```matlab
%% Initialize the system
% UNITS:  kips, inches
addpath('.\functions') % Loads in all of the functions
clear; clc;

% Define Structure Parameters as given
mg = 2000; % kips; weight of the sytem
T0 = 0.2;  % seconds; natural period of the system
Ry0 = 0.15*mg; % kip; yeild force of the system
E0 = 30000; % ksi; Elastic modulus
g = 386; % in/s^2; gravitational constant
L = 60; % in; arbitrary length

[K0, A, s_y0] = eqivalent_truss(mg, T0, Ry0, L, E0, g); % Finds the equivalent parameter of a equivalent truss for the SDOF
% MatData = Initialize_MatData(E0, s_y0, b, R0, cR1, cR2, a1, a2, L, A)
MatData = Initialize_MatData(E0, s_y0, 0.02, 5, 3, 0.15, 0, 0, L, A); % Initialize the material data
% Initialize_Material_State(sig, Et, epmin, epmax,epex, ep0, s0, epr, sr, kon, [initial_strains])
MatState = Initialize_Material_State(0, MatData.E, MatData.fy, -MatData.ey, MatData.ey, 0, MatData.ey, 0, 0, 0, [0 ,0, 0 ,0]); % Initialize the material State

%% Nonlinear SDOF System Subjected to Quasi-Static Cyclic Loading
P = load('P.txt');% Load Forces
record_static1 = Static(P, MatData,MatState,"Newton", 10); % Run static analysis with Newton Method
record_static2 = Static(P, MatData,MatState,"ModifiedNewton", 200); % Run static analysis with modified Newton Method
record_static3 = Static(P, MatData,MatState,"ModifiedNewton -initial", 800); % Run static analysis with modified Newton Method

%% Nonlinear SDOF System Subjected to Dynamic Loading
% Newmark Beta
time_step_method.alpha = 0.5;
time_step_method.beta = 0.25;
time_step_method.time_step = 0.02; % sec
% Define Structural Properties for Dynamic Analysis
MatData.mass = mg/g; % mass of the system
xi = 0.02; % Damping coefficient
MatData.damping =  2*xi*sqrt(K0*mg/g); % damping of the system
% Accelerations
[~, acc] = readvars(".\SYL360.txt");  % Loads in time and acceleration
acc= acc(2:end)*g; % Removes the first zero
record_Trans1 = Transient(acc, MatData, MatState, time_step_method,"Newton" , 10);
record_Trans2 = Transient(acc, MatData, MatState, time_step_method,"ModifiedNewton" , 10);
record_Trans4 = Transient(2*acc, MatData, MatState, time_step_method,"Newton" , 10); % Scaling the acceleration by 2

%% Nonlinear SDOF System Subjected to Dynamic Loading
% Decreasing the Time Step
time_step_method.time_step = 0.01; % sec
acc = interp(acc,2); % Linearly interpolate the data
record_Trans3 = Transient(acc, MatData, MatState, time_step_method,"Newton" , 10);

%% Nonlinear SDOF System Subjected to Dynamic Loading
% Linear System
time_step_method.time_step = 0.02; % sec
[~, acc] = readvars(".\SYL360.txt");  % Loads in time and acceleration
acc= acc(2:end)*g; % Resets the accelerations
MatData.fy = 10000*MatData.fy; % Sets the yield stress such that it remains elastic
record_Trans5 = Transient(acc, MatData, MatState, time_step_method,"Newton" , 10); % Linear Elastic System
record_Trans6 = Transient(2*acc, MatData, MatState, time_step_method,"Newton" , 10); % Linear Elastic System
```