ME 7120: Finite Element Method Applications

**Project III**

By

**Hao Li,**

**Obidigbo Chigozie**

**Mohammed Al Rifaie,**

**Prof:** Dr. Slater

**Submitted:** December 9nd, 2016

Table of Contents

[Nomenclature i](#_Toc468285453)

[Project Description **Error! Bookmark not defined.**](#_Toc468285455)

[Methodology 3](#_Toc468285455)

[Newmark Method 6](#_Toc468285455)

[Results 8](#_Toc468285455)

[Conclusion 15](#_Toc468285463)

[Appendix 16](#_Toc468285464)

# Nomenclature

𝟂 = Natural Frequency

Ω = Critcal frequency

𝛽 = Control characteristics

γ = Control characteristics

, = Displacement

, = Velocity

. = Acceleration

ξ = Damping ratio

P = Load

L = Length

A = Cross Sectional Area

E = Young’s Modulus of Elasticity

t = Time

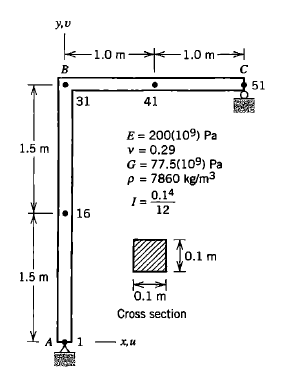
F = Force

K = Stiffness Matrix

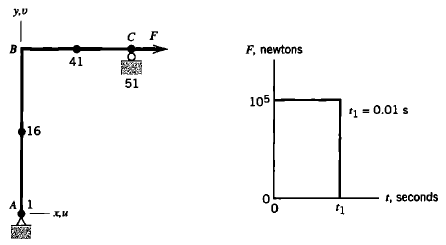
M = Mass Matrix

C = Damping Matrix

**Project Description**

For this project III, Finite Element Method was used to formulate and solve a time dependent structural problem. The stiffness matrix K and mass matrix M were obtained using WFEM. The objective is to formulate the damping matrix C using the stiffness matrix and mass matrix for the L-shaped structure of fig 11.17-1 from the text book “Concepts and Applications of Finite Element Analysis, 4th Edition, Wiley, 2001”, as shown in Figure 1.

*Figure 1: Plane structure and its properties*

Using WFEM gives a size of [306x306] for M and K because there are 6 DOFs at each node. For 2D problem, there are one DOF at node 1, 2 DOFs at node 51 and three DOFs at nodes (2-50) so that the size becomes [150x150]. The 3D stiffness and mass matrices are reduced to 2D and the boundary conditions are applied using the find\_C.m function in Matlab. This function also generates the C matrix after solving eigenvalue and eigenvector problem. After which we use the integration method of Newmark beta to calculate for the transient response of the step loading of the system with a force F = N over a total time t = 0.01s, calculating the acceleration , velocity , and displacement , of the system over this time period, as shown in Figure 2.

*Firgure 2: Frame loaded at point C by a horizontal force F and the Prescribed variation of force F with time t.*

Finally, a comparison of the five Newmark beta methods are used to compare the responses generated using WFEM with those gotten from the text book.

**Methodology**

The equation of motion that we are going to solve for the problem is shown below

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

where is the mass matrix, is the damping matrix, is the stiffness matrix. , and are displacement, velocity, and acceleration. is the applied load.

1. **Undamped system with no loading**

First, Let us consider the undamped case with no loading in the system. Reduce the problem from 3D to 2D and apply the boundary condition, the reduced equation of motion is obtained with reduced mass and stiffness matrix shown below

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Perform the Cholesky decomposition on the reduced mass matrix

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

Let and pre-multiply to the reduced equation of motion

|  |  |  |
| --- | --- | --- |
|  |  | (4) |
|  |  | (5) |

where .

Let and submit it into Eq. (5). The eigenvalue problem is obtained below

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Let , where is the eigenvector for each mode.

Let and submit it into Eq. (5), and pre-multiply , we got

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

This ends up with the decoupled system

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

where

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Then we know

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

The mode shape is calculated by the following equation

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

1. **damped system with loading**

Let us consider the damped system with loading shown below

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

Submit Eq. (10) and pre-multiply , we got

|  |  |  |
| --- | --- | --- |
|  |  | (13) |
|  |  | (14) |

where

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

Then, we can get the matrix by the following equation

|  |  |  |
| --- | --- | --- |
|  |  | (16) |
|  |  |  |

**Newmark Method**

This is used to analysis the transient response of the system. The Newmark Method can be summarized in the following steps:

Step 1: From the known values of , , find .

Step 2: Select suitable values of , and based on Table 1.

|  |  |  |
| --- | --- | --- |
|  |  | *(17)* |

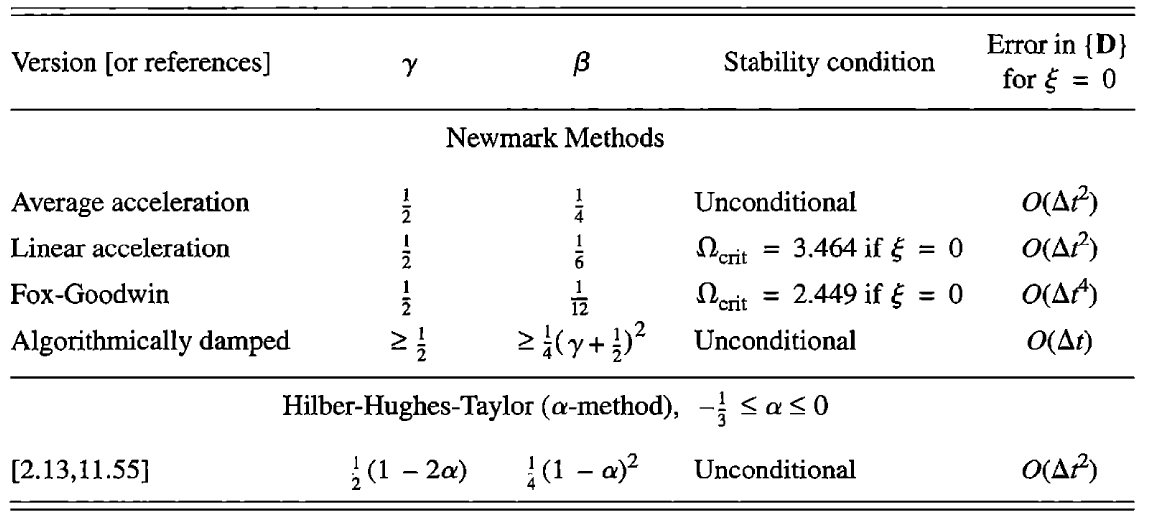
Step 3: Calculate the displacement vector , using the following expression

Step 4: Find the acceleration and velocity vectors at time based on the following equations

|  |  |  |
| --- | --- | --- |
|  |  | (18) |
|  |  | (19) |

Step 5: Repeat

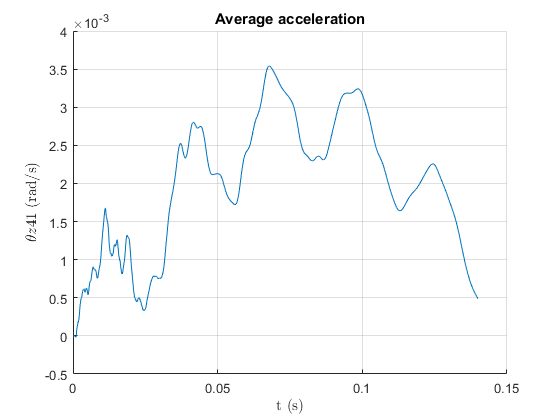
Table 1. Stability and accuracy of different Newmark Methods



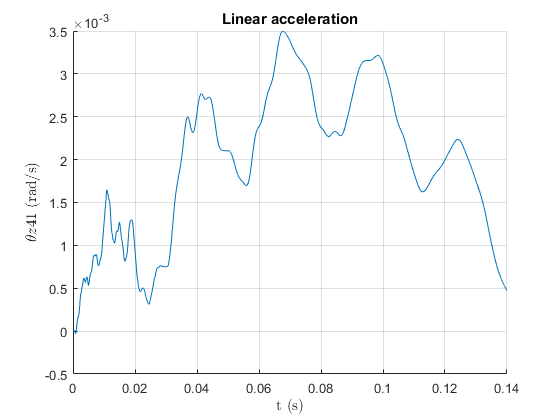
Note: For accuracy ≠ {0}, doing so may reduce accuracy from second order to first order.

{} = ( – [K]{} – [C]{)

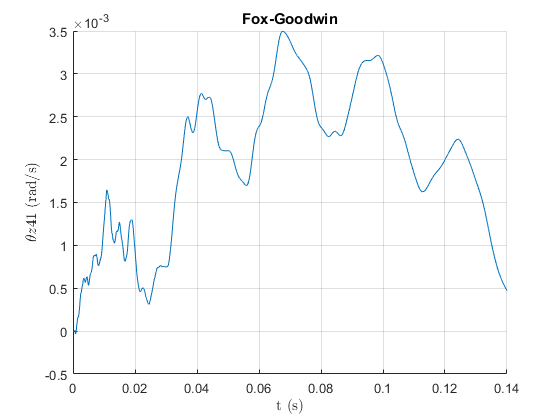
**Results**

The next 6 figures show the rotation θz at node 41 vs time(s) for 5 methods compared with text book, ξ=0.02.

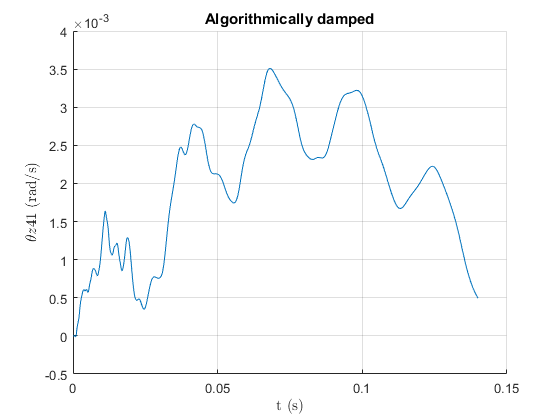
*Figure3: Average acceleration,* ξ *=0.02*



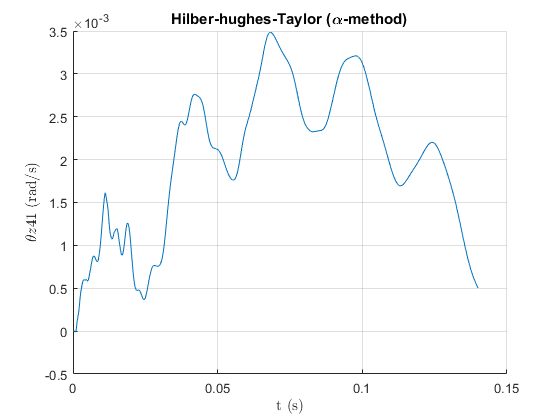
*Figure4: Linear acceleration,* ξ *=0.02*



*Figure 5: Fox-Goodwin,* ξ *=0.02*

**

*Figure 6: Algorithmically damped,* ξ *=0.02*



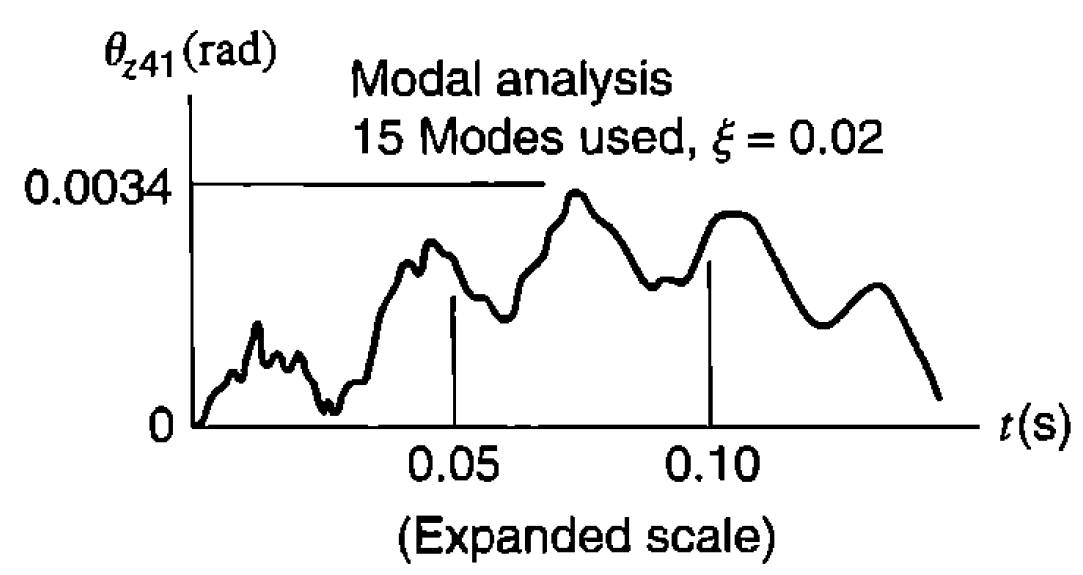
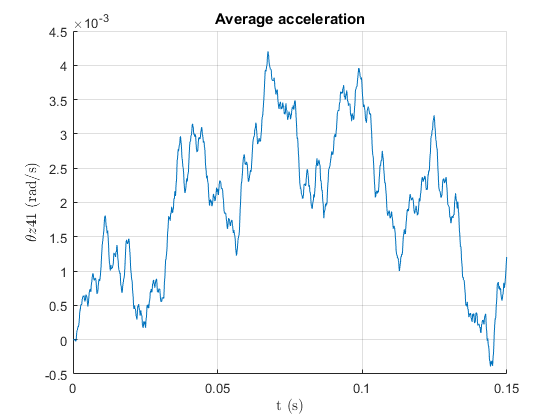
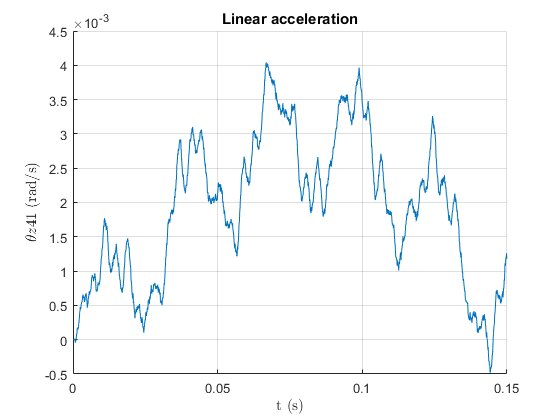
*Figure 7: Hilber-hughes-Taylor (α-method),* ξ *=0.02*

Figure 8: Text book figure, ξ =0.02

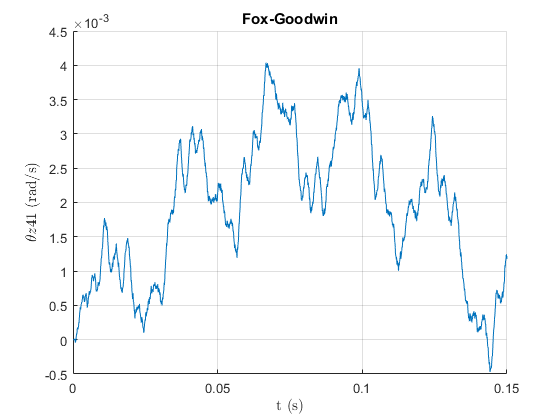
The next 6 figures show the rotation θz at node 41 vs time(s) for 5 methods compared with text book when ξ =0.



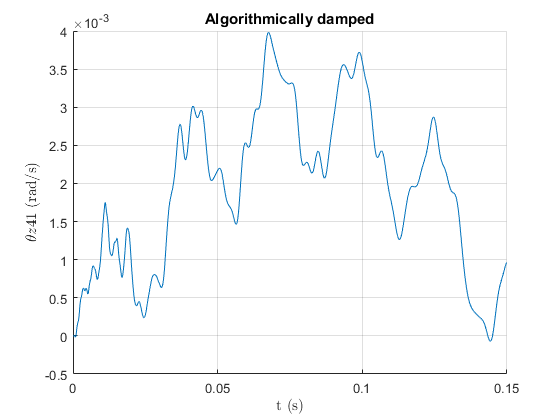
*Figure 9: Average acceleration, ξ = 0*



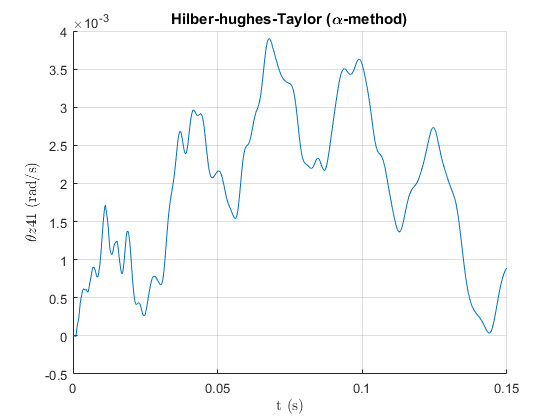
*Figure 10: Linear acceleration, ξ =0*



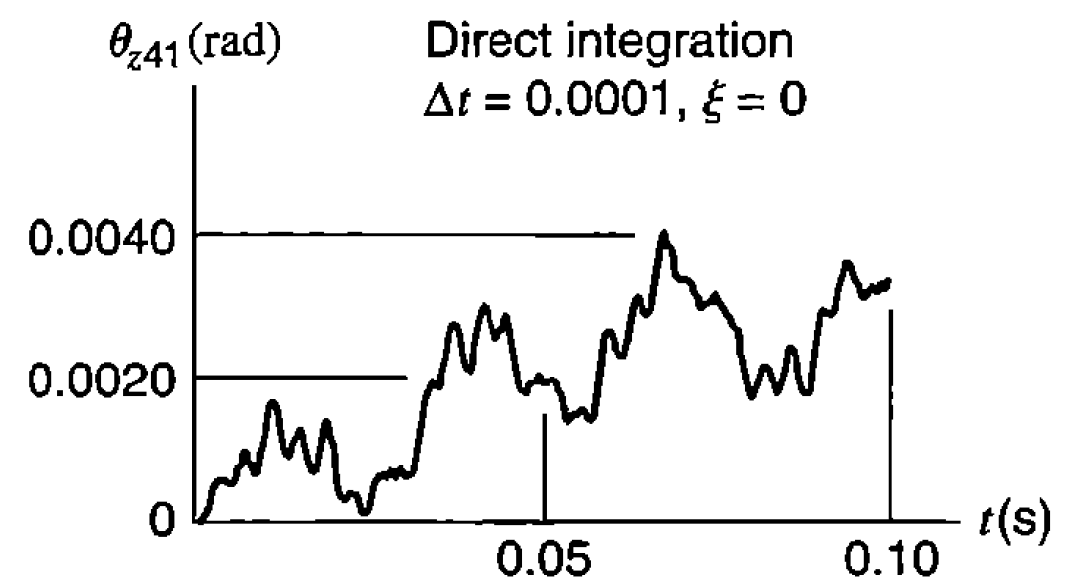
*Figure 11: Fox-Goodwin,* ξ = 0



*Figure 12: Algorithmically damped,* ξ = 0



*Figure 13: Hilber-hughes-Taylor (α-method), ξ = 0*



*Figure 14: Text book figure,* ξ = 0

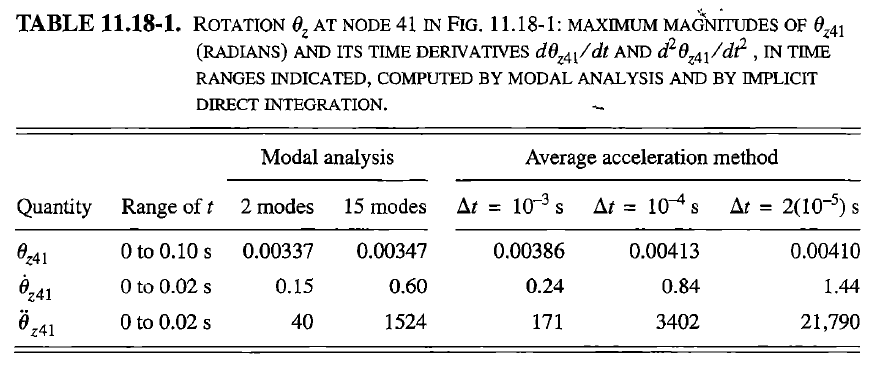


Table 3: From WFEM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Qty.** | **Range of t** | **Avg. acc.**  **∆t=10-4** | **Linear acc. ∆t=10-4** | **Fox Goodwin ∆t=10-4** | **Algorithmically damped ∆t=10-4** | **Α**  **method ∆t=10-4** |
| θz41 | 0 to 0.1s | 0.00355 | 0.003496 | 0.003498 | 0.00352 | 0.003494 |
| θz41 | 0 to 0.02s | 0.68 | 0.7317 | 0.7321 | 0.4692 | 0.4354 |
| θz41 | 0 to 0.02s | 1.303e4 | 1.678e4 | 1.671e4 | 2024 | 1448 |

**Conclusion**

The project was to show how most modern FEA solve dynamic analysis in its back end. The results for the modal analysis of the structure were close to those from the text book as can be seen in the attached pictures of mode shapes in the zipped file. In calculating the structures response to force F over time t using the Newmark method ≠ {0} but rather = (( – [K] {} – [C]{)) for accuracy. Using the Hilber-Hughes-Taylor method gives you the same results as the average acceleration method if the α is set equal to 0.

**Appendix**

* find\_C.mat
* Newmark\_method.mat

The rest of the matlab codes and images are included in a zipped folder.

**find\_C.mat**

function [Kr Mr C wmax] = find\_C(K,M)

% This function is for finding reduced M & K matrieces and obtaining C matrix

NumNode = 51; % Number of Nodes

%% K and M in 3D (306\*306)

K\_3D = full(K); % Global Stiffness Matrix

M\_3D = full(M); % Global Mass Matrix

%% Reduce K and M from 3D to 2D

dof\_2D = []; % Dof Number in 2D case

for i = 1: NumNode

dof\_2D = [dof\_2D, [1,2,6]+6\*(i-1)];

end

dof\_2D;

% K and M in 2D (153\*153)

K\_2D = K\_3D(dof\_2D,dof\_2D);

M\_2D = M\_3D(dof\_2D,dof\_2D);

%% Apply BCs (DOF1 = DOF2 = DOF152 = 0)

dof\_BC = [1,2,152];

dof\_all = 1:length(K\_2D);

for i = 1:length(dof\_BC)

index = find(dof\_all == dof\_BC(i));

dof\_all(index) = [];

end

% Reduced K and M

Kr = K\_2D(dof\_all,dof\_all);

Mr = M\_2D(dof\_all,dof\_all);

%% Calculate Phi

L = chol(Mr)';

K\_h = L\Kr/L';

[vectors\_V, values]=eig(K\_h);

% Sort the eigenvalues and eigenvectors in ascending order

[values, index] = sort(diag(values));

vectors\_V = vectors\_V(:,index);

w = sqrt(values); % natural frequencies

wmax=w(150); % Omega max

Phi = (L')\vectors\_V;

%% Calculate C matrix

zeta = 0.02;

C = Phi'\diag(2\*zeta\*w)/Phi;

end

Newmark\_method.mat

% Newmark method

load K\_M.mat % Load M and K matrieces

[Kr, Mr, C, wmax] = find\_C(K,M);

Inv=eye(150,150); % Identity matrix will be used to take inverse of [150x150] matrix

% Average acceleration

beta = 1/4; % beta

gamma = 1/2; % gamma

% % Linear acceleration

% beta = 1/6; % beta

% gamma = 1/2; % gamma

% % Fox-Goodwin

% beta = 1/12; % beta

% gamma = 1/2; % gamma

% % Algorithmically damped

% gamma = 0.6; % gamma

% beta = (1/4)\*(gamma+0.5)^2; % beta

% % Hilber-hughes-Taylor (alpha-method)

% alpha =-0.2; % -1/3 <= alpha <=0

% beta = (1/4)\*(1-alpha)^2; % beta

% gamma = 0.7; % gamma

% % Finding dt using omega critical. use this for Linear acceleration and

% % Fox-Goodwin methods.

% Z=0.02; % Damping ratio

% Ocrit=(Z\*(gamma-0.5)+sqrt((gamma/2)-beta+(Z^2)\*(gamma-0.5)^2))/(gamma/2-beta);

% dt=Ocrit/wmax;

dt = 0.0001; % delta t. use this for above 3 methods (1st, 4th and 5th).

tf = 0.15; % Final t

n=floor(tf/dt); % Steps

t=zeros(n,1); % Time

D=zeros(150,n); % Displacement

DD=zeros(150,n); % Velocity

DDD=zeros(150,n); % Acceleration

% Initial conditions for Disp., vel., and acc.

R0=zeros(150,1); % Force, Rt

R0(149,1) = 100000;

D(:,1) = zeros(150,1);

DD(:,1) = zeros(150,1);

DDD(:,1) = Mr\R0;

Rt=zeros(150,1); % Force, Rt

for i=1:n

% Impulse loading applied at node 51 starts form t=0 to t=0.01s.

if t(i)<=0.01

Rt(149,1) = 100000;

else

Rt(149,1) = 0;

end

D(:,i+1) = (((1/(beta\*dt^2))\*Mr +(gamma/(beta\*dt))\*C+ Kr )\Inv )\*(Rt +...

Mr\*( (1/(beta\*dt^2))\*D(:,i) + (1/(beta\*dt))\*DD(:,i) + (1/(2\*beta)-1)\*DDD(:,i) )...

+ C\*((gamma/(beta\*dt))\*D(:,i)+(gamma/beta-1)\*DD(:,i)+(gamma/beta-2)\*(dt/2)\*DDD(:,i)));

DDD(:,i+1) = (1/(beta\*dt^2))\*( D(:,i+1) - D(:,i) - dt\*DD(:,i) ) - (1/(2\*beta) - 1)\*DDD(:,i);

DD(:,i+1) = (gamma/(beta\*dt))\*(D(:,i+1) - D(:,i)) - (gamma/beta - 1)\*DD(:,i) -...

dt\*(gamma/(2\*beta) - 1)\*DDD(:,i);

t(i+1)=t(i)+dt;

end

figure(1)

grid on; hold on

plot(t,D(121,:)) % Displacement of the rotational DOF at node 41

ylabel('$\theta z41$ (rad/s)','interpreter','latex')

xlabel('t (s)','interpreter','latex')

title('Average acceleration')

figure(2)

grid on; hold on

plot(t,DD(121,:)) % Velocity of the rotational DOF at node 41

ylabel('$\dot{\theta}z41$ (rad/s)','interpreter','latex')

xlabel('t (s)','interpreter','latex')

title('Average acceleration')

figure(3)

grid on; hold on

plot(t,DDD(121,:)) % Acceleration of the rotational DOF at node 41

ylabel('$\ddot{\theta}z41$ (rad/s)','interpreter','latex')

xlabel('t (s)','interpreter','latex')

title('Average acceleration')