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

$\boldsymbol{\omega}$	=	Natural Frequency
Ω	=	Critcal frequency
β	=	Control characteristics
γ	=	Control characteristics
D_0,	=	Displacement
$\dot{D}_0$,	=	Velocity
$\ddot{D}_0$.	=	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.

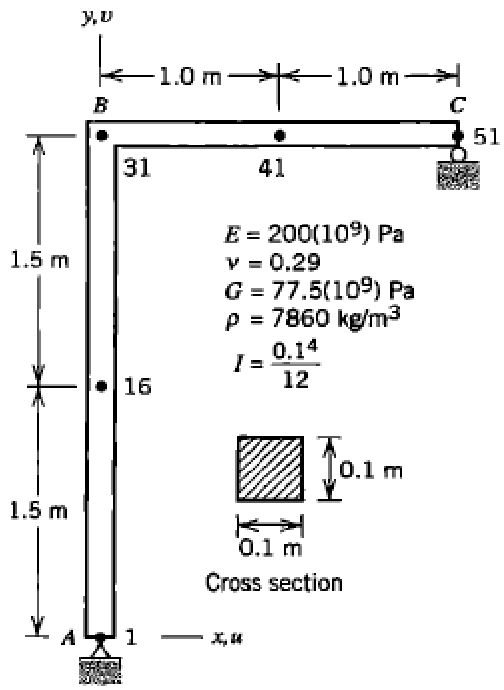
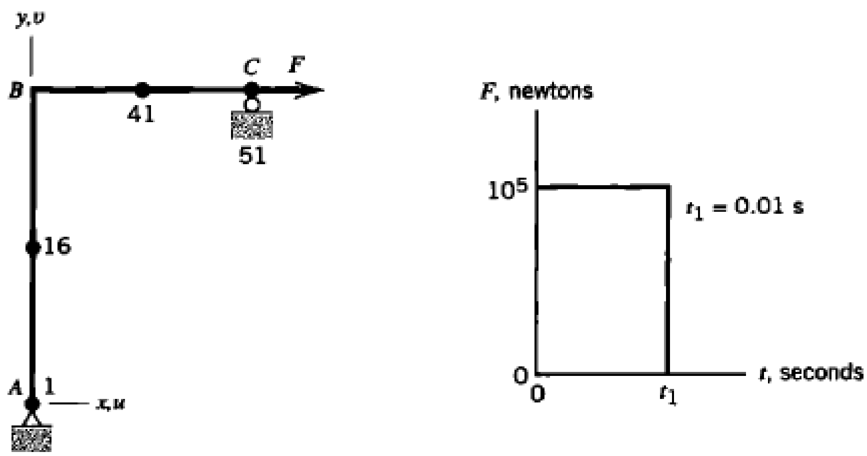


Figure 1: Plane structure and its properties

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 = 10^5 N over a total time t = 0.01s, calculating the acceleration $\ddot{D}$, velocity $\dot{D}$, and displacement D , 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

$$M\ddot{x} + C\dot{x} + Kx = f \tag{1)}$$

where M is the mass matrix, C is the damping matrix, K is the stiffness matrix. x, $\dot{x}$ and $\ddot{x}$ are displacement, velocity, and acceleration. f 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

$$M_r\ddot{x} + K_r x = 0 \tag{2}$$

Perform the Cholesky decomposition on the reduced mass matrix M_r

$$M_r = LL^T \tag{3}$$

Let $x = (L^T)^{-1}q$ and pre-multiply L^{-1} to the reduced equation of motion

$$L^{-1}M_r(L^T)^{-1}\ddot{q} + L^{-1}K_r(L^T)^{-1}q = 0 \tag{4}$$

$$I\ddot{q} + \widetilde{K}q = 0 \tag{5}$$

where $\widetilde{K} = L^{-1}K_r(L^T)^{-1}$.

Let $q = ve^{\omega t}$ and submit it into Eq. (5). The eigenvalue problem is obtained below

$$\widetilde{K}v = \omega^2 v \tag{6}$$

Let $V = [v_1, v_2, \dots, v_n]$, where v_i is the eigenvector for each mode.

Let $q = Vr$ and submit it into Eq. (5), and pre-multiply V^T, we got

$$V^T V\ddot{r} + V^T \widetilde{K} V r = 0 \tag{7}$$

This ends up with the decoupled system

$$\ddot{r} + \Lambda r = 0 \tag{8}$$

where

$$\Lambda = \begin{bmatrix} \omega_1^2 & & & \\ & \omega_2^2 & & \\ & & \dots & \\ & & & \omega_n^2 \end{bmatrix} \tag{9}$$

Then we know

$$x = (L^T)^{-1} q = (L^T)^{-1} V r = \Phi r \tag{10}$$

The mode shape is calculated by the following equation

$$\Phi = [\phi_1, \phi_2, \dots, \phi_n] = (L^T)^{-1} V \tag{11}$$

2. damped system with loading

Let us consider the damped system with loading shown below

$$M_r \ddot{x} + C_r \dot{x} + K_r x = f_r \tag{12}$$

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

$$\Phi^T M_r \Phi \ddot{r} + \Phi^T C_r \Phi \dot{r} + \Phi^T K_r \Phi r = \Phi^T f_r \tag{13}$$

$$I\ddot{r} + \Omega\dot{r} + \Lambda r = \Phi^T f_r \tag{14}$$

where

$$\Omega = \begin{bmatrix} 2\xi\omega_1 & & & \\ & 2\xi\omega_2 & & \\ & & \cdots & \\ & & & 2\xi\omega_n \end{bmatrix} \tag{15}$$

Then, we can get the C_r matrix by the following equation

$$C_r = (\Phi^T)^{-1}\Omega\Phi^{-1} \tag{16}$$

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 D_0, $\dot{D}_0$, find $\ddot{D}_0$.

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

Step 3: Calculate the displacement vector $D(t+\Delta t)$, using the following expression

$$\mathrm{D}(\mathrm{t}+\Delta t) = \left[\frac{1}{\beta \Delta t^2} M_r + \frac{\gamma}{\beta \Delta t} C_r + K_r\right]^{-1} \times \left[R(t+\Delta t) + M_r\left(\frac{1}{\beta \Delta t^2} D(t) + \frac{1}{\beta \Delta t}\dot{D}(t) + \left(\frac{1}{2\beta} - 1\right)\ddot{D}(t)\right) + C_r\left(\frac{\gamma}{\beta \Delta t} D(t) + \left(\frac{\gamma}{\beta} - 1\right)\dot{D}(t) + \left(\frac{\gamma}{\beta} - 2\right)\frac{\Delta t}{2}\ddot{D}(t)\right)\right] \tag{17}$$

Step 4: Find the acceleration and velocity vectors at time $t+\Delta t$ based on the following equations

$$\ddot{D}(t+\Delta t) = \frac{1}{\beta \Delta t^2}\left(\mathrm{D}(\mathrm{t}+\Delta t) - \mathrm{D}(\mathrm{t}) - \Delta t \dot{D}(t)\right) - \left(\frac{1}{2\beta} - 1\right)\ddot{D}(t) \tag{18}$$

$$\dot{D}(t+\Delta t)=\frac{\gamma}{\beta\Delta t}(\mathrm{D}(\mathrm{t}+\Delta t)-D(t))-\left(\frac{\gamma}{\beta}-1\right)\dot{D}(t) \\ -\Delta t\left(\frac{\gamma}{2\beta}-1\right)\ddot{D}(t) \tag{19}$$

Step 5: Repeat

Table 1. Stability and accuracy of different Newmark Methods

Version [or references]	γ	β	Stability condition	Error in $\{\mathbf{D}\}$ for $\xi = 0$
		Newmark Methods		
Average acceleration	$\frac{1}{2}$	$\frac{1}{4}$	Unconditional	$O(\Delta t^2)$
Linear acceleration	$\frac{1}{2}$	$\frac{1}{6}$	$\Omega_{crit} = 3.464$ if $\xi = 0$	$O(\Delta t^2)$
Fox-Goodwin	$\frac{1}{2}$	$\frac{1}{12}$	$\Omega_{crit} = 2.449$ if $\xi = 0$	$O(\Delta t^4)$
Algorithmically damped	$\geq \frac{1}{2}$	$\geq \frac{1}{4}(\gamma+\frac{1}{2})^2$	Unconditional	$O(\Delta t)$
		Hilber-Hughes-Taylor (α-method), $-\frac{1}{3} \leq \alpha \leq 0$		
[2.13,11.55]	$\frac{1}{2}(1-2\alpha)$	$\frac{1}{4}(1-\alpha)^2$	Unconditional	$O(\Delta t^2)$

Note: For accuracy $\ddot{D}_0 \neq \{0\}$, doing so may reduce accuracy from second order to first order.

$$\{\ddot{D}_0\} = [\mathrm{M}]^{-1}(\{\mathrm{R}^{\mathrm{ext}}\}_0 - [\mathrm{K}]\{D_0\} - [\mathrm{C}]\{\dot{D}_0\})$$

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.

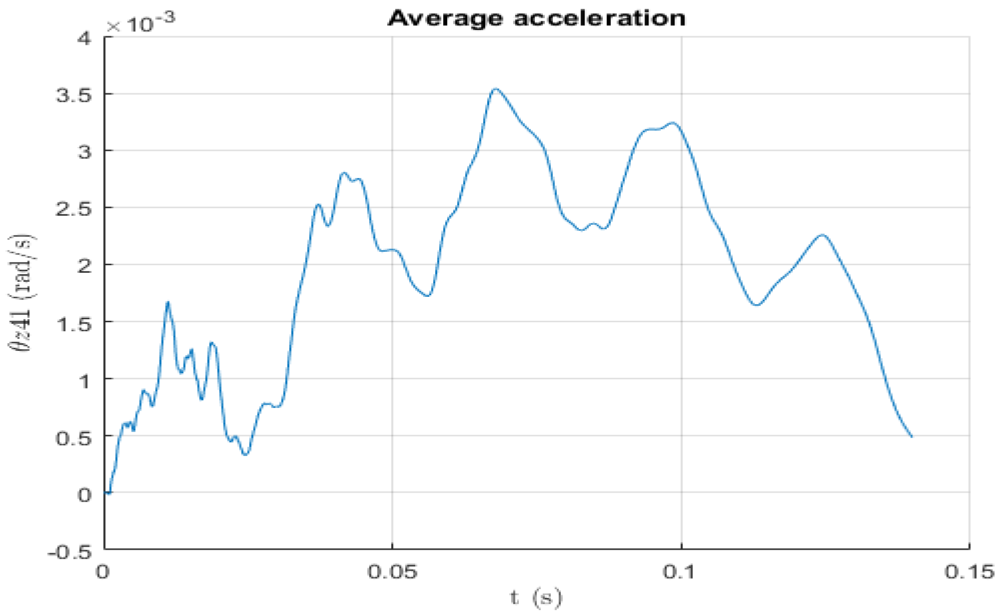


Figure3: Average acceleration, ξ =0.02

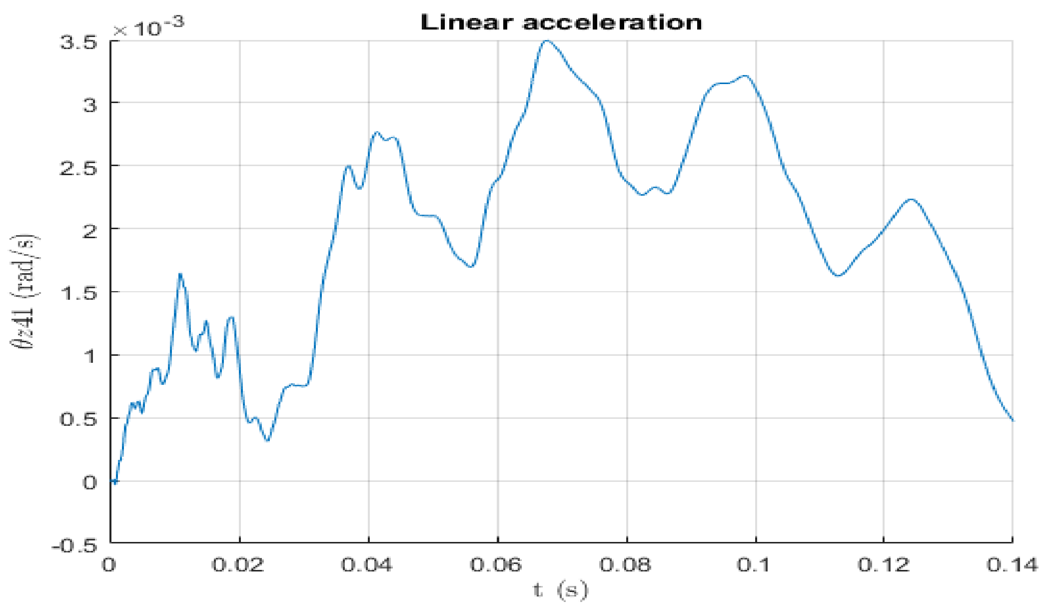


Figure4: Linear acceleration, ξ =0.02

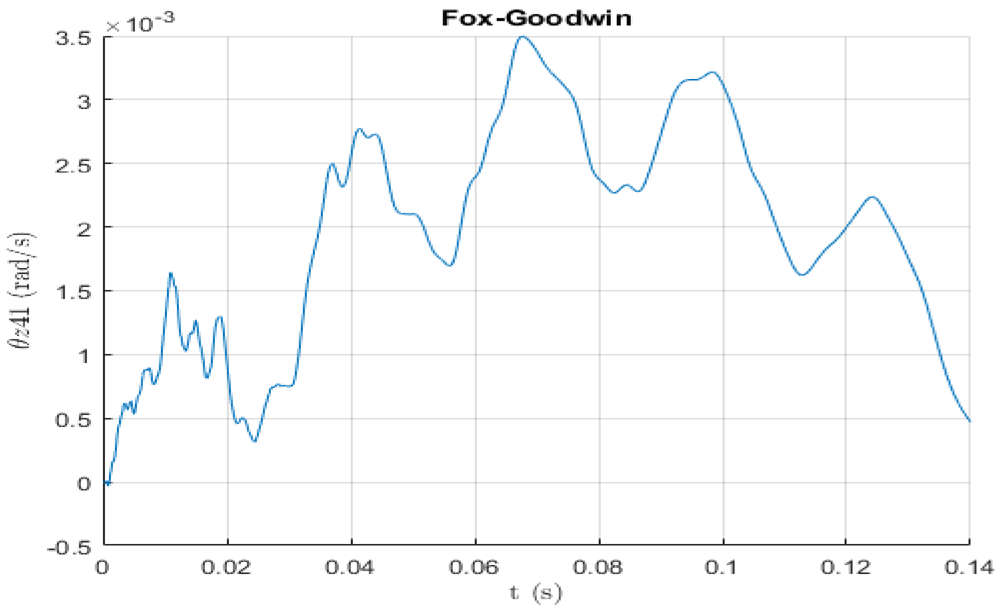


Figure 5: Fox-Goodwin, ξ =0.02

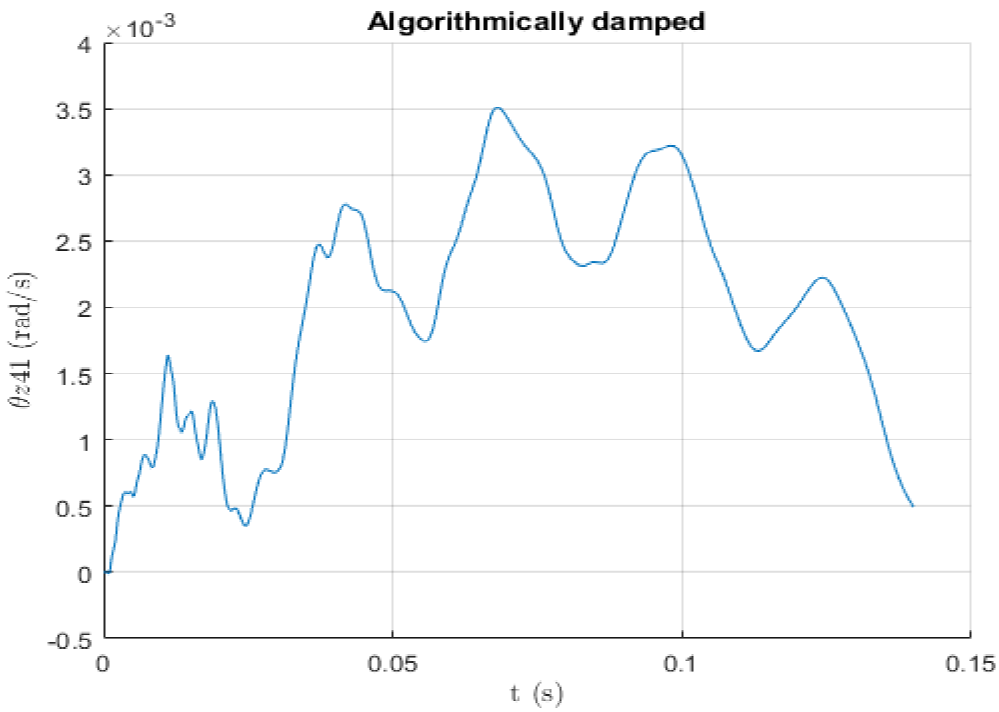


Figure 6: Algorithmically damped, ξ =0.02

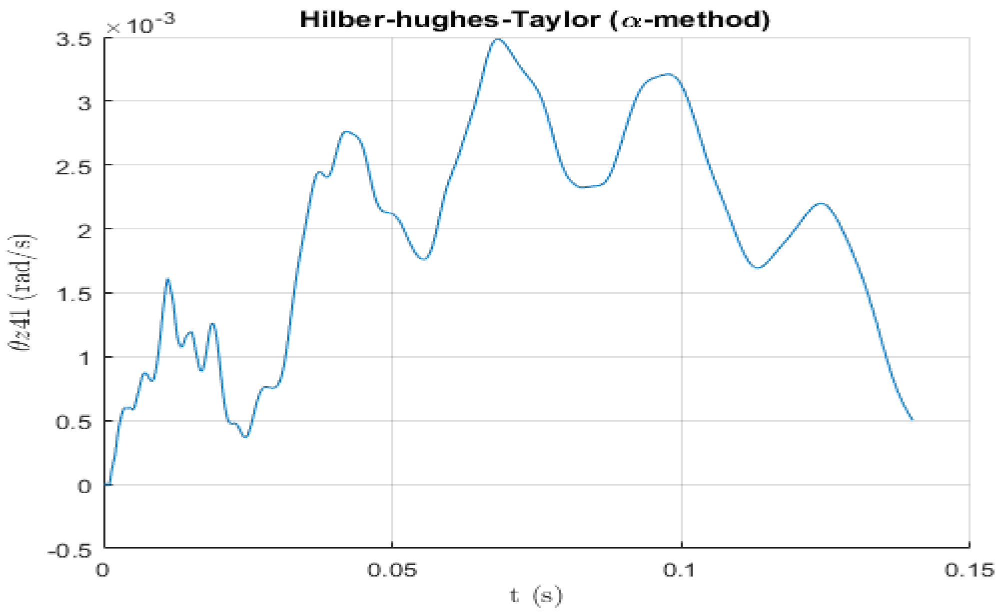


Figure 7: Hilber-hughes-Taylor (α-method), ξ =0.02

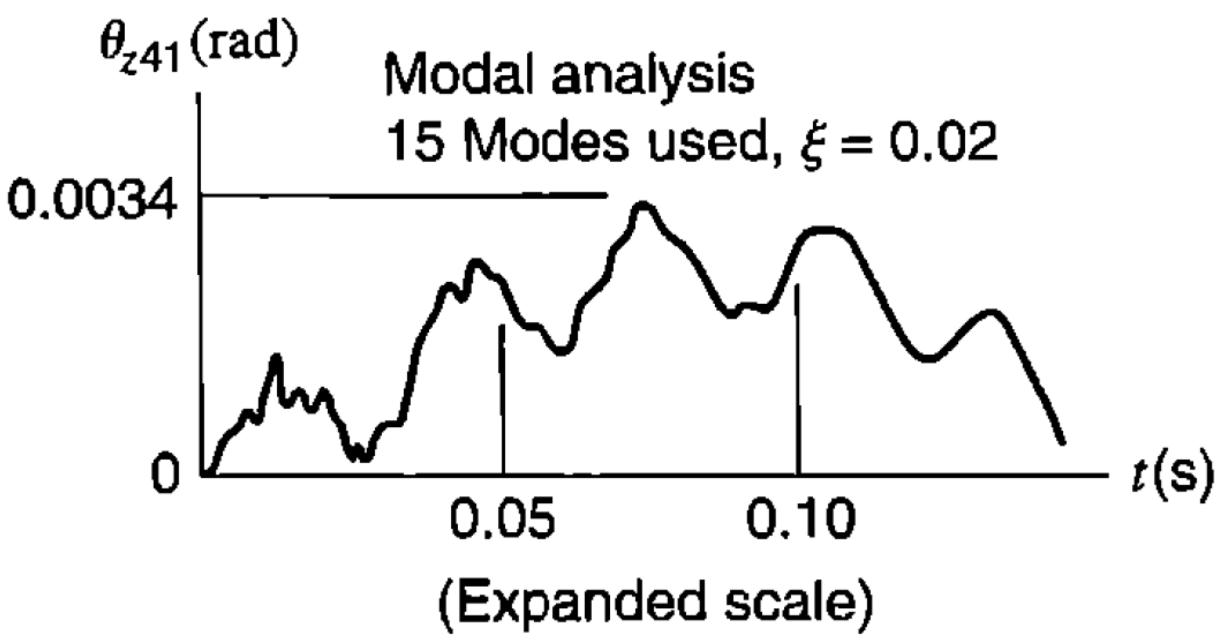


Figure 8: Text book figure, ξ =0.02

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

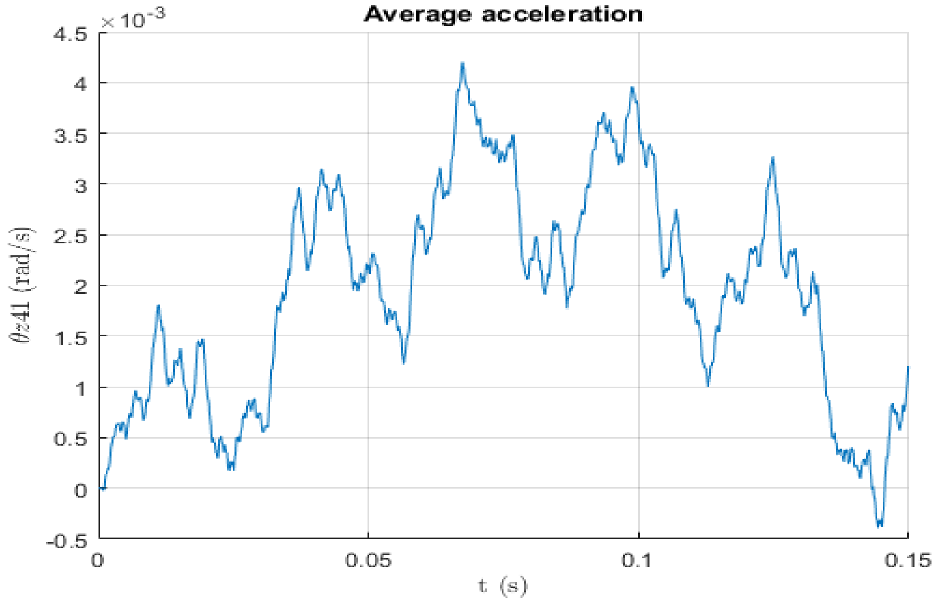


Figure 9: Average acceleration, ξ = 0

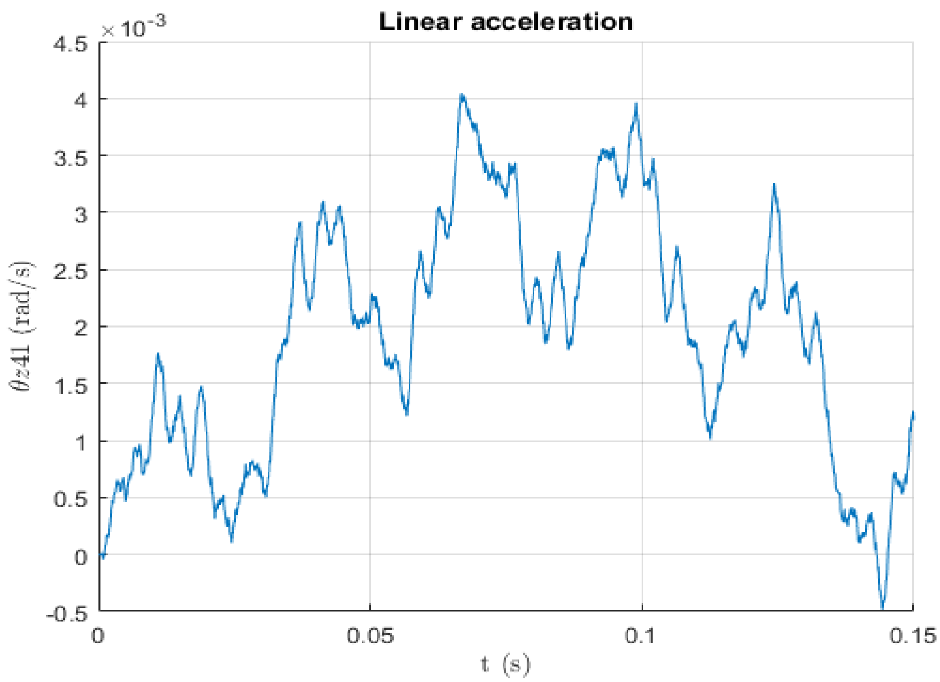


Figure 10: Linear acceleration, ξ =0

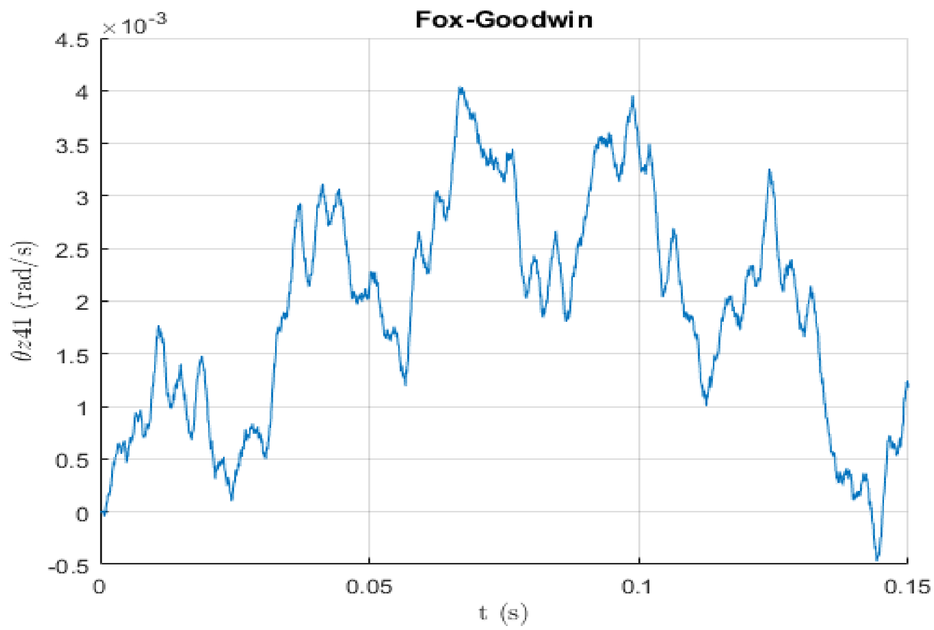


Figure 11: Fox-Goodwin, ξ = 0

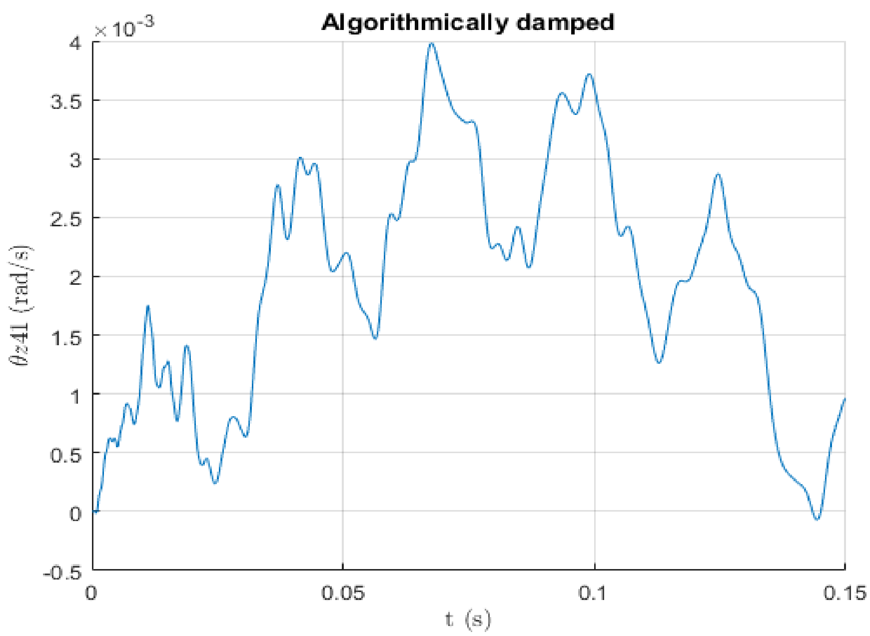


Figure 12: Algorithmically damped, ξ = 0

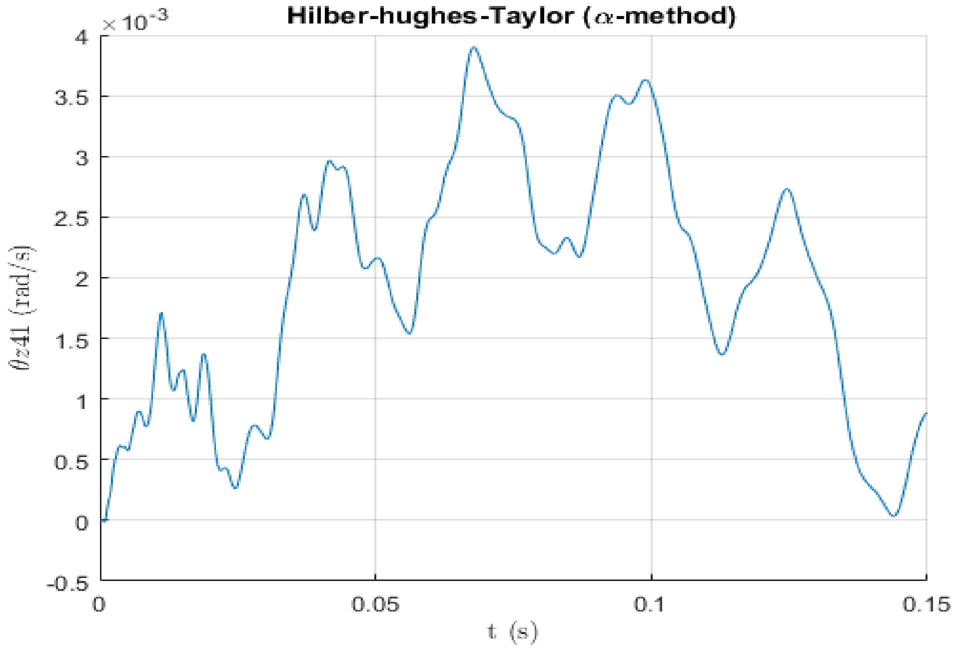


Figure 13: Hilber-hughes-Taylor (α-method), ξ = 0

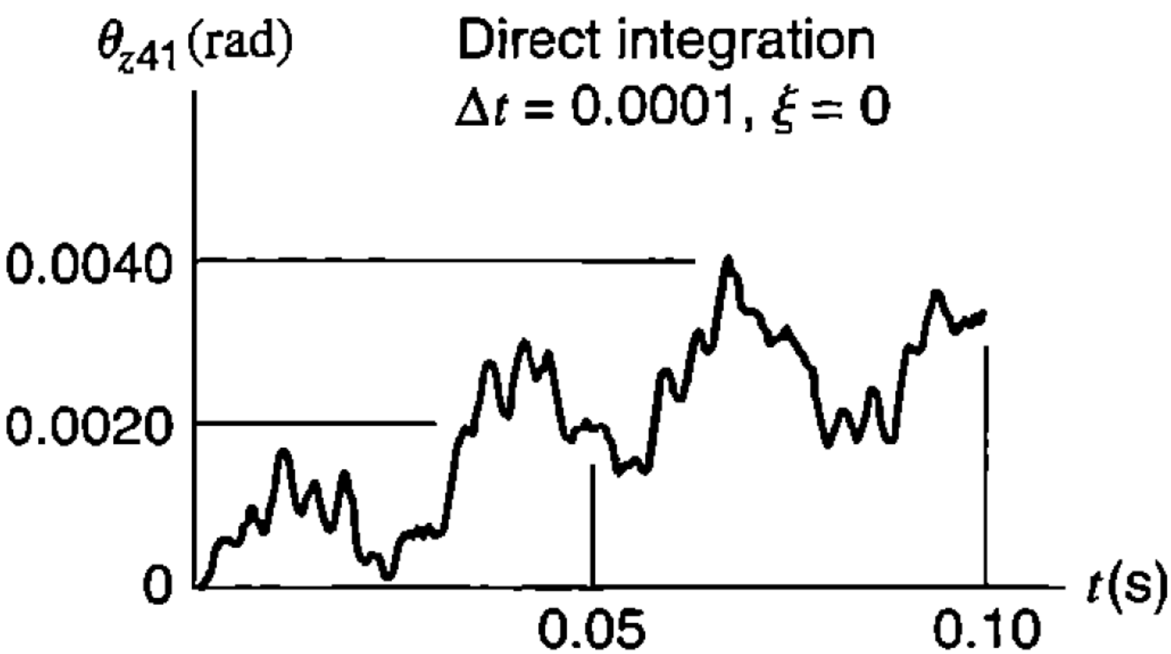


Figure 14: Text book figure, ξ = 0

TABLE 11.18-1. ROTATION θ_z AT NODE 41 IN FIG. 11.18-1: MAXIMUM MAGNITUDES OF θ_{z41} (RADIANS) AND ITS TIME DERIVATIVES $d\theta_{z41}/dt$ AND $d^2\theta_{z41}/dt^2$, IN TIME RANGES INDICATED, COMPUTED BY MODAL ANALYSIS AND BY IMPLICIT DIRECT INTEGRATION.

		Modal analysis		Average acceleration method		
Quantity	Range of t	2 modes	15 modes	$\Delta t = 10^{-3}$ s	$\Delta t = 10^{-4}$ s	$\Delta t = 2(10^{-5})$ s
θ_{z41}	0 to 0.10 s	0.00337	0.00347	0.00386	0.00413	0.00410
$\dot{\theta}_{z41}$	0 to 0.02 s	0.15	0.60	0.24	0.84	1.44
$\ddot{\theta}_{z41}$	0 to 0.02 s	40	1524	171	3402	21,790

Table 3: From WFEM

Qty.	**Range of t**	**Avg. acc. $\Delta t=10^{-4}$**	**Linear acc. $\Delta t=10^{-4}$**	**Fox Goodwin $\Delta t=10^{-4}$**	**Algorithmically damped $\Delta t=10^{-4}$**	**A method $\Delta t=10^{-4}$**
θ_{z41}	0 to 0.1s	0.00355	0.003496	0.003498	0.00352	0.003494
θ_{z41}	0 to 0.02s	0.68	0.7317	0.7321	0.4692	0.4354
θ_{z41}	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 $\ddot{D}_0 \neq \{0\}$ but rather = $([\mathrm{M}]^{-1}(\{\mathrm{R}^{\mathrm{ext}}\}_0$ – [K] $\{D_0\}$ – [C]{ $\dot{D}_0$})) 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 matriecces 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')
```