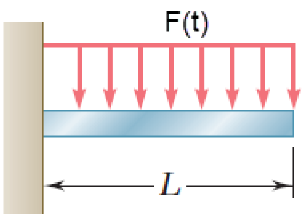
DREXEL UNIVERSITY
Department of Mechanical Engineering & Mechanics
Applied Engineering Analytical & Numerical Methods II

MEM 592 - Winter 2019

HOMEWORK #5: Due Friday, March 1st, 2019 at 2:00 PM

1. [40 points]

The boundary-value problem governing the deflection at the end of the beam subject to uniform loading, F(t) which is changing over the time, is



$$w''(t) = 4w(t) - 4t, \qquad for\ \ 0 < t < 1\ \ with\ \ w(0) = 0\ \ \ and\ \ \ w(1) = 2.$$

a)Find the exact solution for the deflection at the end of the beam, w(t), during the given time.

b) Write formula of the second-order FD approximation for w(t). Then, plot your results and compare with the exact solution in part (a). (Use h=0.01 and 0.1)

c) What is the maximum error on the interval when h=0.01?

2. [30 points]

Consider the following boundary value problem:

$$y''(x) = 9y + e^{3x} + \sin(3x), \qquad 0 \le x \le 1\,, \qquad y(1) = 2, \qquad y(0) = 0,$$

1. Find the exact solution for the problem.
2. Use linear shooting method to solve the problem. Follow the steps below:
    a) Consider $y'(0) = 2$, as your first initial guess. Then, find the y(x) in the whole domain for this initial guess. (using Euler's method, h=0.1)
    b) Consider $y'(0) = -2$, as your second initial guess. Then, find the y(x) in the whole domain for this initial guess. (using Euler's method, h=0.1)
    c) Use shooting method in slide09, chater05 to find y(x) in the whole domain.
    d) Plot your results from parts (1), (2.a), (2.b) and (2.c) in one figure.
    e) Plot your results from parts (1), (2.a), (2.b) and (2.c) in one figure for h=0.01.

3. [30 points]

Consider the following boundary value problem:

$$y'' = p(x)y' + q(x)y + r(x), \qquad a \le x \le b\,, \qquad y(a) = \alpha, \qquad y(b) = \beta, \qquad Eq.(1)$$

To approximate the unique solution to the linear boundary-value problem, first consider the two initial-value problems:

$$y_1'' = p(x)y_1' + q(x)y_1 + r(x), \qquad a \le x \le b\,, \qquad where\;\; y_1(a) = \alpha, y'_1(a) = 0 \;\; Eq.(2)$$

and

$$y_2'' = p(x)y_2' + q(x)y_2, \qquad a \le x \le b\,, \qquad where\;\; y_2(a) = 0, y'_2(a) = 1 \;\; Eq.(3)$$

Then, the unique solution to the linear boundary-value problem can be found as:

$$y(x) = y_1(x) + \left(\frac{\beta - y_1(b)}{y_2(b)}\right) y_2(x), \qquad Eq.(4).$$

a) Verify that Eq.(4) is the solution for Eq. (2).

b) Consider the following boundary value problem:

$y'' = \left(\frac{-2}{x}\right) y' + \left(\frac{2}{x^2}\right) y + \frac{sin(Ln(x))}{x^2} \quad Eq.(5),\; 1 \le x \le 2\,,\; y(1) = 1,\; y(2) = 2,$ has the exact solution:

$$y^{exact}(x) = c_1 x + c_2 \left(\frac{1}{x^2}\right) - \frac{3}{10}\sin\big(Ln(x)\big) - \frac{1}{10}\cos\big(Ln(x)\big),$$

$$c_2 = \frac{1}{70}\Big(8 - 12\sin\big(Ln(2)\big) - 4\cos\big(Ln(2)\big)\Big) \;\; and \;\; c_1 = \frac{11}{10} - c_2$$

- Rewrite the Eq.(2) for this problem(Eq.(5)) to reach a system of initial value problem and then use fourth-order Runge-Kutta to solve for $y_1(x)$. (h=0.1)
- Rewrite the Eq.(3) for this problem (Eq.(5)) to reach a system of initial value problem and then use fourth-order Runge-Kutta to solve for $y_2(x)$. (h=0.1)
- Find y(x) according to Eq.(4). Then plot your results for y(x) and $y^{exact}(x)$ in one figure.
- Consider h=0.01, then find y(x) according to Eq.(4). Then plot your results for y(x) and $y^{exact}(x)$ in one figure.

HW 5 Hyukjun Kwon

1) $W''(t) = 4W(t) - 4t$, for $0 < t < 1$

$W(0) = 0$

$W(1) = 2$

a) Exact

$W'' - 4W = -4t$

Y_h) $W'' - 4W = 0$

$\lambda^2 - 4 = 0 \rightarrow \lambda = 2, -2$

$Y_h = C_1 e^{2t} + C_2 e^{-2t}$

Y_p) $r(t) = -4t$

$Y_p = k_1 t + k_0$

$Y_p' = k_1$

$Y_p'' = 0$

substitute

$0 - 4k_1 t - 4k_0 = -4t$

$0 - 4k_0 = 0 \rightarrow k_0 = 0$

$-4k_1 = -4 \longrightarrow k_1 = 1$

So. $Y_p = t$

Y) $Y_h + Y_p$

$W = C_1 e^{2t} + C_2 e^{-2t} + t$

$W(0) = C_1 + C_2 = 0$

$$W(1) = C_1 e^2 + C_2 e^{-2} + 1 = 2$$

$$\begin{bmatrix} 1 & 1 \\ e^2 & e^{-2} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} \frac{e^2}{e^4-1} \\ \frac{-e^2}{e^4-1} \end{bmatrix}$$

Therefore,

$$\left[W(t) = \left(\frac{e^2}{e^4-1}\right) e^{2t} - \left(\frac{e^2}{e^4-1}\right) e^{-2t} + t \right]$$

b) 2nd order FD approximation

$$\frac{W_{J+1} - 2W_J + W_{J-1}}{h^2} = 4W_J - 4t_J$$

$$W_{J+1} - 2W_J + W_{J-1} = 4h^2 W_J - 4h^2 t_J$$

$$\left[W_{J+1} + (-2-4h^2) W_J + W_{J-1} = -4h^2 t_J \right]$$

→ Forms tridiagonal system

Special treatment required for $J=1$ & $J=N-1$

$J=1$)

$$\underline{W_2 + (-2-4h^2) W_1} = -4h^2 t_J - \overset{\rightarrow 0}{W_0} = \underline{-4h^2 t_1}$$

$J=N-1$)

$$W_N + (-2-4h^2) W_{N-1} + W_{N-2} = -4h^2 t_{N-1}$$

$$\underline{(-2-4h^2) W_{N-1} + W_{N-2}} = \underline{-4h^2 t_{N-1} - 2}$$

In Matrix,

$$\begin{bmatrix} (-2-4h^2) & (1) & & & & \\ (1) & (-2-4h^2) & (1) & & & \\ 0 & (1) & (-2-4h^2) & (1) & & \\ \vdots & \vdots & \ddots & \ddots & \ddots & \\ & & (1) & (-2-4h^2) & (1) \\ & & & (1) & (-2-4h^2) \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \vdots \\ W_{N-2} \\ W_{N-1} \end{bmatrix} = \begin{bmatrix} -4h^2 t_1 \\ -4h^2 t_2 \\ -4h^2 t_3 \\ \vdots \\ \\ -4h^2 t_{N-1} - 2 \end{bmatrix}$$

Depending on h value, N changes

$h = 0.01$ & $h = 0.1$ are plotted with exact solution in Matlab

Also once W vector is found,

W_0 and W_L values will be attached $\rightarrow$ $W = \begin{array}{l} 0 \leftarrow [W_0] \\ \begin{bmatrix} W_1 \\ \vdots \\ W_{N-1} \end{bmatrix} \\ 2 \leftarrow [W_N] \end{array}$

Concatenate

3) Max Error

$\rightarrow$ Matlab

max(abs(Error))

2) $y''(x) = 9y + e^{3x} + \sin(3x), \quad 0 \le x \le 1$

$y(1) = 2$

$y(0) = 0$

2-1) Exact Solution

$y''(x) - 9y = e^{3x} + \sin(3x)$

$y_h)$ $y'' - 9y = 0$

$\lambda^2 - 9 = 0 \rightarrow \lambda = 3, -3$

$y_h = C_1 e^{-3t} + C_2 e^{3t}$

$y_p)$ $r(t) = e^{3x} + \sin(3x)$

$y_p = Axe^{3x} + B\cos(3x) + C\sin(3x)$

$y_p' = Ae^{3x} + 3Axe^{3x} - 3B\sin(3x) + 3C\cos(3x)$

$y_p'' = 3Ae^{3x} + 3Ae^{3x} + 9Axe^{3x} - 9B\cos(3x) - 9C\sin(3x)$

$6Ae^{3x} + \cancel{9Axe^{3x}} - 9B\cos(3x) - 9C\sin(3x) - \cancel{9Axe^{3x}}$

$-9B\cos(3x) - 9C\sin(3x) = e^{3x} + \sin(3x)$

$6A = 1 \rightarrow A = 1/6$

$-18B = 0 \rightarrow B = 0$

$-18C = 1 \rightarrow C = -1/18$

$y_p = \frac{1}{6}xe^{3x} - \frac{1}{18}\sin(3x)$

So...

$y = C_1 e^{-3t} + C_2 e^{3t} + \frac{1}{6}xe^{3x} - \frac{1}{18}\sin(3x)$

$$Y(0) = C_1 + C_2 = 0$$

$$Y(1) = C_1 e^{-3} + C_2 e^{3} + \frac{1}{6}e^{3} - \frac{1}{18}\sin(3) = 2$$

$$\begin{bmatrix} 1 & 1 \\ e^{-3} & e^{3} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -1.3397 \end{bmatrix} \rightarrow \begin{bmatrix} 0.0669 \\ -0.0669 \end{bmatrix}$$

Therefore,,

$$\left[Y = (0.0669)e^{-3t} - (0.0669)e^{3t} + \frac{1}{6}xe^{3x} - \frac{1}{18}\sin(3x) \right]_{\text{,,}}$$

2-2) use

$$y = y_1 \qquad y' = y_2 \quad \rightarrow \quad y_1' = y_2 \qquad y_2' = 9y_1 + e^{3x} + \sin(3x)$$

conditions

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_0 = \begin{bmatrix} 0 \\ ? \end{bmatrix}, \quad \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_1 = \begin{bmatrix} 2 \\ \text{any\#} \end{bmatrix}$$

$?$ ↳ will be given

2 ← requirement

a) Euler (FE)

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_{n+1} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_n + h \begin{bmatrix} y_2 \\ 9y_1 + e^{3x} + \sin(3x) \end{bmatrix}_n$$

with $IC = \begin{bmatrix} 0 \\ 2 \end{bmatrix}_0$

b) Euler (FE)

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_{n+1} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_n + h \begin{bmatrix} y_2 \\ 9y_1 + e^{3x} + \sin(3x) \end{bmatrix}_n$$

with $IC = \begin{bmatrix} 0 \\ -2 \end{bmatrix}_n$

c) $y = c_1 y_1 + c_2 y_2$

↳ y found in (b)

↳ $c_2 = \dfrac{y_1(L) - y_L}{y_1(L) - y_2(L)} = 0.5970$

↳ y found in (a)

$c_1 = \dfrac{y_L - y_2(L)}{y_1(L) - y_2(L)} = 0.4030$

d) Matlab

e) Matlab

3)

EQ1
$$\begin{cases} y'' = p(x)y' + q(x)y + r(x), \quad a \le x \le b \\ y(a) = \alpha \\ y(b) = \beta \end{cases}$$

EQ2
$$\begin{cases} y_1'' = p(x)y_1' + q(x)y_1 + r(x), \quad a \le x \le b \\ y_1(a) = \alpha \\ y_1'(a) = 0 \end{cases}$$

EQ3
$$\begin{cases} y_2'' = p(x)y_2' + q(x)y_2, \quad a \le x \le b \\ y_2(a) = 0 \\ y_2'(a) = 1 \end{cases}$$

EQ4
$$\left(\; y(x) = y_1(x) + \left(\frac{\beta - y_1(b)}{y_2(b)}\right) y_2(x) \right.$$

Reza said it was 1...!

a) Verify EQ4 is solution for EQ1.

$$(y(x))' = y_1'(x) + \underbrace{\cancel{\left(\frac{\beta - y_1(b)}{y_2(b)}\right)' y_2(x)}}_{\to 0} + \left(\frac{\beta - y_1(b)}{y_2(b)}\right) y_2'(x)$$

$$(y(x))'' = y_1''(x) + \underbrace{\cancel{\left(\frac{\beta - y_1(b)}{y_2(b)}\right)' y_2'(x)}}_{\to 0} + \left(\frac{\beta - y_1(b)}{y_2(b)}\right) y_2''(x)$$

So... substituting y_1', y_2', y_1'', y_2'' to above equation...

$$y'(x) = y_1'(x) + \left(\frac{\beta - y_1(b)}{y_2(b)}\right) y_2'(x)$$

$$y''(x) = P(x)y_1' + q(x)y_1 + r(x) + \left(\frac{\beta - y_1(b)}{y_2(b)}\right)\left(P(x)y_2' + q(x)y_2\right)$$

Substituting $y'(x)$ and $y''(x)$ into EQ1

EQ1: $y'' - P(x)y' - q(x)y = r(x)$

$$\cancel{P(x)y_1'} + q(x)y_1 + r(x) + \left(\frac{\beta - y_1(b)}{y_2(b)}\right)\left(\cancel{P(x)y_2'} + q(x)y_2\right)$$

$$-P(x)y_1'(x) - P(x)\left(\frac{\beta - y_1(b)}{y_2(b)}\right)y_2'(x) - \cancel{q(x)y_1(x)}$$

$$-q(x)\left(\frac{\beta - y_1(b)}{y_2(b)}\right)y_2(x) = r(x)$$

Simplifying...

$$r(x) = r(x) \quad \checkmark \text{ verified}$$

b)

EQ 5
$$\begin{cases} y'' = \left(-\frac{2}{x}\right)y' + \left(\frac{2}{x^2}\right)y + \frac{\sin(\ln(x))}{x^2}, & 1 \le x \le 2 \\ y(1) = 1 \\ y(2) = 2 \end{cases}$$

Exact Solution
$$y_{ext}(x) = C_1 x + C_2\left(\frac{1}{x^2}\right) - \frac{3}{10}\sin(\ln(x)) - \frac{1}{10}\cos(\ln(x))$$

Where $C_1 = \frac{11}{10} - C_2$ and $C_2 = \frac{1}{70}\left(8 - 12\sin(\ln(2)) - 4\cos(\ln(2))\right)$

b-1) Rewrite for Y_1 $\quad 1 \le x \le 2$

$$Y_1'' = \left(-\frac{2}{x}\right)Y_1' + \left(\frac{2}{x^2}\right)Y_1 + \frac{\sin(\ln(x))}{x^2}$$

$Y_1(1) = 1$

$Y_1(2) = 2$

$Y_1'(1) = 0$

use

$Y_1 = z_1$

$Y_1' = z_2$

$$\longrightarrow \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix}_n = \begin{bmatrix} z_2 \\ \left(\frac{2}{x^2}\right)z_1 + \left(-\frac{2}{x^2}\right)z_2 + \frac{\sin(\ln(x))}{x^2} \end{bmatrix}_n$$

use 4th-order Runge-Kutta

$$(z)_{n+1} = z_n + \frac{1}{6}k_1 + \frac{1}{3}(k_2 + k_3) + \frac{1}{6}k_4$$

$$k_1 = h\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix}_n$$

↳ 0.1 for now

$$k_2 = hf\left(z_n + \frac{1}{2}k_1,\ x_n + \frac{h}{2}\right)$$

$f \to \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix}$

our input needs to be adjusted.

$$k_3 = hf\left(z_n + \frac{1}{2}k_2,\ x_n + \frac{1}{2}h\right)$$

$$k_4 = hf(z_n + k_3,\ x_n + h)$$

b-2) Rewrite for Y_2

$1 \leq x \leq 2$

$Y_2(1) = 0$

$Y_2'(1) = 1$

$Y_2(2) = 2$

$$Y_2'' = \left(-\frac{2}{x}\right)Y_2' + \left(\frac{2}{x^2}\right)Y_2$$

use

$$\begin{aligned} Y_2 &= u_1 \\ Y_2' &= u_2 \end{aligned} \longrightarrow \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{bmatrix}_n = \begin{bmatrix} u_2 \\ \left(\frac{2}{x^2}\right)u_1 + \left(-\frac{2}{x}\right)u_2 \end{bmatrix}_n$$

use 4th-order Runge Kutta

$$(u)_{n+1} = u_n + \frac{1}{6}k_1 + \frac{1}{3}(k_2 + k_3) + \frac{1}{6}k_4$$

$$k_1 = h\begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{bmatrix}, \quad k_2 = hf\left(u_n + \frac{1}{2}k_1, x_n + \frac{1}{2}h\right)$$

$$k_3 = hf\left(u_n + \frac{1}{2}k_2, x_n + \frac{1}{2}h\right)$$

$$k_4 = hf(u_n + k_3, x_n + h)$$

b-3) using Y_1 and Y_2, we can find Y based on EQ 4

$$Y = Y_1 + \underbrace{\left(\frac{\beta - Y_1(b)}{Y_2(b)}\right)}_{\text{constant!} = 0.9176} Y_2$$

b-4) $h = 0.01 \longrightarrow$ Matlab

MEM 592 - Homework 5

Author: Hyukjun Kwon

Date: February 27, 2019


```
clear,clc,close all
```

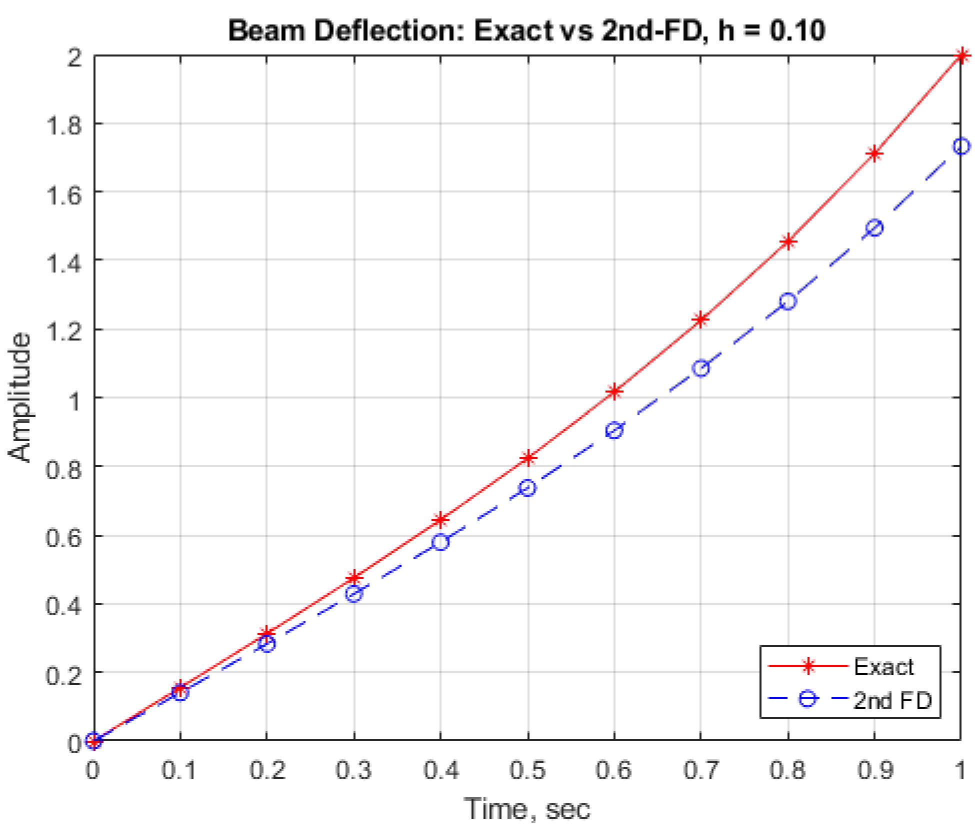
Problem #1 - Beam boundary value problem

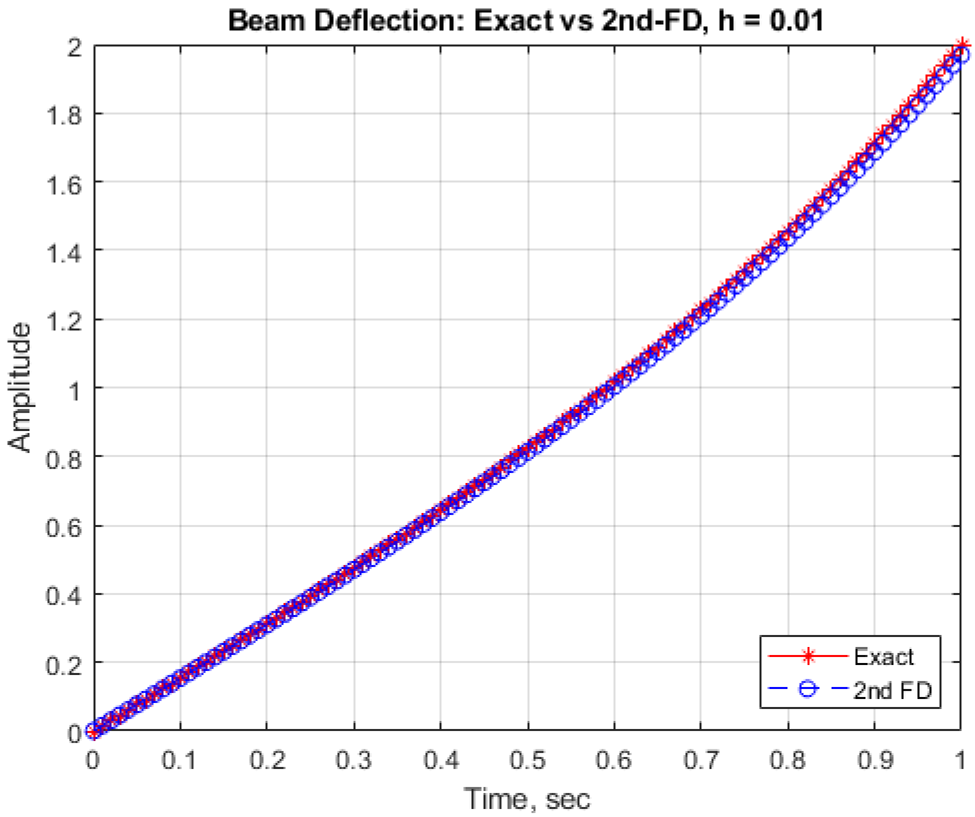
a) Exact Solution

```
clear
clc

% Exact Solution
y = @(t) (exp(2)/(exp(4)-1)).*exp(2.*t) + (-exp(2)/(exp(4)-1)).*exp(-2.*t) + t;

% 2nd FD Method
for i = 1:2
    figure(i)
    h_to_use = [0.1 0.01];
    h = h_to_use(i);

    t = 0:h:1;
    A_FD = zeros(length(t)-1,length(t)-1);
    A_FD(1,[1 2]) = [-2-(4*(h^2)) 1];
    A_FD(end,[end-1 end]) = [1 -2-(4*(h^2))];
    for j = 2:length(t)-2
        A_FD(j,[j-1 j j+1]) = [1 -2-(4*(h^2)) 1];
    end

    C_FD = zeros(length(t)-1,1);
    C_FD(1) = -4*(h^2)*t(2);
    C_FD(end) = (-4*(h^2)*t(end))-2;
    for k = 2:length(t)-2
        C_FD(k) = -4*(h^2)*t(k+1);
    end

    B_FD = zeros(length(t),1);
    B_FD(1) = 0;
    B_FD(2:end) = A_FD\C_FD;
```

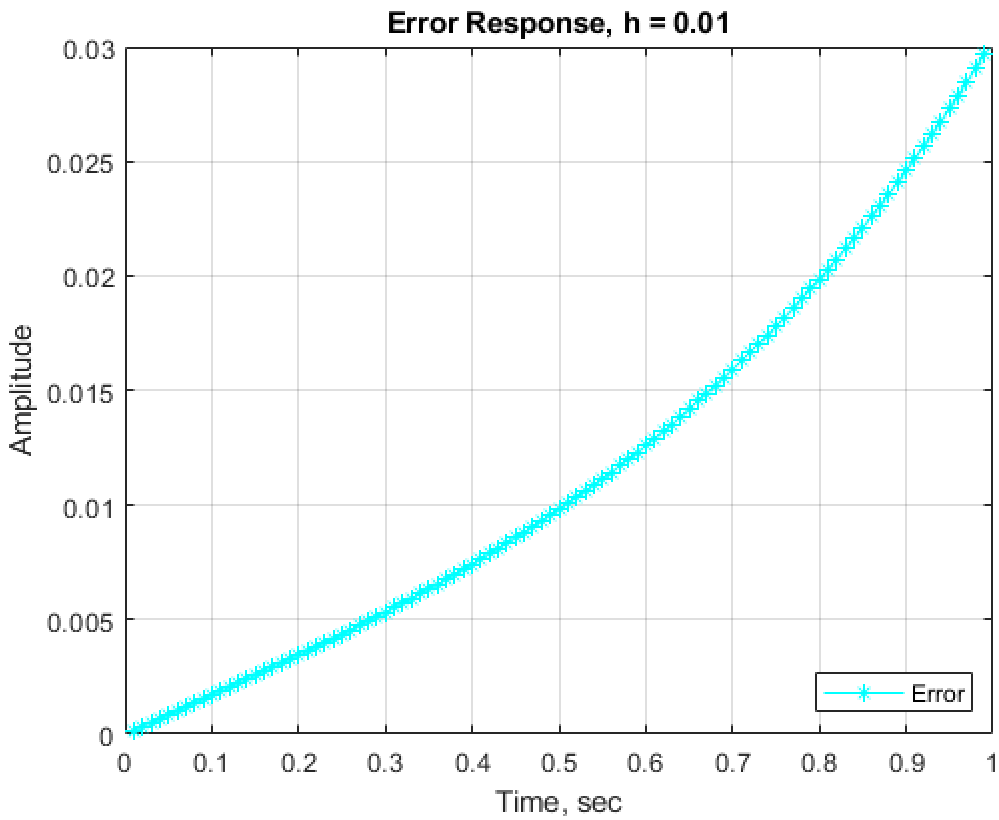
```
    plot(t,y(t),'r*-',t,B_FD,'bo--')
    grid on
    xlabel('Time, sec')
    ylabel('Amplitude')
    legend('Exact','2nd FD','location','southeast')
    title(sprintf('Beam Deflection: Exact vs 2nd-FD, h = %.2f',h))
end

% Max Error when h = 0.01
figure(3)
error = y(t)'-B_FD;
plot(t(2:end-1),error(2:end-1),'c*-')
grid on
xlabel('Time, sec')
ylabel('Amplitude')
legend('Error','location','southeast')
title(sprintf('Error Response, h = %.2f',h))

maxError = max(abs(error));
disp(maxError)
% Max Error is 0.0303.
```

```
0.0303
```





Beam Deflection: Exact vs 2nd-FD, h = 0.01
Amplitude
Time, sec
Exact
2nd FD
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.2
0.4
0.6
0.8
1.2
1.4
1.6
1.8
2

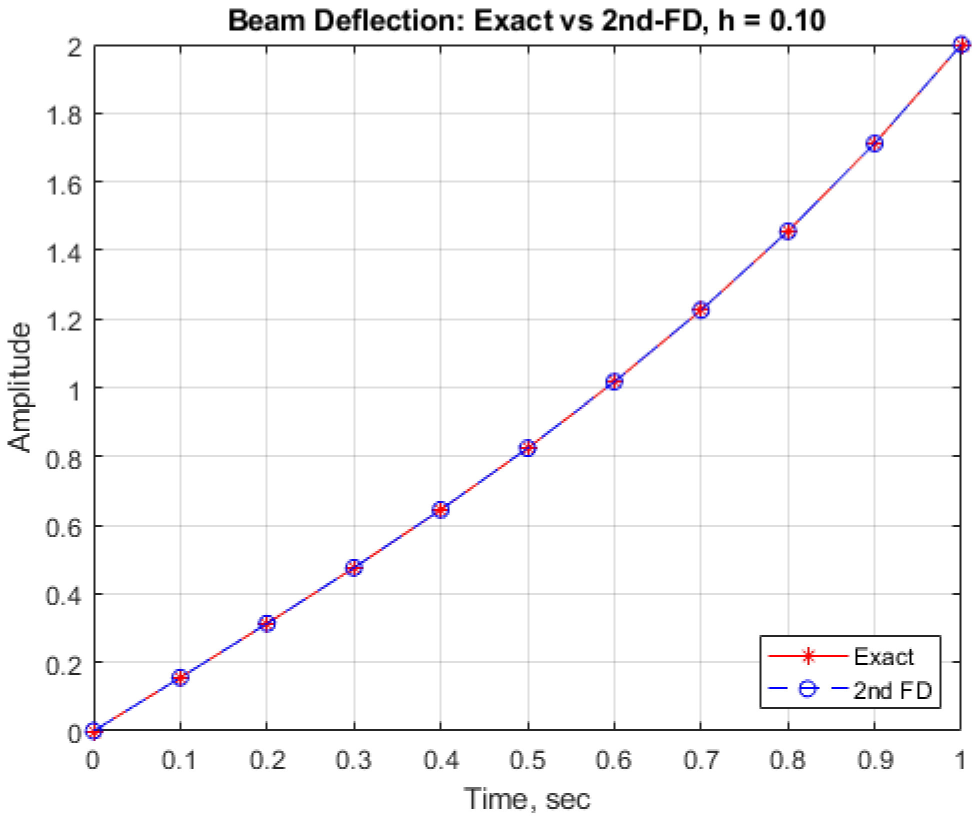


Problem #1 - Beam boundary value problem

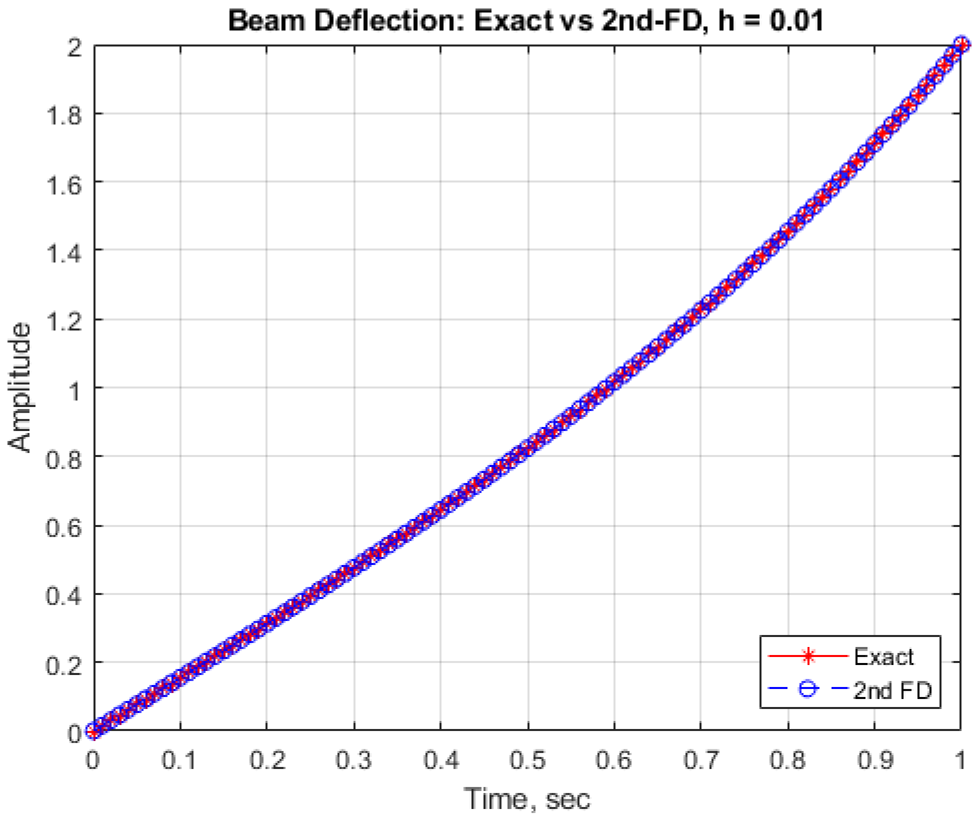
```
clear
clc

% Exact Solution
y = @(t) (exp(2)/(exp(4)-1)).*exp(2.*t) + (-exp(2)/(exp(4)-1)).*exp(-2.*t) + t;

% 2nd FD Method
for i = 4:5
    figure(i)
    h_to_use = [0.1 0.01];
    h = h_to_use(i-3);

    t = 0:h:1;
    A_FD = zeros(length(t)-2,length(t)-2);
    A_FD(1,[1 2]) = [-2-(4*(h^2)) 1];
    A_FD(end,[end-1 end]) = [1 -2-(4*(h^2))];
    for j = 2:length(t)-3
        A_FD(j,[j-1 j j+1]) = [1 -2-(4*(h^2)) 1];
    end

    C_FD = zeros(length(t)-2,1);
    C_FD(1) = -4*(h^2)*t(2);
    C_FD(end) = (-4*(h^2)*t(end-1))-2;
```

```
    for k = 2:length(t)-3
        C_FD(k) = -4*(h^2)*t(k+1);
    end

    B_FD = zeros(length(t),1);
    B_FD(1) = 0;
    B_FD(2:end-1) = A_FD\C_FD;
    B_FD(end) = 2;

    plot(t,y(t),'r*-',t,B_FD,'bo--')
    grid on
    xlabel('Time, sec')
    ylabel('Amplitude')
    legend('Exact','2nd FD','location','southeast')
    title(sprintf('Beam Deflection: Exact vs 2nd-FD, h = %.2f',h))
end

% Max Error when h = 0.01
figure(6)
error = y(t)'-B_FD;
plot(t(2:end-1),error(2:end-1),'c*-')
grid on
xlabel('Time, sec')
ylabel('Amplitude')
legend('Error','location','southeast')
title(sprintf('Error Response, h = %.2f',h))

maxError = max(abs(error));
disp(maxError)
% Max Error is 4.4252e-06.
```

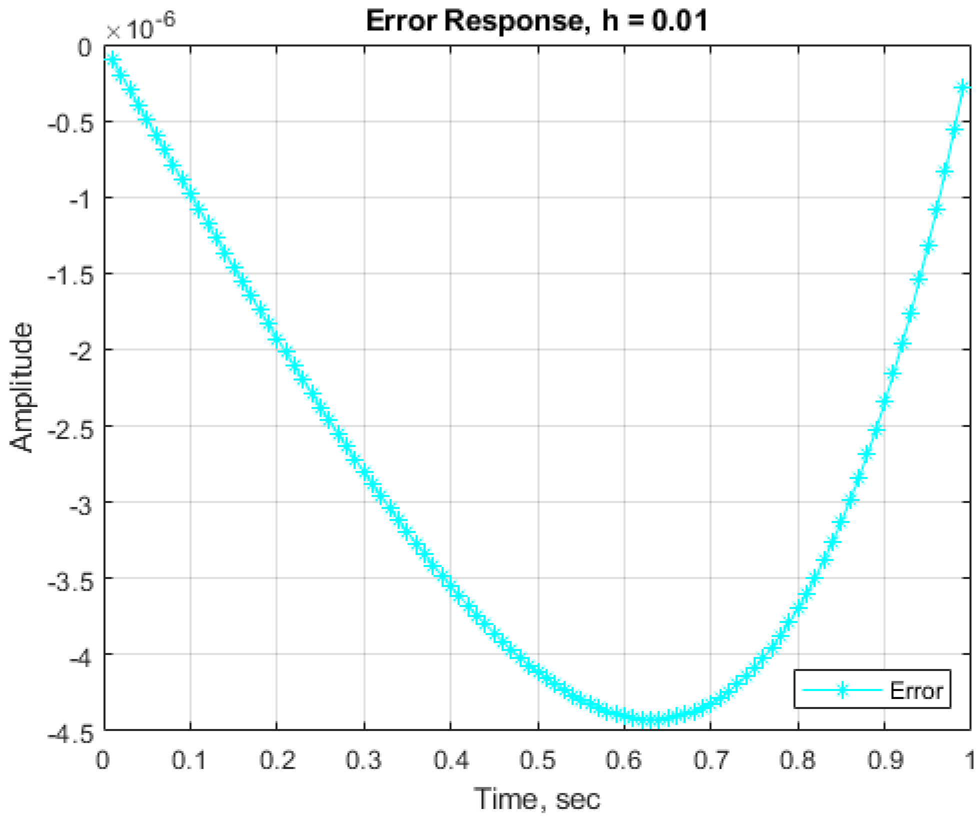
```
4.4252e-06
```



Beam Deflection: Exact vs 2nd-FD, h = 0.10
Amplitude
Time, sec
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Exact
2nd FD



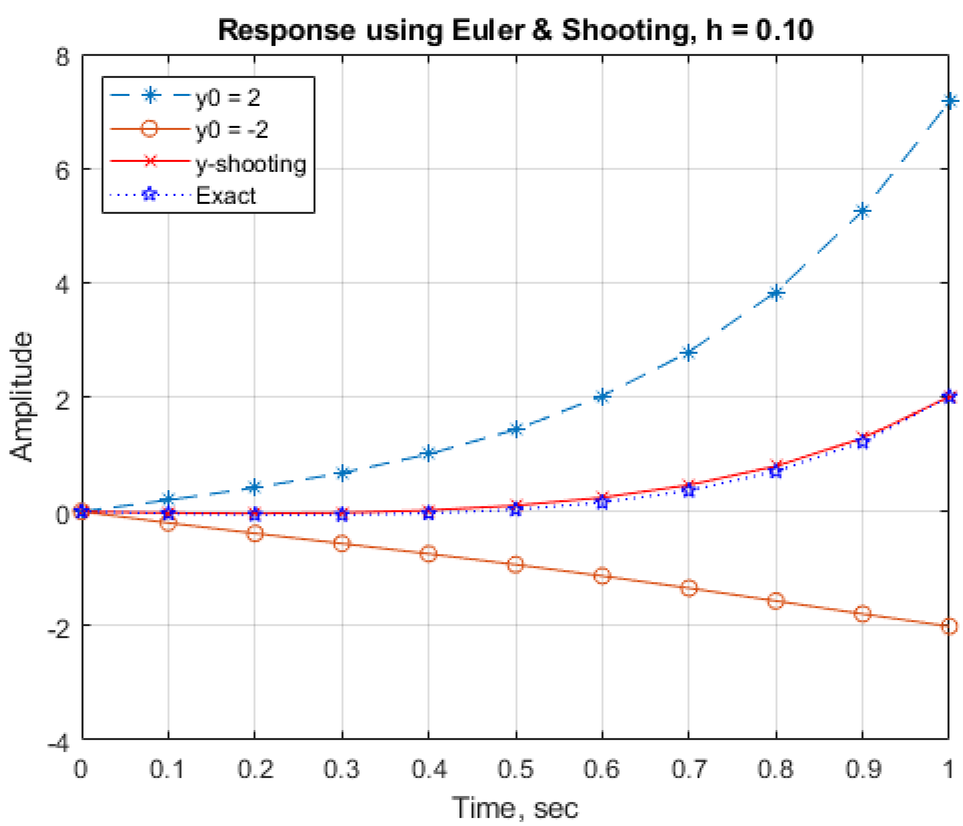
Beam Deflection: Exact vs 2nd-FD, h = 0.01
Amplitude
Time, sec
Exact
2nd FD
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.2
1.4
1.6
1.8
2

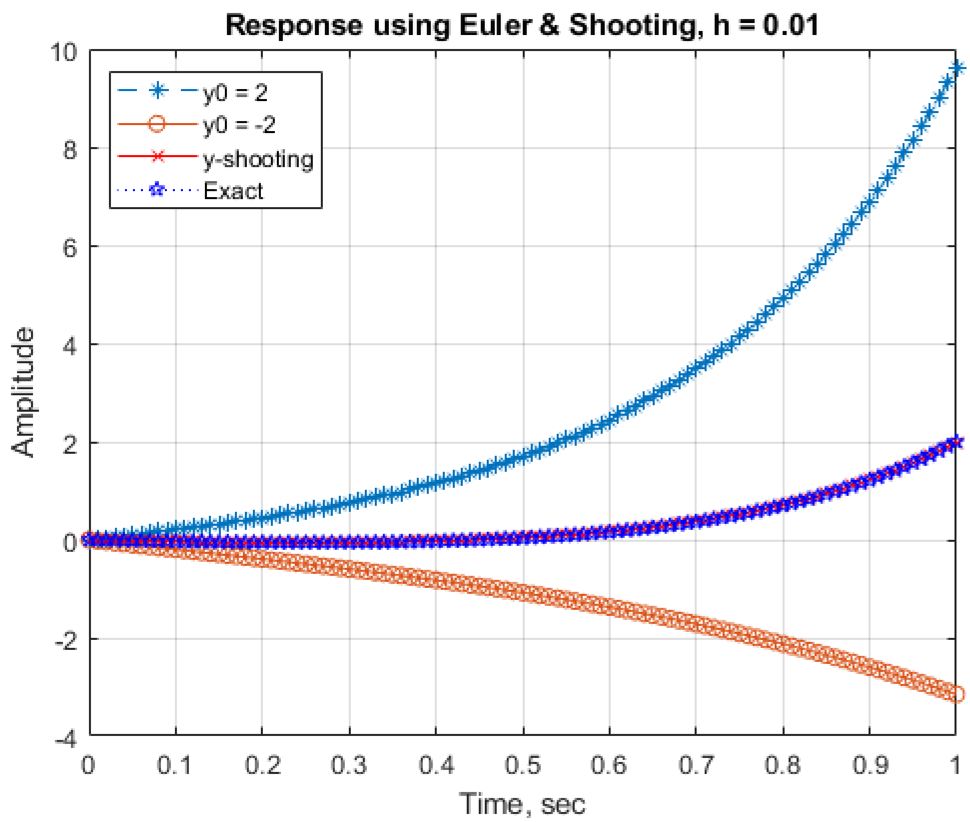


Problem #2 - Boundary Value Problem

```
clear
clc

for j = 7:8
    figure(j)
    h_to_use = [0.1 0.01];
    h = h_to_use(j-6);
    t = 0:h:1;
    y_n_store = zeros(2,length(t));

    % Exact
    b_Exact_1 = [1 1;exp(-3) exp(3)];
    b_Exact_2 = [0;2+(1/18)*sin(3)-(1/6)*exp(3)];
    b_Exact_3 = b_Exact_1\b_Exact_2;

    b_y_Exact = @(t) b_Exact_3(1).*exp(-3.*t) + b_Exact_3(2).*exp(3.*t)...
                 + (1/6).*t.*exp(3.*t) - (1/18).*sin(3.*t);

    % Shooting Method - 2 Initial Contdition
    for k = 1:2
        IC_to_use = [2 -2];
        IC = [0 IC_to_use(k)]';
        y_n = zeros(2,length(t));
```

```
        y_n(:,1) = IC;
        for i = 2:length(t)
            y_n(:,i) = y_n(:,i-1) + h*[y_n(2,i-1);9*y_n(1,i-1)+exp(3*t(i))+sin(3*t(i))];
        end
        y_n_store(k,:) = y_n(1,:);
    end

    % Linear Shooting Method
    y_boundary = 2;
    c1 = (y_boundary - y_n_store(2,end))/(y_n_store(1,end) - y_n_store(2,end));
    c2 = (y_n_store(1,end) - y_boundary)/(y_n_store(1,end) - y_n_store(2,end));
    y_linear = c1*y_n_store(1,:) + c2*y_n_store(2,:);

    % Plot
    plot(t,y_n_store(1,:),'*--',t,y_n_store(2,:),'o-'...
        ,t,y_linear,'rx-',t,b_y_Exact(t),'bp:')
    grid on
    xlabel('Time, sec')
    ylabel('Amplitude')
    legend('y0 = 2','y0 = -2','y-shooting','Exact','Location','northwest')
    title(sprintf('Response using Euler & Shooting, h = %.2f',h))
end
```



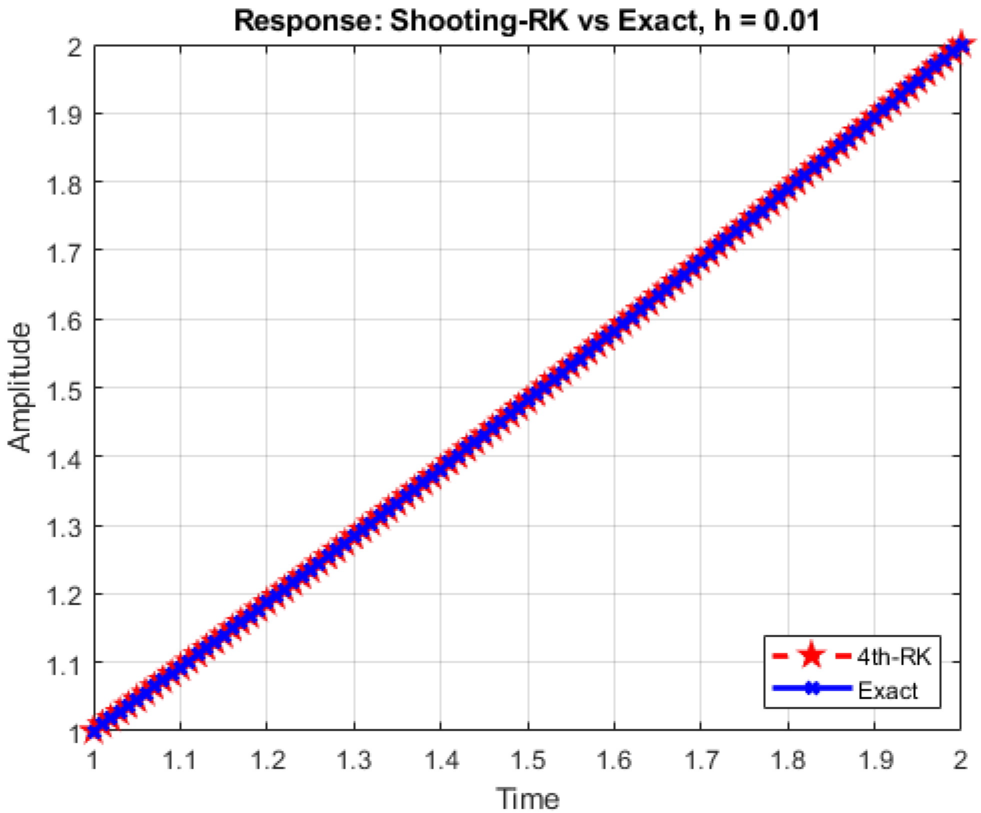


Problem #3 - Boundary Value Problem

```
clear
clc

c2 = (1/70)*((8-(12*sin(log(2)))) - (4*cos(log(2))));
c1 = (11/10) - c2;
y_exact = @(t) c1.*t + c2.*(1./(t.^2)) - (3/10).*sin(log(t)) - (1/10).*cos(log(t));

for j = 9:10
    figure(j)
    h_to_use = [0.1 0.01];
    h = h_to_use(j-8);
    t = 1:h:2;

    % Finding y1
    y1 = zeros(2,length(t));
    y1(:,1) = [1 0]';
    for i = 2:length(t)

        z1 = y1(1,i-1);
        z2 = y1(2,i-1);
        time = t(i-1);

        z1_prime = @(z2) z2;
```

```
    z2_prime = @(z1,z2,t) (2/((t^2)))*z1 + (-2/t)*z2...
                + sin(log(t))/(t^2);

    k1 = h*[z1_prime(z2);...
            z2_prime(z1,z2,time)];

    k2_input = [z1+(0.5*k1(1));...
                z2+(0.5*k1(2))];
    k2 = h*[z1_prime(k2_input(2));...
            z2_prime(k2_input(1),...
            k2_input(2),time+(0.5*h))];

    k3_input = [z1+(0.5*k2(1));...
                z2+(0.5*k2(2))];
    k3 = h*[z1_prime(k3_input(2));...
            z2_prime(k3_input(1),...
            k3_input(2),time+(0.5*h))];

    k4_input = [z1+k3(1);...
                z2+k3(2)];
    k4 = h*[z1_prime(k4_input(2));...
            z2_prime(k4_input(1),...
            k4_input(2),time+h)];

    y1(:,i) = y1(:,i-1) + (1/6)*k1 + (1/3)*(k2+k3) + (1/6)*k4;
end

% Finding y2
y2 = zeros(2,length(t));
y2(:,1) = [0 1]';
for k = 2:length(t)

    u1 = y2(1,k-1);
    u2 = y2(2,k-1);
    time = t(k-1);

    u1_prime = @(u2) u2;
    u2_prime = @(u1,u2,t) (2/((t^2)))*u1 + (-2/t)*u2;

    k1 = h*[u1_prime(u2);...
            u2_prime(u1,u2,time)];

    k2_input = [u1+(0.5*k1(1));...
                u2+(0.5*k1(2))];
    k2 = h*[u1_prime(k2_input(2));...
            u2_prime(k2_input(1),...
            k2_input(2),time+(0.5*h))];

    k3_input = [u1+(0.5*k2(1));...
                u2+(0.5*k2(2))];
    k3 = h*[u1_prime(k3_input(2));...
            u2_prime(k3_input(1),...
```

```
                k3_input(2),time+(0.5*h))];

        k4_input = [u1+k3(1);...
                    u2+k3(2)];
        k4 = h*[u1_prime(k4_input(2));...
                u2_prime(k4_input(1),...
                k4_input(2),time+h)];

        y2(:,k) = y2(:,k-1) + (1/6)*k1 + (1/3)*(k2+k3) + (1/6)*k4;
    end

    % Solution y
    beta = 2;
    const = ((beta - y1(1,end))/y2(1,end));
    y = y1(1,:) + const*y2(1,:);

    % Plot
    plot(t,y,'rp--',t,y_exact(t),'bx-','linewidth',2)
    grid on
    xlabel('Time')
    ylabel('Amplitude')
    legend('4th-RK','Exact','location','southeast')
    title(sprintf('Response: Shooting-RK vs Exact, h = %.2f',h))
end
```



Published with MATLAB® R2018b