 0 & 0\\
1 & \left(-4+\frac{1}{2}h^3\right) & 6 & \left(-4-\frac{1}{2}h^3\right) & 1 & 0 & \cdots & 0\\
0 & 1 & \left(-4+\frac{1}{2}h^3\right) & 6 & \left(-4+\frac{1}{2}h^3\right) & 1 & 0 & \cdots\\
0 & 0 & 1 & \left(-4+\frac{1}{2}h^3\right) & 6 & \left(-4+\frac{1}{2}h^3\right) & 1 & 0\\
\cdots & \cdots & \ddots & \cdots & \ddots & \cdots & \cdots & \cdots\\
0 & \cdots & \left(-4+\frac{1}{2}h^3\right) & 6 & \left(-4+\frac{1}{2}h^3\right) & 1 & 0 & \cdots\\
 & & & 1 & \left(-4+\frac{1}{2}h^3\right) & 6 & \left(-4-\frac{1}{2}h^3\right) & 1\\
0 & 0 & \cdots & 0 & 1 & \left(-4+\frac{1}{2}h^3\right) & 6 & \left(-4-\frac{1}{2}h^3\right)\\
0 & 0 & 0 & \cdots & 0 & 1 & \left(-4+\frac{1}{2}h^3\right) & 7
\end{bmatrix}
\begin{bmatrix} y_1\\ y_2\\ y_3\\ y_4\\ \vdots\\ \vdots\\ \vdots\\ y_{N-2}\\ y_{N-1}\\ y_N \end{bmatrix} =$$

$$\begin{bmatrix}
h^4 e^h - 2hy_0' + y_0\left(4 - \frac{1}{2}h^3\right)\\
h^4 e^{2h} - y\left(0\right)\\
h^4 e^{3h}\\
h^4 e^{4h}\\
\vdots\\
\vdots\\
\vdots\\
h^4 e^{x_{N-2}}\\
h^4 e^{x_{N-1}} - y\left(1\right)\\
h^4 e^{x_N} - 2hy'\left(1\right) + \left(\frac{1}{2}h^3 + 4\right)y\left(1\right)
\end{bmatrix}$$

The system is now solved for x, which is the y_i solution and plotted. The Matlab code is given below.

0.2.1 Results

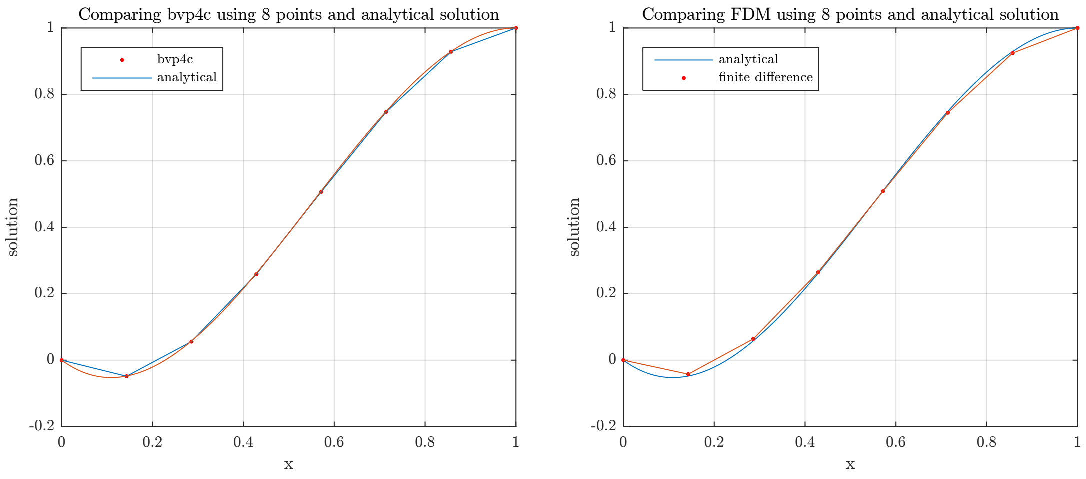
The program `nma_EMA_471_HW2_prob_2.m` generates 4 result for different grid sizes. It uses 8, 15, 30, 100 grid points each time, to compare the result. bvp4c and the finite difference method, both used the same grid size each time. Each time, the result is compare with the analytical solution (which used a much smaller grid than both).

```
....
```

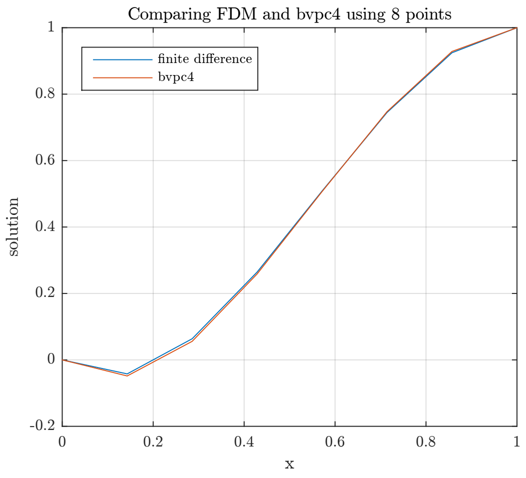
```
x_for_analytic=0:0.001:1;
analytical_solution = get_analytical_solution(x_for_analytic);
make_on_test(8,x_for_analytic,analytical_solution);
make_on_test(15,x_for_analytic,analytical_solution);
make_on_test(30,x_for_analytic,analytical_solution);
make_on_test(100,x_for_analytic,analytical_solution);
....
```

.

At small number of grid points (large h), bvp4c seems to be more accurate at the boundary. Here is side by side showing the result for grid size of 8 points over the whole range.



The finite difference was also plotted against the bvp4c solution. The differences between them show up near where the solution changes most rapidly, around $x = 0.2$ and near the right boundary also. Here is the plot when using 8 grid points

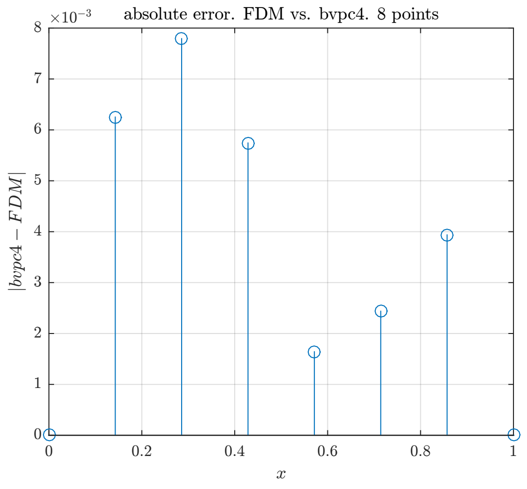


To see more clearly the difference, the absolute difference between bvp4c and finite difference solution was plotted, by plotting $|y_{\text{FDM}} - y_{\text{bvp4c}}|$ at each grid point

```
....
```

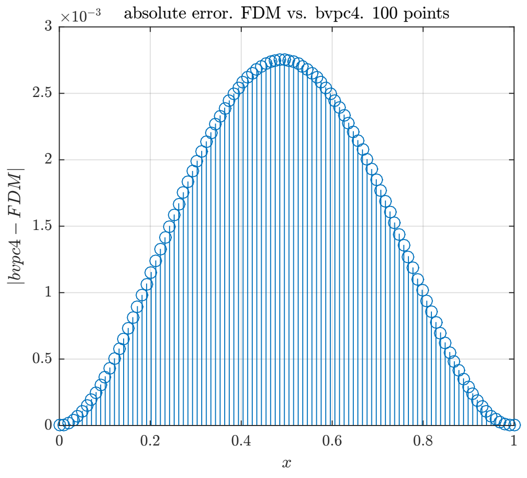
```
figure();
stem(x_bvp4c,abs(y_bvp4c-y_FDM));
...
```

.



As more grid points added, the difference between bvp4c and the analytical solution became smaller. The same for finite difference solution. At 100 grid points, the largest absolute difference between bvp4c and FDM was 0.00275, near the middle of the range. This is compared to the difference being 0.008 when using 8 grid points.

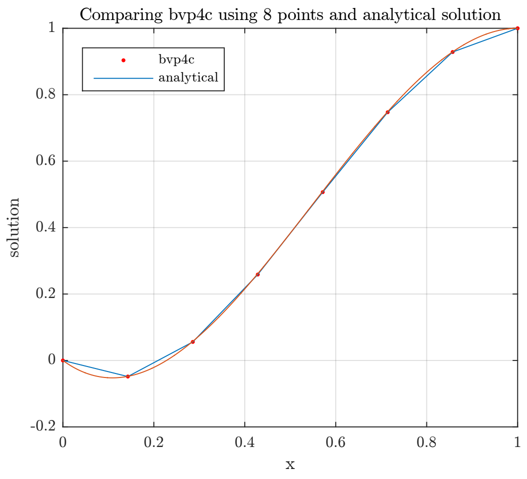
This plot shows the difference at 100 grid points

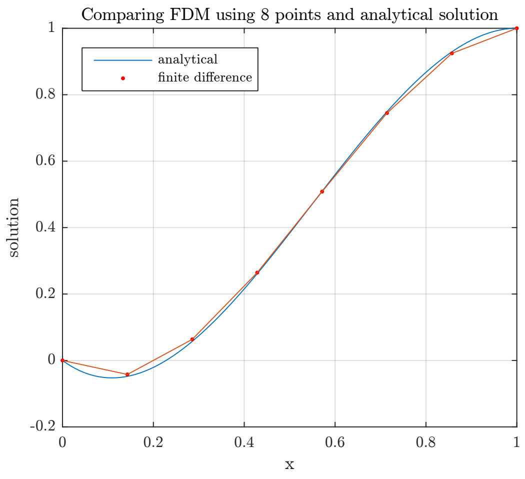


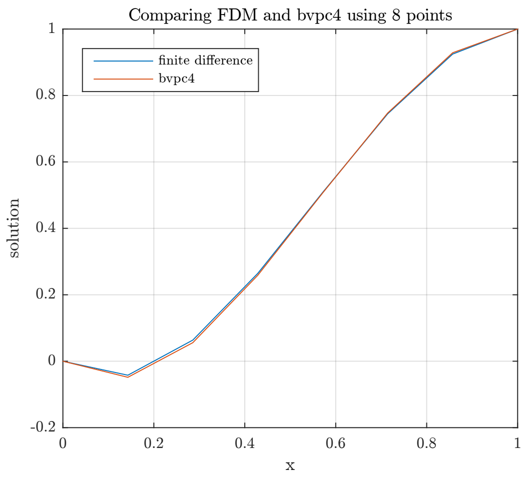
A total of 16 plots generated (4 for each test case) as described above. Below all 16 plots are given, with description title of each plot. Then the source code used to generate these plots is listed.

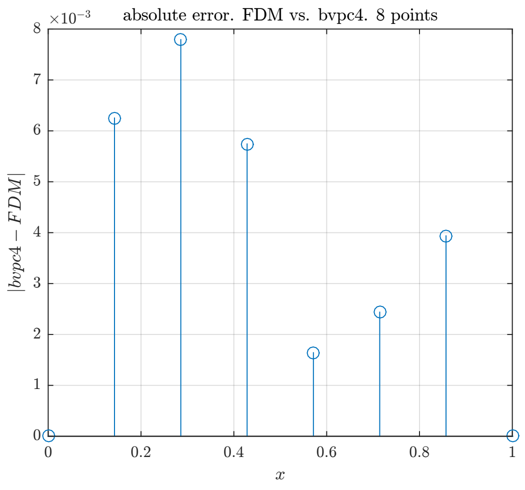
In conclusion: As more grid points added, both bvp4c and FDM approached the analytical solution. There remains difference between bvp4c and FDM in the middle range. Both converged to the same result at the boundaries. This is expected, since the solution there is given from the problem boundary conditions.

8 grid points plots

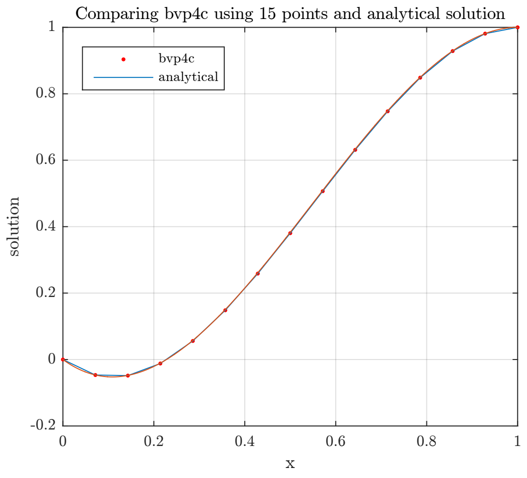


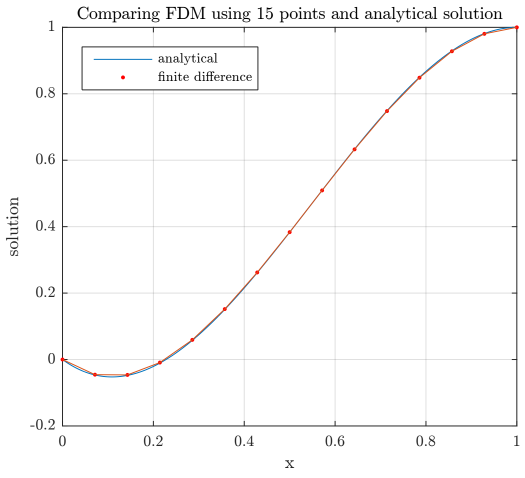


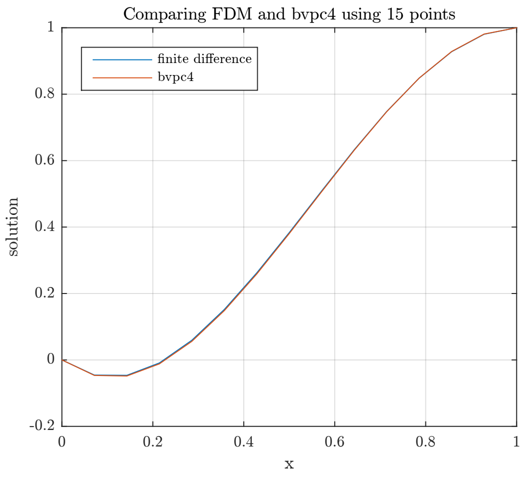


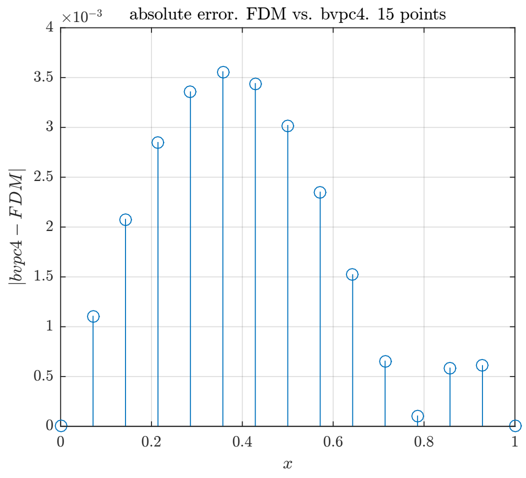


15 grid points plots

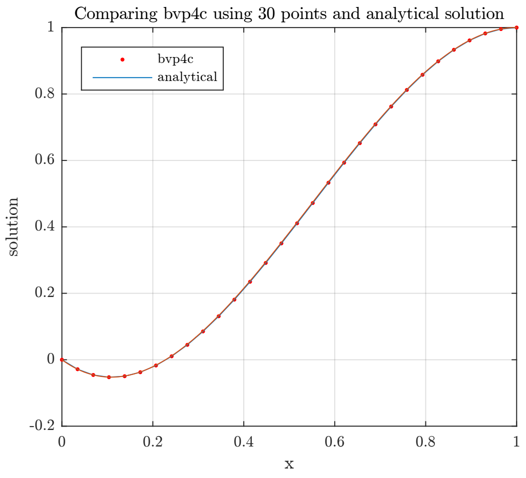


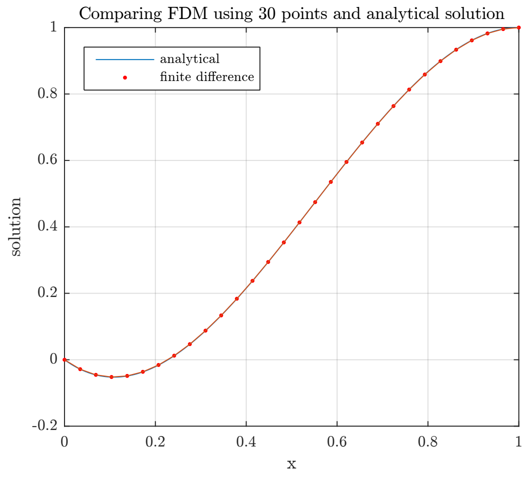


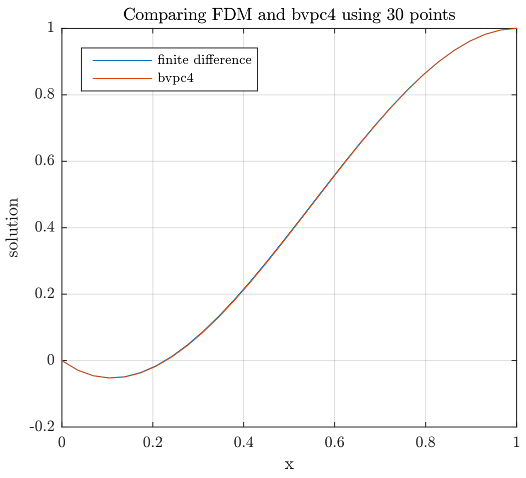


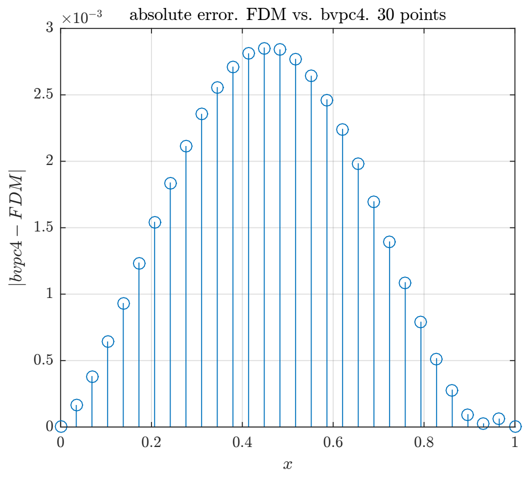


30 grid points plots

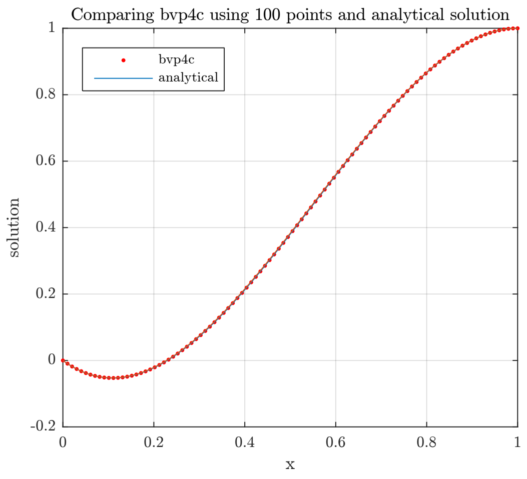


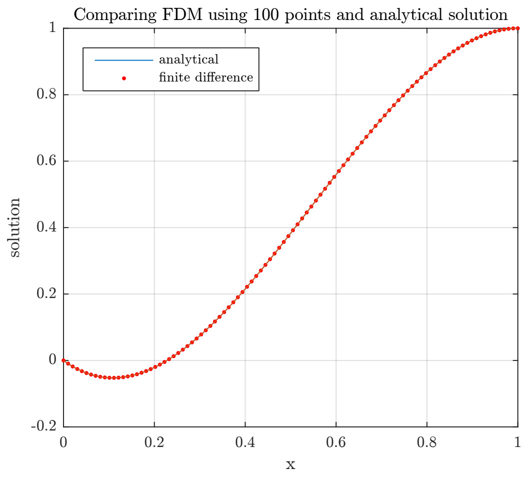


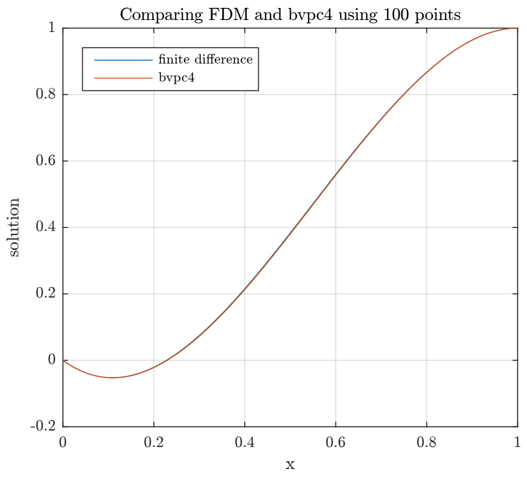


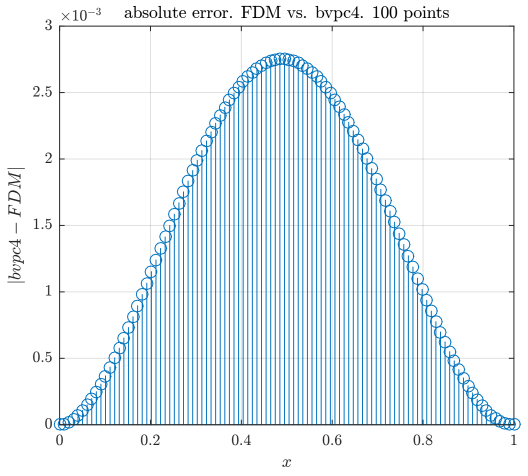


100 grid points plots









0.2.2 Source code

```
function nma_EMA471_HW2_prob_2
%Nasser M. Abbasi

clc; close all;

%generate the analytical solution first, to use to compare
%the numerical results against.
x_for_analytic=0:0.001:1;
analytical_solution = get_analytical_solution(x_for_analytic);

%generate results for different grid sizes
make_one_test(8,x_for_analytic,analytical_solution);
make_one_test(15,x_for_analytic,analytical_solution);
make_one_test(30,x_for_analytic,analytical_solution);
make_one_test(100,x_for_analytic,analytical_solution);
end

%-------------------------------------------------
%This function generates all the plots for bvp4c and FDM
%using specific number of grid points
%
function make_one_test(N,x_anaytic,y_analytical)

%reset for plotting only
reset(0);
set(groot,'defaulttextinterpreter','Latex');
set(groot, 'defaultAxesTickLabelInterpreter','Latex');
set(groot, 'defaultLegendInterpreter','Latex');

x_bvp4c  = linspace(0,1,N);
solinit1 = bvpinit(x_bvp4c,[1 0 0 0]); %use specificed N grid points

%options = bvpset('RelTol',1e-6,'AbsTol',1e-6);  Was not
%needed at home pc! The above is only needed at school Matlab,
%which is 2014a. But not 2015a for some reason. One only need
%to pick the correct initial guess

sol = bvp4c(@rhs,@bc,solinit1);

%extract the x and y solution for plotting. These
%variables are used later in the plots

%evaluate at our grid point, to compare with FDM
y_bvp4c = deval(sol,x_bvp4c);
y_bvp4c = y_bvp4c(1,:)';

%plot bvp4c vs. analytical
figure();
plot(x_bvp4c,y_bvp4c,'r.',x_bvp4c,y_bvp4c);
hold on;
plot(x_anaytic,y_analytical);
xlabel('x'); ylabel('solution');
title(sprintf( ...
   'Comparing bvp4c using %d points and analytical solution',N));
legend('bvp4c','analytical','location','northwest');
grid;
set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);

%Obtain FDM solution. Use the same grid spacing as
%bvp4c (ie. same x)
y0    = 0; yL = 1; yd0 = -1; ydL = 0;
y_FDM = finite_difference_solution(x_bvp4c,y0,yL,yd0,ydL);

%plot finite difference vs. analytical
figure();
plot(x_anaytic,y_analytical);
hold on;
plot(x_bvp4c,y_FDM,'r.',x_bvp4c,y_FDM);
legend('analytical','finite difference','location','northwest');
xlabel('x'); ylabel('solution');
title(sprintf(...
    'Comparing FDM using %d points and analytical solution',N));
grid;
set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);


%plot finite difference vs. bvp4c
figure();
plot(x_bvp4c,y_FDM);
hold on;
plot(x_bvp4c,y_bvp4c);
legend('finite difference','bvpc4','location','northwest');
xlabel('x'); ylabel('solution');
title(sprintf('Comparing FDM and bvpc4 using %d points',N));
grid;
set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);


%plot error between bvp4c and FDM, to see more
%clearly the difference at each grid point.
figure();
stem(x_bvp4c,abs(y_bvp4c-y_FDM));
xlabel('$x$'); ylabel('$\left|bvpc4-FDM\right|$');
title(sprintf('absolute error. FDM vs. bvpc4. %d points',N));
grid;
set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);

end

%---------------------------------------------
%This function generates the Finite difference solution
%The HW report contains the derivation used
%
function y = finite_difference_solution(x,y0,yL,yd0,ydL)

%Allocate A matrix and b vector
h = x(2)-x(1);   %h spacing is the same between each grid point
```

```
N = length(x)-2; %number of internal points.
A = zeros(N,N);
b = zeros(N,1);  %allocate RHS, for Ax=b use

%Start filling the A matrix

%fill in the first 2 rows by hand
A(1,1) = 7;
A(1,2) = -4-1/2*h^3;
A(1,3) = 1;

A(2,1) = -4+1/2*h^3;
A(2,2) = 6;
A(2,3) = -4-1/2*h^3;
A(2,4) = 1;

k=0; %column index, used for the loop below, to fill the rest of A
for i = 3:N-2
     k = k+1;
     A(i,k:k+4) = [1,(-4+1/2*h^3),6,(-4-1/2*h^3),1];
end

%fill in the last 2 rows by hand
k = k+1;
A(N-1,k:k+3) = [1,(-4+1/2*h^3),6,(-4-1/2*h^3)];
k = k+1;
A(N,k:k+2)   = [1,(-4+1/2*h^3),7];

%fill in the b matrix. The first 2 rows by hand
b(1) = h^4*exp(h)+2*h*yd0+y0*(4-1/2*h^3);
b(2) = h^4*exp(2*h)-y0;

%fill the rest of b using loop
for i = 3:N-2
    b(i) = h^4*exp(i*h);
end

%fill the last 2 rows of b
b(N-1) = h^4*exp((N-1)*h)-yL;
b(N)   = h^4*exp(N*h)-2*h*ydL+yL*(4+1/2*h^3);

%now we solve the system.
y=A\b;

%pad in the left and right boundary values. Known values.
y=[y0;y;yL];

end

%---------------------------------------------
%This function returns the analytical solution. Solved in the HW report
function y=get_analytical_solution(t)
y=1/3*t.*exp(t)-3.956115643*exp(t)-16.13974994*exp(-.5*t).*...
                                             cos(.866025404*t)...
    -6.289760819*exp(-.5*t).*sin(.866025404*t)+20.09586558;
end

%-------------------------------------------------------------------------%
%This function used by bvp4c, the RHS. Same as ODE45 RHS function

function f = rhs(t,x)
x1 = x(2);
x2 = x(3);
x3 = x(4);
x4 = x(1)+exp(t);
f = [ x1
    x2
    x3
    x4 ];
end

%-------------------------------------------------------------------------%
%This is the boundary conditions function for bvp4c
function res = bc(ya,yb)
res = [ ya(1)
    ya(2)+1
    yb(1)-1
    yb(2)
    ];
end
```

0.3 Problem 3

PROBLEM DESCRIPTION

(3) (15 pts) One of the topics often discussed in Advanced Mechanics of Materials is the thick rotating cylinder. It is possible to show that the governing equation can be obtained in terms of the radial stress:

$$r\frac{d^2\sigma_r}{dr^2}+3\frac{d\sigma_r}{dr}=-(3+\nu)\rho\omega^2 r$$

with the hoop stress obtained after the fact from:

$$\sigma_\theta = r\frac{d\sigma_r}{dr}+\sigma_r+\rho\omega^2 r^2$$

The second order equation in the radial stress is solved subjected to the constraint that the radial stress equal the negative of the pressure applied at the inner and outer radii. If there is no internal or external pressure, the boundary conditions become:

$$\sigma_r(r_i)=0 \qquad , \qquad \sigma_r(r_o)=0$$

Suppose we have a thick cylinder with inner radius 1 cm, outer radius 10 cm with mass density 1000 kg/m^3. It is subjected to a angular velocity of 700 rad/s. Find the peak radial and hoop stresses for this cylinder. There is an analytical solution to this linear equation, and it also lends itself to a finite difference solution. Compare both of these to the solution generated using `bvp4c`.

SOLUTION

The first step is to determine the analytical solution, in order to compare with the numerical solutions. The ODE to solve is (below, σ is used instead of σ_r to simplify the notation)

$$r\frac{d^2\sigma}{dr^2}+3\frac{d\sigma}{dr}=-(3+v)\,\rho\omega^2 r \tag{1}$$

Since v,ρ,ω are all constants, the above can be written as

$$r\frac{d^2\sigma}{dr^2}+3\frac{d\sigma}{dr}=kr$$

Where $k=-(3+v)\,\rho\omega^2$. To solve the above, we introduce $f=\frac{d\sigma}{dr}$ and it becomes a first order ODE

$$\begin{aligned} r\frac{df}{dr}+3f &= kr \\ \frac{df}{dr}+\frac{3}{r}f &= k \end{aligned} \tag{2}$$

This is separable now, and can be solved for f. Looking at the homogenous ODE first,

$$\begin{aligned} \frac{df}{dr}+\frac{3}{r}f &= 0 \\ \frac{df}{f} &= -\frac{3}{r}dr \end{aligned}$$

Integrating both sides

$$\begin{aligned}\ln f &= -3\ln r + c_1\\ f &= e^{-3\ln r + c_1}\\ &= c_2 e^{-3\ln r}\end{aligned}$$

Hence the homogenous solution is

$$f_h = \frac{c_2}{r^3}$$

Where c_2 is constant of integration. To find the particular solution f_p, and since the RHS of (2) is constant k, then we guess $f_p = crk$, where c is constant. Hence (2) becomes

$$\begin{aligned}ck + \frac{3}{r}rck &= k\\ 4c &= 1\\ c &= \frac{1}{4}\end{aligned}$$

Hence

$$f_p = \frac{r}{4}k$$

And the complete solution is

$$\begin{aligned}f &= f_h + f_p\\ &= \frac{c_2}{r^3} + \frac{r}{4}k\end{aligned}$$

Now that we found $f(r)$, and since $f = \frac{d\sigma}{dr}$ then we have

$$\frac{d\sigma}{dr} = \frac{c_2}{r^3} + \frac{r}{4}k$$

Which is separable. Hence

$$\begin{aligned}d\sigma &= \left(\frac{c_2}{r^3} + \frac{r}{4}k\right)dr\\ \sigma &= \int\left(\frac{c_2}{r^3} + \frac{r}{4}k\right)dr\\ &= \frac{-c_2}{2}\frac{1}{r^2} + \frac{r^2}{8}k + c_3\end{aligned}$$

Hence the analytical solution is

$$\sigma_r(r) = \frac{-C_1}{2}\frac{1}{r^2} + \frac{r^2}{8}k + C_2 \tag{3}$$

Where C_1, C_2 are constants of integration, which we will now find from boundary conditions. At $r = r_i, \sigma_r = 0$ and at $r = r_o, \sigma_r = 0$, hence we obtain two equations

$$\begin{aligned}0 &= \frac{-C_1}{2}\frac{1}{r_i^2} + \frac{r_i^2}{8}k + C_2\\ 0 &= \frac{-C_1}{2}\frac{1}{r_o^2} + \frac{r_o^2}{8}k + C_2\end{aligned}$$

Solving these gives

$$C_1 = \frac{-k}{4} r_i^2 r_o^2$$
$$C_2 = \frac{-k}{8}\left(r_i^2 + r_o^2\right)$$

Therefore (3) becomes

$$\sigma_r(r) = \frac{k}{8}\left(r_i^2 r_o^2 \frac{1}{r^2} + r^2 - \left(r_i^2 + r_o^2\right)\right)$$

But $k = -(3+v)\rho\omega^2$, hence the above becomes

$$\boxed{\sigma_r(r) = \tfrac{(3+v)\rho\omega^2}{8}\left(\left(r_i^2 + r_o^2\right) - r_i^2 r_o^2 \tfrac{1}{r^2} - r^2\right)} \tag{4}$$

Now we will find the analytical solution for the hoop stress

$$\sigma_\theta = r\frac{d\sigma}{dr} + \sigma_r + \rho\omega^2 r^2$$

From (4), we see that

$$\frac{d\sigma}{dr} = \frac{(3+v)\rho\omega^2}{8}\left(\frac{2}{r^3} r_i^2 r_o^2 - 2r\right)$$

Hence

$$\boxed{\sigma_\theta = \tfrac{(3+v)\rho\omega^2}{8}\left(\tfrac{2}{r^2} r_i^2 r_o^2 - 2r^2\right) + \sigma_r + \rho\omega^2 r^2} \tag{5}$$

Now that we have found the analytical solutions, we can implement the numerical solution and compare. We start with bvp4c to verify the analytical solution with, then implement the finite difference method. We need first to convert the ODE to state space.

$$\frac{d^2\sigma}{dr^2} + \frac{3}{r}\frac{d\sigma}{dr} = -(3+v)\rho\omega^2$$

Let $x_1 = \sigma_r$, and let $x_2 = \frac{d\sigma}{dr}$, hence

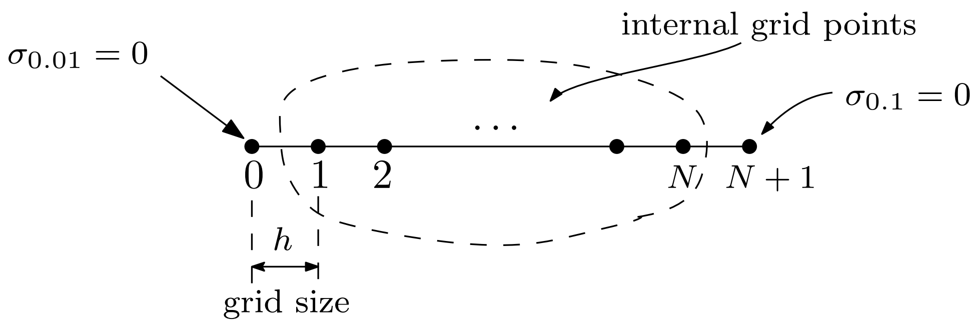
$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -(3+v)\rho\omega^2 - \frac{3}{r}x_2$$

The boundary conditions are (using bvp4c notations) $y_a(1) = 0, y_b(1) = 0$.

0.3.1 Finite difference method

$$\frac{d^2\sigma}{dr^2} + 3\frac{1}{r}\frac{d\sigma}{dr} = -(3+v)\rho\omega^2$$

The following grid is used



N grid points. $N-2$ internal grid points

The grid is only over $r = 0.01 \cdots 0.1$ meters since this is the distance between the internal radius and the outer radius. For $\frac{d^2\sigma}{dr^2}$ we use 3 points centered difference approximation, for $i = 1 \cdots N$ which has $O\left(h^2\right)$ local truncation error

$$\frac{d^2\sigma}{dr^2} = \frac{\sigma_{i+1} - 2\sigma_i + \sigma_{i-1}}{h^2}$$

Where σ_0, σ_{N+1} are given as the boundary conditions (both are zero). For $\frac{d\sigma}{dr}$ we use two points centered difference, which also has $O\left(h^2\right)$ local truncation error

$$\frac{d\sigma}{dr} = \frac{\sigma_{i+1} - \sigma_{i-1}}{2h}$$

Therefore, the ODE becomes

$$\begin{aligned}
r_i \frac{\sigma_{i+1} - 2\sigma_i + \sigma_{i-1}}{h^2} + 3\frac{\sigma_{i+1} - \sigma_{i-1}}{2h} &= -(3+v)\,\rho\omega^2 r_i \\
\sigma_{i+1} - 2\sigma_i + \sigma_{i-1} + \frac{3}{2r_i}h\left(\sigma_{i+1} - \sigma_{i-1}\right) &= -h^2(3+v)\,\rho\omega^2 \\
\sigma_{i-1}\left(1 - \frac{3}{2r_i}h\right) - 2\sigma_i + \sigma_{i+1}\left(1 + \frac{3}{2r_i}h\right) &= -h^2(3+v)\,\rho\omega^2
\end{aligned}$$

Where $r_i = r_i + ih = 0.01 + ih$, since are starting from 0.01, the above becomes

$$\sigma_{i-1}\left(1 - \frac{3}{2\left(r_i + ih\right)}h\right) - 2\sigma_i + \sigma_{i+1}\left(1 + \frac{3}{2\left(r_i + ih\right)}h\right) = -h^2(3+v)\,\rho\omega^2$$

For $i = 1$,

$$\sigma_0\left(1 - \frac{3}{2\left(r_i + h\right)}h\right) - 2\sigma_1 + \sigma_2\left(1 + \frac{3}{2\left(r_i + h\right)}h\right) = -h^2(3+v)\,\rho\omega^2$$

But σ_0 is the boundary conditions, which we move to the RHS, hence

$$-2\sigma_1 + \sigma_2\left(1 + \frac{3}{2\left(r_i + h\right)}h\right) = -h^2(3+v)\,\rho\omega^2 - \sigma_0\left(1 - \frac{3}{2\left(r_i + h\right)}h\right)$$

For $i = 2$

$$\sigma_1\left(1 - \frac{3}{2\left(r_i + 2h\right)}h\right) - 2\sigma_2 + \sigma_3\left(1 + \frac{3}{2\left(r_i + 2h\right)}h\right) = -h^2(3+v)\,\rho\omega^2$$

For $i = 3$

$$\sigma_2\left(1 - \frac{3}{2\left(r_i + 3h\right)}h\right) - 2\sigma_3 + \sigma_4\left(1 + \frac{3}{2\left(r_i + 3h\right)}h\right) = -h^2\left(3 + v\right)\rho\omega^2$$

The same pattern repeats. For $i = N$

$$\sigma_{N-1}\left(1 - \frac{3}{2\left(r_i + Nh\right)}h\right) - 2\sigma_N + \sigma_{N+1}\left(1 + \frac{3}{2\left(r_i + Nh\right)}h\right) = -h^2\left(3 + v\right)\rho\omega^2$$

But σ_{N+1} is given (the right side boundary conditions, hence

$$\sigma_{N-1}\left(1 - \frac{3}{2\left(r_i + Nh\right)}h\right) - 2\sigma_N = -h^2\left(3 + v\right)\rho\omega^2 - \sigma_{N+1}\left(1 + \frac{3}{2\left(r_i + Nh\right)}h\right)$$

And for $i = N - 1$

$$\sigma_{N-2}\left(1 - \frac{3}{2\left(r_i + \left(N-1\right)h\right)}h\right) - 2\sigma_{N-1} + \sigma_N\left(1 + \frac{3}{2\left(r_i + \left(N-1\right)h\right)}h\right) = -h^2\left(3 + v\right)\rho\omega^2$$

Therefore the $Ax = b$ system is below. The A matrix is

$$\begin{bmatrix}
-2 & 1 + \frac{3h}{2(r_i+h)} & 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
1 - \frac{3h}{2(r_i+2h)} & -2 & 1 + \frac{3h}{2(r_i+2h)} & 0 & \cdots & \cdots & \cdots & \cdots \\
0 & 1 - \frac{3h}{2(r_i+3h)} & -2 & 1 + \frac{3h}{2(r_i+3h)} & 0 & \cdots & \cdots & \cdots \\
\vdots & 0 & 0 & 0 & 0 & \ddots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & 0 & 1 - \frac{3h}{2(r_i+(N-1)h)} & -2 & 1 + \frac{3h}{2(r_i+(N-1)h)} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 - \frac{3h}{2(r_i+Nh)} & -2
\end{bmatrix}$$

And the x vector is $\begin{bmatrix}\sigma_1 & \sigma_1 & \cdots & \sigma_{N-2} & \sigma_{N-1} & \sigma_N\end{bmatrix}^T$ and the b vector is

$$\begin{bmatrix}
-h^2\left(3 + v\right)\rho\omega^2 - \sigma_0\left(1 - \frac{3h}{2(r_i+h)}\right) \\
-h^2\left(3 + v\right)\rho\omega^2 \\
-h^2\left(3 + v\right)\rho\omega^2 \\
\vdots \\
\vdots \\
-h^2\left(3 + v\right)\rho\omega^2 \\
-h^2\left(3 + v\right)\rho\omega^2 \\
-h^2\left(3 + v\right)\rho\omega^2 - \sigma_{N+1}\left(1 + \frac{3h}{2(r_i+Nh)}\right)
\end{bmatrix}$$

Finding numerical solution for hoop stress

To compare the analytical solution for σ_θ with the numerical one, we need to evaluate $\sigma_\theta = r\frac{d\sigma}{dr} + \sigma_r + \rho\omega^2 r^2$ numerically from the numerical solution we found about using finite difference method. Using finite difference, this expression becomes

$$\sigma_{\theta_i} = \left(r_{int} + ih\right)\frac{\sigma_{i+1} - \sigma_{i-1}}{2h} + \sigma_i + \rho\omega^2\left(r_{int} + ih\right)^2$$

Where $r_{int} = 0.01$ meter. Since we do not know σ_{-1} and can not obtain derivative at the edge to approximate using phantom grid point, we will start from $i = 1 \cdots N$, so that $\sigma_{i-1} = \sigma_0$ which is known.

Hence for $i = 1$ we have

$$\sigma_{\theta_1} = (r_{int} + h)\frac{\sigma_2 - \sigma_0}{2h} + \sigma_1 + \rho\omega^2 (r_{int} + h)^2$$

And for $i = N$

$$\sigma_{\theta_N} = (r_{int} + Nh)\frac{\sigma_{N+1} - \sigma_{N-1}}{2h} + \sigma_N + \rho\omega^2 (r_{int} + Nh)^2 \tag{6}$$

In the above, σ_{N+1} and σ_0 are the boundary conditions we are given. All other σ_i values are obtained from the numerical solution we did in the last section (the finite difference solution). This was implemented in Matlab.

0.3.2 Results

The program was run for 10, 15, 20, 25, 30, 100 grid points. Each time, 4 plots were generated:

1. Plot comparing the analytical and FDM result for σ_r Title contains location and value of maximum σ_r reported by FDM based method.
2. Plot comparing the analytical and bvp4c result for σ_r Title contains location and value of maximum σ_r reported by bvp4c based method
3. Plot comparing bvp4c and FDM result for σ_r
4. Plot comparing analytical result and FDM for σ_θ Title contains location and value of maximum σ_θ reported by FDM based method.

The result shows that bvp4c was more accurate than FDM for small number of grid points (large h) since the bvp4c curve was closer to the analytical solution than FDM curve. To get more accurate numerical hoop stress, the number of grid points needed to be over 50. At $N = 100$ grid points, the result of σ_θ matched the analytical solution extremely well as can be seen from the plots below. The following table gives the maximum radial stress found and the location as reported by bvp4c and FDM based methods. Units for stress is N/m^2 and units for r is meters. This shows the maximum radial stress occurs at left edge $r = 0.01$ meter as expected. This is where the inner hole edge starts. The maximum radial stress is located at $r = 0.032$ meter. About 2 cm after the inner hole edge.

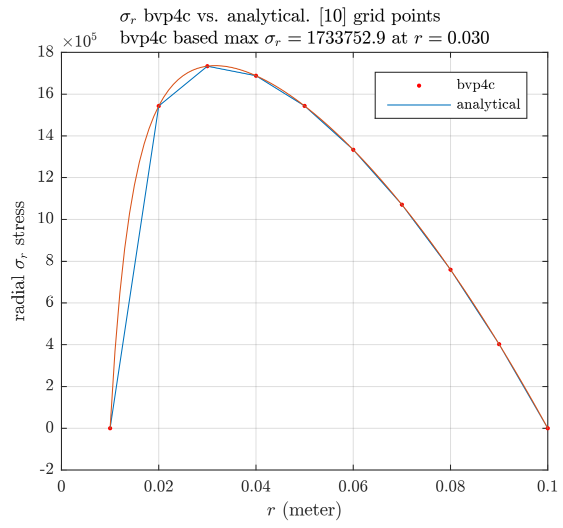
N (grid points)	Max σ_r FDM based, and r location	Max σ_r bvp4c based, and location
10	σ_r =1,783,600 at $r = 0.03$	σ_r =1,733,752 at $r = 0.03$
15	σ_r =1,731,188 at $r = 0.29$	σ_r =1,752,483 at $r = 0.029$
20	σ_r =1,732,767 at $r = 0.034$	σ_r =1,741,523 at $r = 0.034$
25	σ_r =1,735,518 at $r = 0.033$	σ_r =1,741,592 at $r = 0.033$
30	σ_r =1,735,984 at $r = 0.032$	σ_r =1,740,593 at $r = 0.032$
100	σ_r =1,736,401 at $r = 0.032$	σ_r =1,736,759 at $r = 0.032$

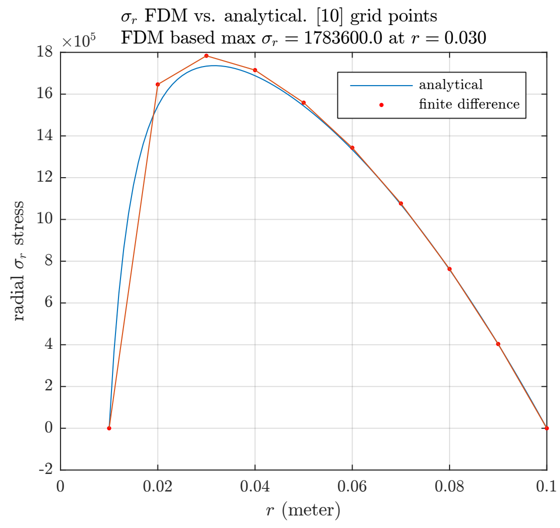
The following table gives the maximum hoop stress σ_θ found and the location as reported FDM based method. Units for stress is N/m^2 and units for r is meters. Each plot below also display this information in the title.

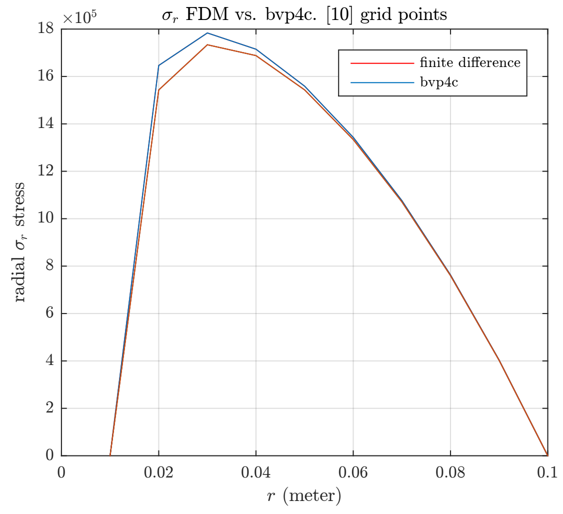
N (grid points)	Max σ_θ value and location
10	σ_θ =3,626,000 at $r = 0.01$
15	σ_θ =3,639,120 at $r = 0.01$
20	σ_θ =3,665,659 at $r = 0.01$
25	σ_θ =3,697,797 at $r = 0.01$
30	σ_θ =3,730,811 at $r = 0.01$
100	σ_θ =4,004,798 at $r = 0.01$

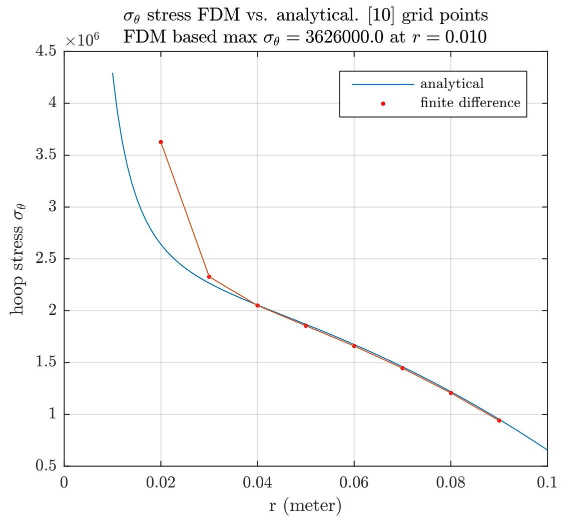
There are total of 24 plots below. Four plots for each N. This helps show that more grid points resulted in more accurate numerical solution.

10 grid points

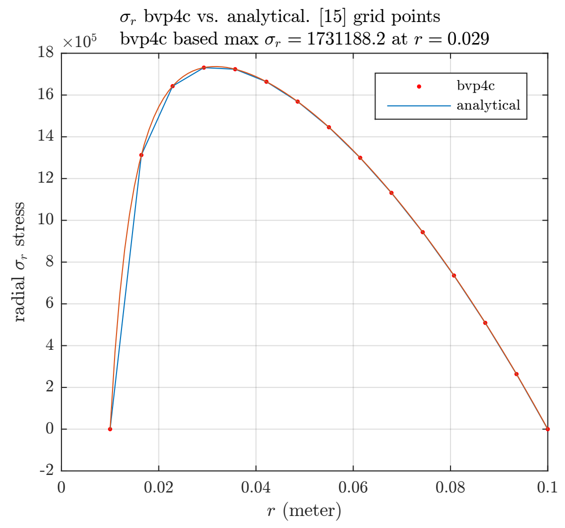


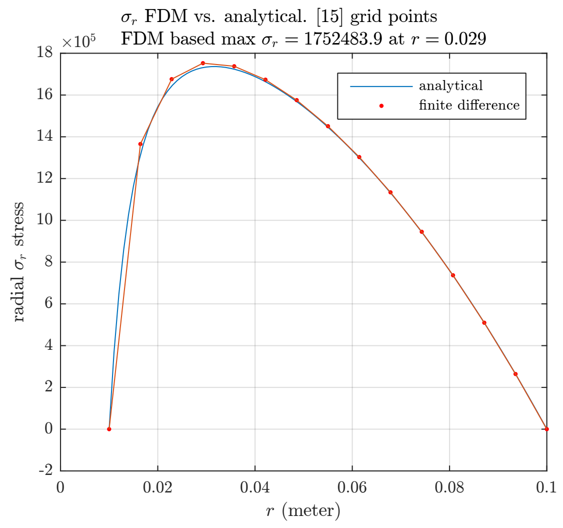


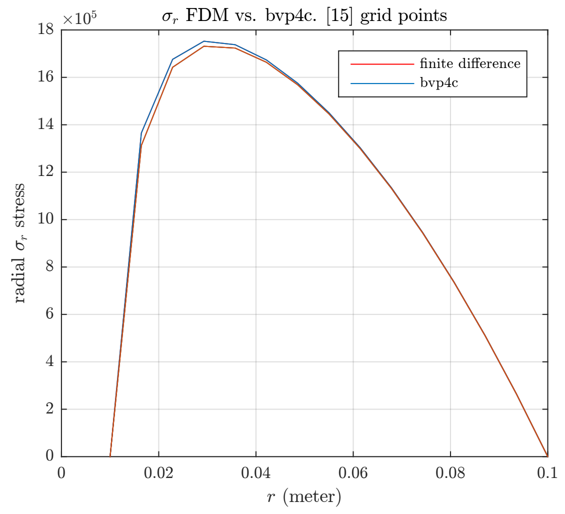


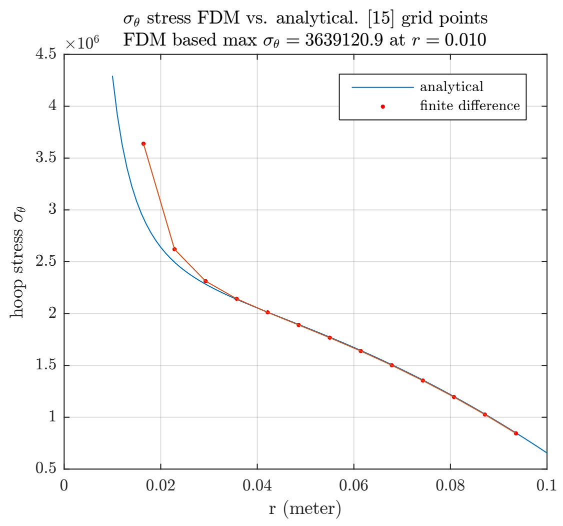


15 grid points

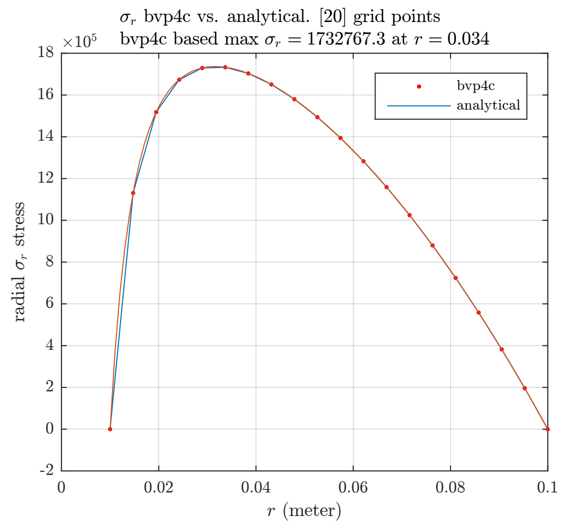


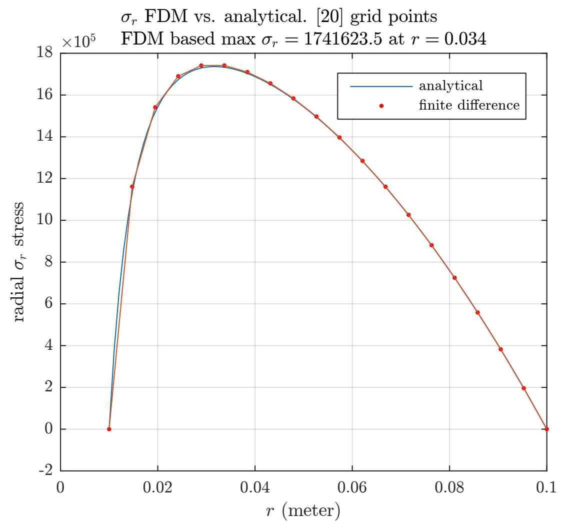


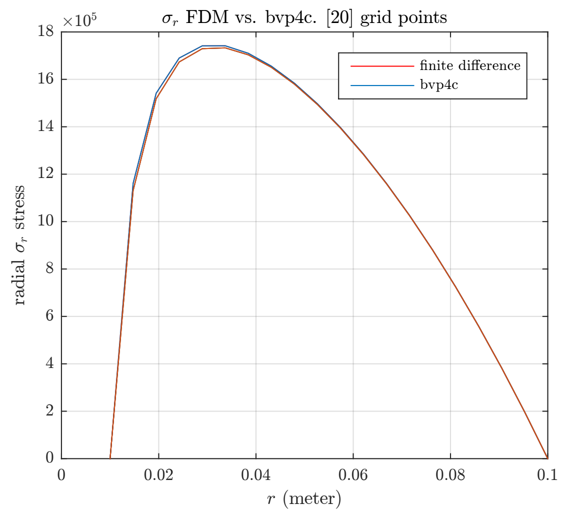


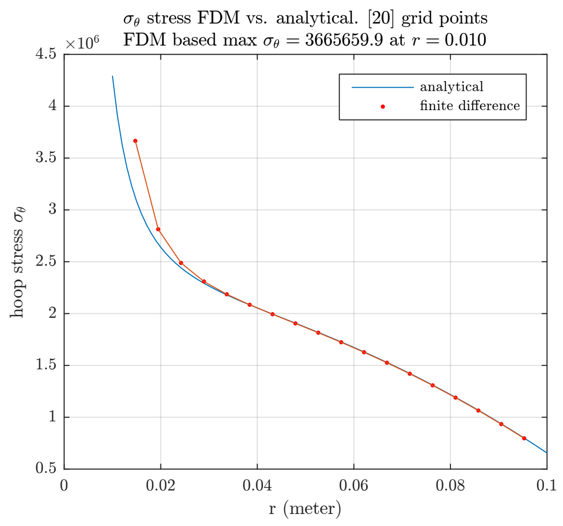


20 grid points

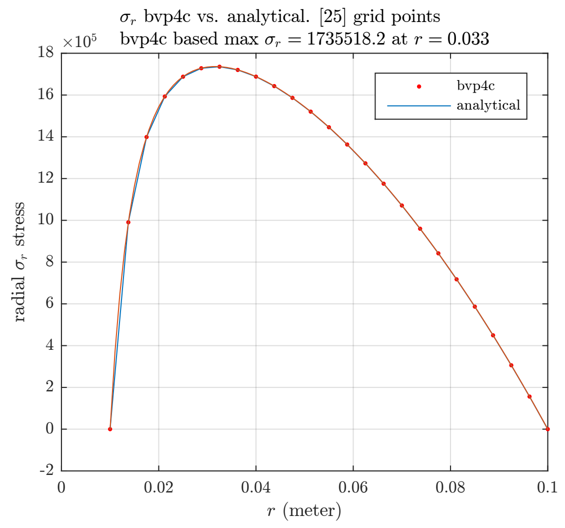


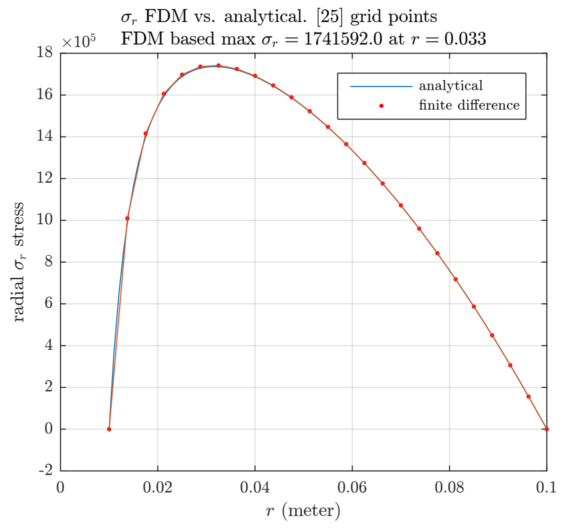


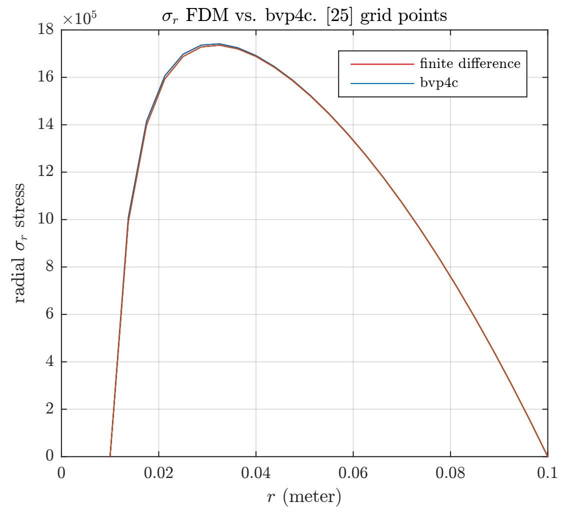


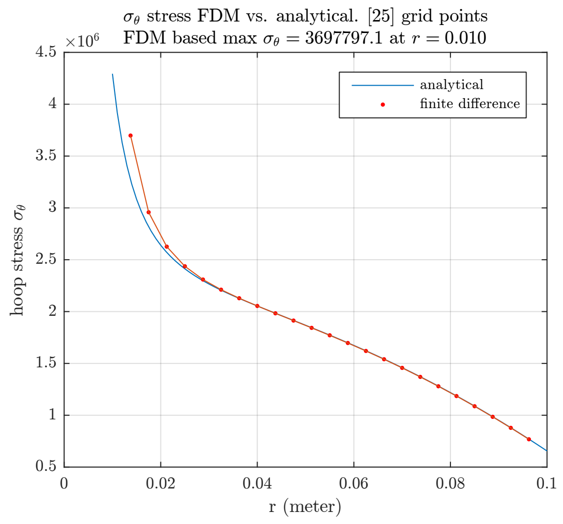


25 grid points

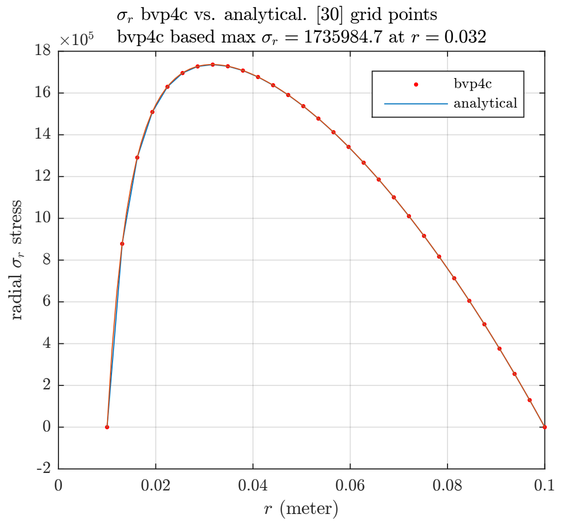


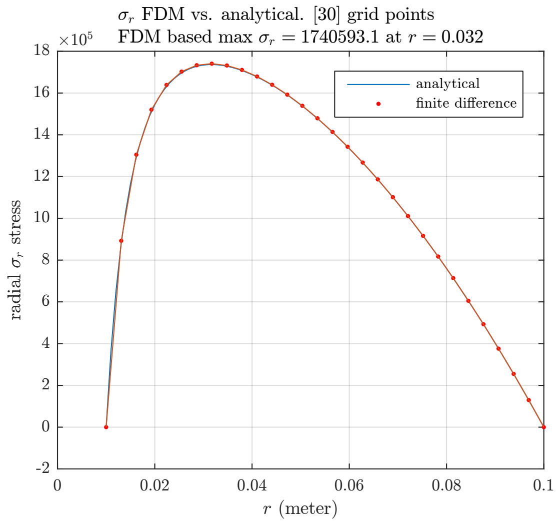


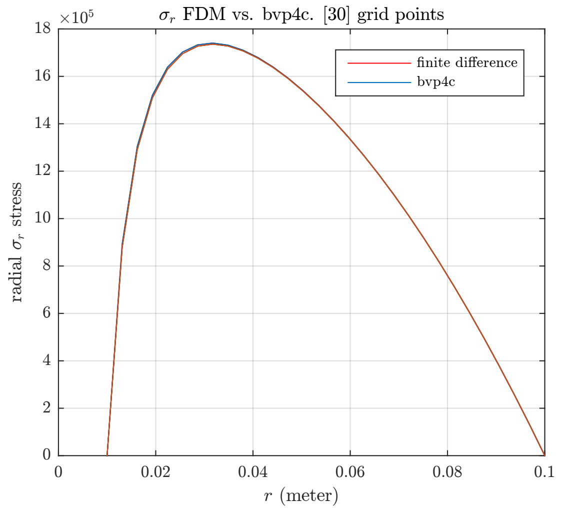


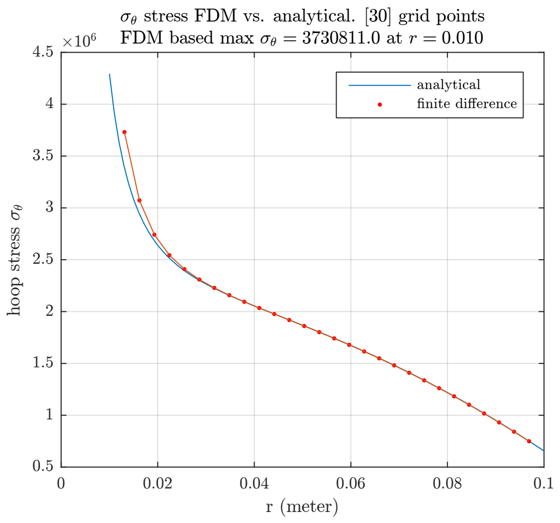


30 grid points

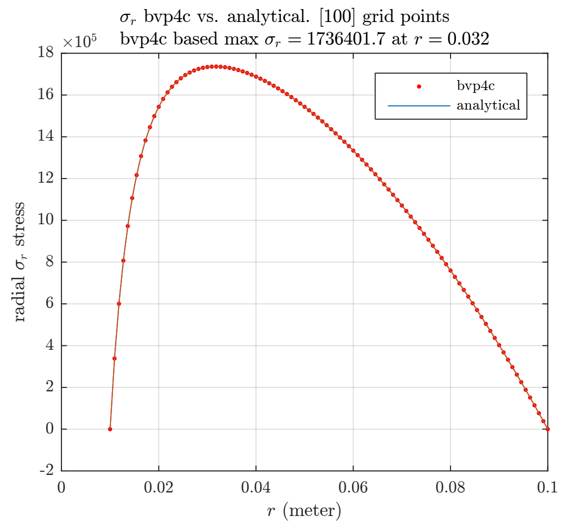


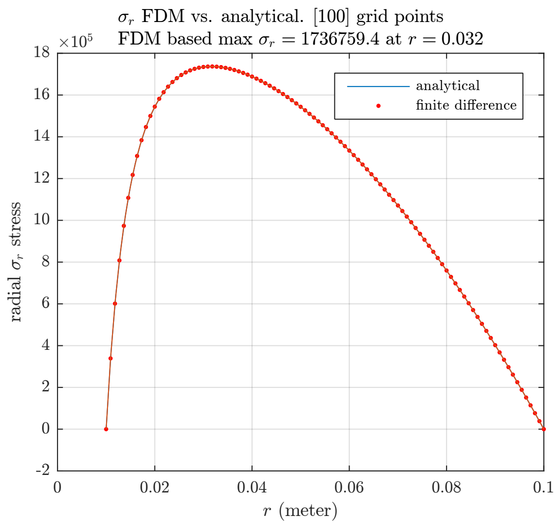


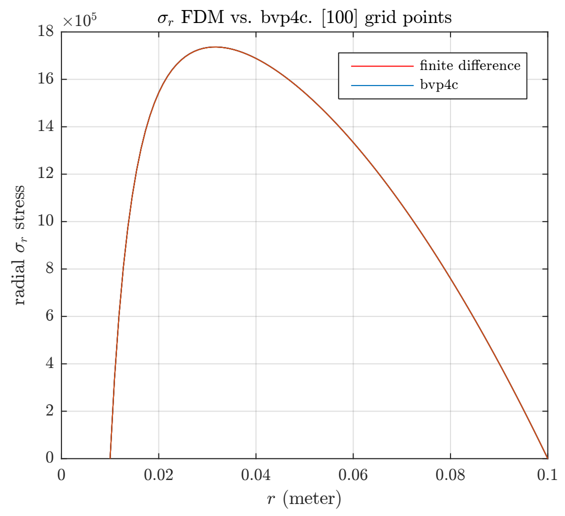


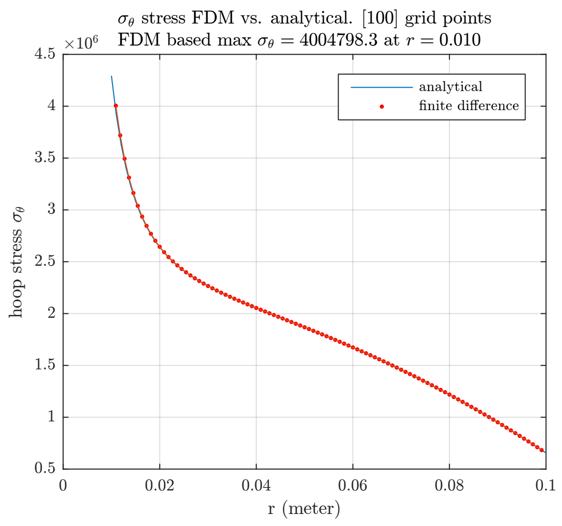


100 grid points









0.3.3 source code

```
function nma_EMA471_HW2_prob_3
%Solves problem 3, HW2. EMA 471, spring 2016
%Nasser M. Abbasi

clc; close all;

%generate the analytical solution first, to use to compare the numerical
%results against.

ri          = 0.01; %meter
ro          = 0.1;  %meter
x_analytic  = ri:0.001:ro;
density     = 1000; %kg/m^3
omega       = 700;  %radians per second
```

```
nu          = 0.5;  %Poisson's ratio

y_analytic = get_analytical_solution(x_analytic,ri,ro,density,omega,nu);

%generate results for different grid sizes
N             = 100; %select the number of grid points
[x_FDM,y_FDM] = make_one_test(N,x_analytic,y_analytic,ri,ro,density,omega,nu);

%now find the analytical solution for hoop stress, and compare with
%the numerical one
y_analytic_hoop = get_analytical_solution_hoop(x_analytic, ...
                                         y_analytic,ri,ro,density,omega,nu);

%compare with numerical result for hoop stress
make_one_test_hoop(N,x_FDM,y_FDM,x_analytic,y_analytic_hoop,ri,density,omega);

end

%-------------------------------------------------
%This function generates all the plots for bvp4c and FDM
%using specific number of grid points
%
function [x_bvp4c,y_FDM] = make_one_test(N,x_anaytic,...
                                        y_analytical,ri,ro,density,omega,nu)

%reset for plotting only
reset(0);
set(groot,'defaulttextinterpreter','Latex');
set(groot, 'defaultAxesTickLabelInterpreter','Latex');
set(groot, 'defaultLegendInterpreter','Latex');

x_bvp4c  = linspace(ri,ro,N);
solinit1 = bvpinit(x_bvp4c,[1 1]); %use specificed N grid points
%options = bvpset('RelTol',1e-6,'AbsTol',1e-6);  %Was not needed for this

sol = bvp4c(@rhs,@bc,solinit1);

%extract the x and y solution for plotting. These variables are used
%later in the plots
y_bvp4c = deval(sol,x_bvp4c); %evaluate at our grid point, to compare with FDM
y_bvp4c = y_bvp4c(1,:)';

%plot bvp4c vs. analytical
figure();
plot(x_bvp4c,y_bvp4c,'r.',x_bvp4c,y_bvp4c);
[max_radial_stress,I]=max(y_bvp4c);
hold on;
plot(x_anaytic,y_analytical);
xlabel('$r$ (meter)'); ylabel('radial $\sigma_r$ stress');
title({sprintf('$\\sigma_r$ bvp4c vs. analytical. [%d] grid points',N),...
      sprintf('bvp4c based max $\\sigma_r = %3.1f$ at $r= %3.3f$',max_radial_stress,x_bvp4c(I))});
legend('bvp4c','analytical','location','northeast');
grid;
set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);
```

```

%Obtain FDM solution. Use the same grid spacing as bvp4c (ie. same x spacing)
y0    = 0;
yL    = 0;
y_FDM = finite_difference_solution(x_bvp4c,y0,yL,ri,density,omega,nu);
[max_radial_stress,I] = max(y_FDM);

%plot finite difference vs. analytical
figure();
plot(x_anaytic,y_analytical);

hold on;
plot(x_bvp4c,y_FDM,'r.',x_bvp4c,y_FDM);
legend('analytical','finite difference','location','northeast');
xlabel('$r$ (meter)'); ylabel('radial $\sigma_r$ stress');
title({sprintf('$\\sigma_r$ FDM vs. analytical. [%d] grid points',N),...
      sprintf('FDM based max $\\sigma_r = %3.1f$ at $r= %3.3f$',max_radial_stress,x_bvp4c(I))});
grid;
set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);

%plot finite difference vs. bvp4c
figure();
plot(x_bvp4c,y_FDM,'r',x_bvp4c,y_FDM);

hold on;
plot(x_bvp4c,y_bvp4c,'k',x_bvp4c,y_bvp4c);
legend('finite difference','bvp4c','location','northeast');
xlabel('$r$ (meter)'); ylabel('radial $\sigma_r$ stress');
title(sprintf('$\\sigma_r$ FDM vs. bvp4c. [%d] grid points',N));
grid;
set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);


    %----------------------------------------------------------------------%
    %This internal function used by bvp4c, the RHS. Same as ODE45 RHS function

    function f = rhs(t,x)
        x1 = x(2);
        x2 = -(3+nu)*density*omega^2 - 3/t * x(2);
        f = [ x1
              x2];
    end
    %----------------------------------------------------------------------%
    %This internal is the boundary conditions function for bvp4c
    function res = bc(ya,yb)
        res = [ ya(1)
                yb(1)];
    end
end
%----------------------------------------------
%This function generate numerical solution for hoop stress, using the
%solution found earlier for the stress from finite difference and
%compare the result to the analytical solution of hoop stress.
function make_one_test_hoop(N,x_FDM,sigma_FDM,x_analytic,y_analytic_hoop,...
                                                       ri,density,omega)

%Obtain FDM based hoop stress, uses stress allready solved using FDM
h=x_FDM(2)-x_FDM(1);
i=(2:length(x_FDM)-1)';

%tried one point difference for finding hoop stress at initial and
%end points, but since O(h), it gives bad result. So keep the calculation
%for the internal points only. Block commented out
%
%hoop_stress_FDM=zeros(length(x_FDM),1);
%hoop_stress_FDM(1) = ri * (sigma_FDM(2)-sigma_FDM(1))/h + ...
%                      sigma_FDM(1) + density*omega^2*ri^2;
%hoop_stress_FDM(end) = ro * (sigma_FDM(end)-sigma_FDM(end-1))/h + ...
%                      sigma_FDM(end)+density*omega^2*(ro)^2;
%hoop_stress_FDM(2:end-1) = (ri+(i-1)*h) .* (sigma_FDM(i+1)-sigma_FDM(i-1))/(2*h) + ...
%                      sigma_FDM(i)+density*omega^2*(ri+(i-1)*h).^2;

%This below implements eq(6) in the HW report.
hoop_stress_FDM = (ri+(i-1)*h) .* (sigma_FDM(i+1)-sigma_FDM(i-1))/(2*h) + ...
                      sigma_FDM(i)+density*omega^2*(ri+(i-1)*h).^2;
[max_hoop_stress,I] = max(hoop_stress_FDM);
%plot finite difference vs. analytical hoop stress
figure();
plot(x_analytic,y_analytic_hoop);
hold on;
plot(x_FDM(2:end-1),hoop_stress_FDM,'r.',x_FDM(2:end-1),hoop_stress_FDM);
legend('analytical','finite difference','location','northeast');
xlabel('r (meter)');  ylabel('hoop stress $\sigma_\theta$');
title({sprintf('$\\sigma_\\theta$ stress FDM vs. analytical. [%d] grid points',N),...
  sprintf('FDM based max $\\sigma_\\theta = %3.1f$ at $r= %3.3f$',max_hoop_stress,x_FDM(I))});

grid;
set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);

end
%---------------------------------------------
%This function returns the analytical solution for hoop stress.
%Solved in the HW report
function y = get_analytical_solution_hoop(r,sigma,ri,ro,density,omega,nu)

k = (3+nu)*density*omega^2/8;
y = k*( 2./(r.^2) * ri^2*ro^2 -2 * r.^2) + sigma + density*omega^2*r .^2;

end
%---------------------------------------------
%This function returns the analytical solution. Solved in the HW report
function y=get_analytical_solution(x,ri,ro,density,omega,nu)

k = (3+nu)*density*omega^2/8;
y = k*( (ri^2+ro^2)- ri^2*ro^2 ./ (x.^2) - x.^2);
end
%---------------------------------------------
%This function generates the Finite difference solution
```

```matlab
%The HW report contains the derivation used for Ax=b setup
%
function y = finite_difference_solution(x,y0,yL,ri,density,omega,nu)
%Allocate A matrix and b vector
h = x(2)-x(1);   %h spacing is the same between each grid point
N = length(x)-2; %number of internal points.
A = zeros(N,N);
b = zeros(N,1);  %allocate RHS, for Ax=b use

%Start filling the A matrix

%fill in the first row by hand
A(1,1) = -2;
A(1,2) = (3*h)/(2*(ri+h))+1;

k=0; %column index, used for the loop below, to fill the rest of A
for i = 2:N-1
     k = k+1;
     A(i,k:k+2) = [1-(3*h)/(2*(ri+i*h)),-2,(3*h)/(2*(ri+i*h))+1];
end

%fill in the last  row by hand
k = k+1;
A(N,k:k+1)   = [1-(3*h)/(2*(ri+N*h)),-2];

%fill in the b matrix. The first  row1 by hand
b(1) = -h^2*(3+nu)*density*omega^2-y0*(1-(3*h)/(2*(ri+h)));

%fill the rest of b using loop
for i = 2:N-1
    b(i) = -h^2*(3+nu)*density*omega^2;
end

%fill the last row of b
b(N)   = -h^2*(3+nu)*density*omega^2-yL*((3*h)/(2*(ri+N*h))+1);

%now we solve the system.
y=A\b;
y=[y0;y;yL]; %pad in the left and right boundary values. Known values.
end
```