```
%% Plot Everything
% Short Hand Notations
close all; clc;

force = "Applied Force (kip)";
U = "Relative Displacement (in)";
iter = "Number of Iterations";
LS = "Load Steps (n)";
T = "Time (sec)";
A = "Acceleration (in/sec^2)";

analy1 = "Quasi-Static Cyclic Loading Pushover Analysis";
analy2 = "Time History Analysis";

algo1 = "Netown-Raphson Algorithm";
algo2 = "Modified Netown-Raphson Algorithm";
algo3 = "Modified Netown-Raphson With Intial E_0 Algorithm";

R = "Internal Resisting Force";
UnBaFo = "Unbalanced Force";
MP = "Megenetto-Pinto Uniaxial Model";
plot_size = [1200, 0, 1300,700];

%% Figure 1 - Force Pushover Curve
close all;
P = load('P.txt');
name = "Force Used in Pushover Analysis";
plot(P);
title(name); xlabel(LS); ylabel(force); grid on;
xticks(1:23); xlim([1,23]);
set(gcf, 'Position',  [1200, 0, 1300,500])
print_file(1, name)
%% Figure 2 -Quasi Static Cyclic Loading Using the Newton Algorithm
close;
name = [analy1, "using "+algo1];
fig = 2;
grid on; hold on;
plot_MenegottoPinto(record_static1);
plot_iterations(record_static1);
plot_RU_curve(record_static1, name);
set(gcf, 'Position',  plot_size)
set(gca,"FontSize",18)
print_file(fig, name)
%% Figure 3 -Quasi Static Cyclic Loading Using the Modified Newton Algorithm
close;
fig = 3;
name = [analy1, "using "+algo2];
grid on; hold on;
plot_MenegottoPinto(record_static2);
plot_iterations(record_static2);
plot_RU_curve(record_static2, name);
set(gcf, 'Position',  plot_size)
print_file(fig, name)
%% Figure 4 -Quasi Static Cyclic Loading Using the Modified Newton Algorithm -Inital
close;
name = [analy1, "using "+algo3];
fig = 4;
```

```
grid on; hold on;
plot_MenegottoPinto(record_static3);
plot_iterations(record_static3);
plot_RU_curve(record_static3, name);
set(gcf, 'Position',  plot_size)
print_file(fig, name)
%% Figure 5 -Unbalanecd Force Per Iterations
close all;
fig = 5;
subplot(1,2,1); hold on;
name = [UnBaFo+" Per Iteration","Using " + algo1];
for i = 1:size(record_static1.Unb,1)
    Unb = record_static1.Unb(i,:);
    Unb = Unb(Unb ~=0);
    plot(i*ones(numel(Unb)), Unb,'r-x');
end
title(name); set(gca, 'YScale', 'log','xminorgrid','on','yminorgrid','on'); ylabel("Unbalanced Force (kip)"); xlabel("Load Step (n)"); xlim([0,22]);
subplot(1,2,2); hold on;
name = [UnBaFo+" Per Iteration","Using " + algo2];
for i = 1:size(record_static2.Unb,1)
    Unb = record_static2.Unb(i,:);
    Unb = Unb(Unb ~=0);
    if length(Unb) >= 100; step = 50; else; step = 1; end;
    Unb = Unb(1:step:end);
    plot(i*ones(numel(Unb)), Unb,'r-x');
end
set(gca, 'YScale', 'log','xminorgrid','on','yminorgrid','on'); ylabel("Unbalanced Force (kip)"); xlabel("Load Step (n)");xlim([0,22]);
title(name);
set(gcf, 'Position',  plot_size)
%
print_file(fig, "Unbalanced Force Per Iteration")
%% Figure 6 - Number of Iterations Per Load Step for Static Analysis
close all;
fig = 6;
name = "Number of Iterations Per Load Step for Static Analysis";
title(name); hold on; grid on;
scatter(1:length(record_static3.iter),record_static3.iter,70,'+',"LineWidth",3,'DisplayName',"Modified Newton - intial");
scatter(1:length(record_static2.iter),record_static2.iter,70,'+',"LineWidth",3,'DisplayName',"Modified Newton");
scatter(1:length(record_static1.iter),record_static1.iter,70,'+',"LineWidth",3,'DisplayName',"Newton");
set(gca, 'YScale', 'log'); legend("location","best");
ylabel("Number of Iterations"); xlabel("Load Step"); xlim([0,22]); xticks(0:22)
set(gcf, 'Position',  [1200, 0, 1300,500])
print_file(fig, name)
%% Figure 7 - Milliseconds to Run Each Algorithm
n = 500;
[T1, T2, T3] = deal(zeros(1,n));
for i = 1:n
    tic
    ans = Static_no_record(P, MatData,MatState,"Newton", 10);
    T1(i) = toc;
end
%
for i = 1:n
    disp("Running Modified Newton. On Iteration " + i)
    tic
    ans = Static_no_record(P, MatData,MatState,"ModifiedNewton", 200);
    T2(i) = toc;
```

```
115 end
116 %
117 for i = 1:n
118     disp("Running Modified Newton - initial. On Iteration " + i)
119     tic
120     ans = Static_no_record(P, MatData,MatState,"ModifiedNewton -initial", 800);
121     T3(i) = toc;
122 end
123 close all; figure;
124 fig = 7;
125 name = "Milliseconds to run each Algorithm";
126
127 subplot(1,3,1);
128 histogram(T1*10^3,"BinWidth",0.01,"BinLimit",[1.21,1.25]);
129 xlabel("Milliseconds"); ylabel("Counts");title(algo1,'FontSize',12);
130 subplot(1,3,2);
131 histogram(T2*10^3,"BinWidth",0.01,"BinLimit",[13.5,13.8]);
132 xlabel("Milliseconds"); ylabel("Counts");title(algo2,'FontSize',12);
133 subplot(1,3,3);
134 histogram(T3*10^3,"BinWidth",0.1,"BinLimit",[56,57.5]);
135 xlabel("Milliseconds"); ylabel("Counts"); title(algo3,'FontSize',12);
136 set(gcf, 'Position',  [1200, 0, 1300,500])
137 print_file(fig, name)
138 %% Figure 8 -1994 Northright Earthquake from Sylmar Hospital Station
139 close;
140 name = "1994 Northright Earthquake from Sylmar Hospital Station";
141 fig = 8;
142 [time, acc] = readvars(".\SYL360.txt");  % Loads in time and acceleration
143 acc= acc(2:end); time = time(2:end);
144 plot(time,acc);
145 title(name); xlabel(T);ylabel(A);
146 set(gcf, 'Position',  [1200, 0, 1300,600]);
147 print_file(fig, name)
148 %% Figure 9 -Dyanmic Loading Using the Newton Algorithm
149 close;
150 name = ["Comparing " + analy2, "using "+algo1+" and "+algo2];
151 fig = 9;
152 title(name); grid on; hold on;
153 p = plot_RU_curve(record_Trans1, R+" "+ algo1); p.Color = 'r';
154 plot_RU_curve(record_Trans2, R+" "+ algo2);
155 set(gcf, 'Position',  plot_size)
156 print_file(fig, name)
157 %% Figure 10 -Dyanmic Loading Relative Displacement and Absolute Acceleration
158 close all; clc
159 name = ["Comparing Relative Displacement and Absolute Acceleration for the", strjoin([algo1,algo2]," and ")];
160 sgtitle((name),"FontSize",18);
161 fig = 10;
162 plot_disp_acc(record_Trans1,record_Trans2,[algo1, algo2]);
163 set(gcf, 'Position',  plot_size)
164 print_file(fig,strjoin(name))
165 %% Figure 11- Number of Iterations Per Load Step For Dynamic Anlaysis
166 close all;
167 fig = 11;
168 name = "Number of Iterations Per Load Step for Dyanmic Analysis";
169 title(name); hold on; grid on;
170 scatter(1:length(record_Trans2.iter),record_Trans2.iter,70,'+',"LineWidth",3,'DisplayName',"Modified Newton");
171 scatter(1:length(record_Trans1.iter),record_Trans1.iter,70,'+',"LineWidth",3,'DisplayName',"Newton");
```

```
172 legend("location","best");
173 ylabel("Number of Iterations"); xlabel("Load Step");
174 set(gcf, 'Position',  [1200, 0, 1300,500])
175 print_file(fig, name)
176 %% Figure 12 -Comparing Force History Hisory Using  Delta = 0.1
177 close all; clc
178 name = ["Comparing " + analy2, "with \Deltat=0.2s and \Deltat=0.1s "];
179 fig = 12;
180 title(name); hold on; grid on;
181 p = plot_RU_curve(record_Trans1, R+"; \Deltat = 0.2s"); set(p,'Color','r');
182 plot_RU_curve(record_Trans3, R+"; \Deltat = 0.1s");
183 set(gcf, 'Position',  plot_size)
184 % print_file(fig, "Comparing Changes in Time Step")
185 %% Figure 13 -Dyanmic Loading Relative Displacement and Absolute Acceleration
186 close all; clc; hold on;
187 name = ["Comparing Relative Displacement and Absolute Acceleration" ,"with \Deltat=0.2s and \Deltat=0.1s "];
188 t = sgtitle((name),"FontSize",18); t.FontWeight =' bold';
189 fig = 13;
190 plot_disp_acc(record_Trans1, record_Trans3,["\Deltat = 0.2s", "\Deltat = 0.1s"]);
191 set(gcf, 'Position',  [1200, 0, 1300,650])
192 print_file(fig,"Comparing Displacment Delta ")
193 %% Figure 14 -Comparing Force Hisory with Acceleration*2
194 name = ["Comparing " + analy2, "with Scaling Accleration by 2"];
195 close all; clc; grid on; hold on; title(name)
196 fig = 14;
197 plot_RU_curve(record_Trans4,  R+"; 2\cdotAcceleration");
198 p = plot_RU_curve(record_Trans1, R+"; 1\cdotAcceleration"); set(p,'Color','r');
199 set(gcf, 'Position',  [1200, 0, 1300,650])
200 % print_file(fig, name)
201 %% Figure 15 -Comparing Relative Displacement and Absolute Acceleration with Scaling Acceleration by 2
202 close all; clc; hold on;
203 name = ["Comparing Relative Displacement and Absolute Acceleration" ,"with Scaling Accleration by 2"];
204 t = sgtitle((name),"FontSize",18); t.FontWeight =' bold';
205 fig = 15;
206 plot_disp_acc(record_Trans4, record_Trans1,["Scaling by 2", "Scaling by 1"]);
207 set(gcf, 'Position',  plot_size)
208 print_file(fig,strjoin(name))
209 %% Figure 16- Ratio of Various Paramter as Acceleration is scaled- Nonlinear
210 close all; clc;
211 fig = 16;
212 name = "Comparing Scaling Accleration in Nonlinear and Linear Systems";
213 t = sgtitle((name),"FontSize",18); t.FontWeight =' bold';
214 maxU = max(abs(record_Trans1.U));
215 maxR = max(abs(record_Trans1.R));
216 maxA = max(abs(record_Trans1.A));
217 n = 10;
218 [ratio_U, ratio_R, ratio_G] = deal(zeros(1,n));
219 for i = 1:n
220     record = Transient(i*acc, MatData, MatState, time_step_method,"Newton" , 10);
221     ratio_U(i) = max(abs(record.U))/maxU;
222     ratio_R(i) = max(abs(record.R))/maxR;
223     ratio_G(i) = max(abs(record.A))/maxA;
224 end
225 subplot(1,2,1);hold on; grid minor;
226 title("Nonlinear System")
227 plot(ratio_R,"DisplayName","Ratio of Max Internal Force","LineWidth",2);
228 plot(ratio_U,"DisplayName","Ratio of Max Displacement","LineWidth",2);
```

```
229 plot(ratio_G,"DisplayName","Ratio of Max Rel. Acceleration","LineWidth",2);
230 plot(1:n,'-.',"DisplayName","1:1"); xticks(1:n);
231 xlabel("Acceleration Factor"); ylabel("Ratio of Scaled:Original Values"); legend("Location","Northwest","FontSize",12);
232
233
234 maxU = max(abs(record_Trans5.U));
235 maxR = max(abs(record_Trans5.R));
236 maxA = max(abs(record_Trans5.A));
237 [ratio_U, ratio_R, ratio_G] = deal(zeros(1,n));
238 for i = 1:n
239     record = Transient(i*acc, MatData, MatState, time_step_method,"Newton", 10);
240     ratio_U(i) = max(abs(record.U))/maxU;
241     ratio_R(i) = max(abs(record.R))/maxR;
242     ratio_G(i) = max(abs(record.A))/maxA;
243 end
244 subplot(1,2,2); hold on; grid minor;
245 title("Linear System")
246 plot(ratio_R,"DisplayName","Ratio of Max Internal Force","LineWidth",2);
247 plot(ratio_U,"DisplayName","Ratio of Max Displacement","LineWidth",2);
248 plot(ratio_G,"DisplayName","Ratio of Max Rel. Acceleration","LineWidth",2);
249 plot(1:n,'-.',"DisplayName","1:1");xticks(1:n);
250 xlabel("Acceleration Factor"); ylabel("Ratio of Scaled:Original Values"); legend("Location","Northwest","FontSize",12);
251
252 set(gcf, 'Position',  [1200, 0, 1300,550])
253 print_file(fig,name)
254 %% Figure 17 -Comparing Force History of Linear and Nonlinear System
255 close all; clc
256 name = ["Comparing " + analy2, "Of Nonlinear and Linear- Elastic System"];
257 fig = 17;
258 title(name); hold on; grid on;
259 yyaxis right; p = plot_RU_curve(record_Trans5, "Linear Elastic System"); set(p,'Color','r');
260 yyaxis left ;plot_RU_curve(record_Trans1, "Nonlinear System");
261 set(gcf, 'Position',  plot_size)
262 print_file(fig, name)
263 %% Figure 18 -Comparing Relative Displacement and Absolute Acceleration with Linear System
264 close all; clc; hold on;
265 name = ["Comparing Relative Displacement and Absolute Acceleration" ,"Between Nonlinear and Linear System"];
266 t = sgtitle((name),"FontSize",18); t.FontWeight =' bold';
267 fig = 18;
268 plot_disp_acc(record_Trans5, record_Trans1,["Linear System", "Nonlinear System"]);
269 set(gcf, 'Position',  plot_size)
270 print_file(fig,strjoin(name))
271 %% Figure 19 -Dyanmic Loading Using the Newton Algorithm Base Case
272 close all; clc;
273 name = analy2+ " using "+algo1;
274 fig = 19;
275 title(name); grid on; hold on;
276 p = plot_RU_curve(record_Trans1, R+" "+ algo1);
277 set(gcf, 'Position',  plot_size)
278 print_file(fig, name)
279 %% Figure 10 -Relative Displacement and Absolute Acceleration
280 close all; clc
281 name = "Relative Displacement and Absolute Acceleration for the " + algo1;
282 t = sgtitle((name),"FontSize",18); t.FontWeight =' bold';
283 fig = 20;
284 plot_disp_acc(record_Trans1,[],[]);
285 set(gcf, 'Position',  plot_size)
```

```
print_file(fig,strjoin(name))
%%
clc;
print_info(record_Trans1)
compare_records(record_Trans1, record_Trans4)
compare_records(record_Trans5, record_Trans6)

%%
function print_info(record)
disp(newline + "Printing Information On the Record")
maxU = max(record.U);
t_maxU = record.time(record.U ==max(record.U));
maxR = record.R(record.U ==max(record.U));
minU = min(record.U);
t_minU = record.time(record.U ==min(record.U));
minR = record.R(record.U ==min(record.U));
maxV = max(abs(record.V));
t_maxV = record.time(max(abs(record.V))  == abs(record.V));
maxG = max(abs(record.A+record.acc));
t_maxG = record.time(max(abs(record.A))  == abs(record.A));
str = [sprintf("The maximum positive displacement was %.3f inches at %.2f seconds.",maxU,t_maxU) ;...
       sprintf("At that moment, the structure was experiencing %.3f kip of internal resisting force.", maxR); ...
       sprintf("The maximum negative displacement was %.3f inches at %.2f seconds.",minU,t_minU);...
       sprintf("At that moment, the structure was experiencing %.3f kip of internal resisting force.", minR);...
       sprintf("The maximum absolute velocity of the structure was %.3f in/s at %.3f seconds.",maxV,t_maxV);...
       sprintf("The maximum absolute acceleration felt by the structure was %.3f in/sec^2 at %.3f seconds.",maxG,t_maxG);...
       sprintf("The average iteration for this record is %.2f.",mean(record.iter))];
   disp(strjoin(str));
disp(str);
end

%%
function compare_records(record1, record2)
disp(newline+ "Comparing Two Records")
maxU1 = max(abs(record1.U));
t_maxU1 = record1.time(abs(record1.U) ==max(abs(record1.U)));
maxU2 = max(abs(record2.U));
t_maxU2 = record2.time(abs(record2.U) ==max(abs(record2.U)));
maxR1 = record1.R(abs(record1.U) ==max(abs(record1.U)));
maxR2 = record2.R(abs(record2.U) ==max(abs(record2.U)));
maxG1 = max(abs(record1.A+record1.acc))/386;
maxG2 = max(abs(record2.A+record2.acc))/386;

str = [sprintf("Record 1 had a maximum displacement of %.2f inches at %.2f seconds.",maxU1,t_maxU1) ;...
       sprintf("Record 2 had a maximum displacement of %.2f inches at %.2f seconds.",maxU2,t_maxU2) ;...
       sprintf("The ratio between records 2 to 1 is %.2f.",maxU2/maxU1);...

       sprintf("The structure in Record 1 resisted a maximum force of %.2f kips.",maxR1) ;...
       sprintf("The structure in Record 2 resisted a maximum force of %.2f kips.",maxR2) ;...
       sprintf("The ratio between records 2 to 1 is %.2f.",maxR2/maxR1);...

       sprintf("Record 1 had a maximum acceleration of %.2fg",maxG1) ;...
       sprintf("Record 2 had a maximum acceleration of %.2fg.",maxG2) ;...
       sprintf("The ratio between records 2 to 1 is %.2f.",maxG2/maxG1)];
disp(strjoin(str));
disp(str);
end
```

```
343
344 %% Functions
345 function plot_disp_acc(record1, record2, name)
346     if ~isempty(record2)
347     subplot(2,1,1); hold on;
348     plot(record1.time,record1.U,'Color',[1,0,0,1],"DisplayName",name(1),"LineWidth",1);
349     plot(record2.time,record2.U,'Color',[0,0,1,1],"DisplayName",name(2));
350     xlabel("Time (sec)"); ylabel("Displacement (in)"); title("Relative Displacement");grid on; legend("Location","best");
351
352     subplot(2,1,2); hold on;
353     plot(record1.time,record1.A+record1.acc,'Color',[1,0,0,1],"DisplayName",name(1),"LineWidth",1);
354     plot(record2.time,record2.A+record2.acc,'Color',[0,0,1,1],"DisplayName",name(2));
355     xlabel("Time (sec)"); ylabel("Acceleration(in/sec^2)");title("Absolute Accleration");grid on; legend("Location","best");
356     else
357     subplot(2,1,1);
358     plot(record1.time,record1.U,'b');
359     xlabel("Time (sec)"); ylabel("Displacement (in)"); title("Relative Displacement");grid on;
360     subplot(2,1,2);
361     plot(record1.time,record1.A+record1.acc,'b');
362     xlabel("Time (sec)"); ylabel("Acceleration(in/sec^2)");title("Absolute Accleration");grid on;
363     end
364     % subplot(3,1,2); plot(time,record1.V,'b',"DisplayName","Velocity"); xlabel("Time (sec)"); ylabel("Velocity (in/sec)");
365 end
366 function plot_iterations(record)
367     plot(record.U_iter,record.P_iter,"DisplayName","Unbalanced Force Path");
368 end
369 function plot_MenegottoPinto(record)
370     [MP_U, MP_Force] = Menegotto_Pinto(record.U,record.MatData, record.MatState);
371     plt = plot(MP_U , MP_Force,'r','LineWidth',4, "DisplayName"," Menegotto-Pinto Uniaxial Model" );
372     plt.Color(4) = 0.3;
373 end
374 function pl = plot_RU_curve(record, name)
375     pl = plot(record.U,record.R,'b',"DisplayName",name,'MarkerEdgeColor','r',"MarkerSize",4);
376     xlabel("Relative Displacement (in)"); ylabel("Internal Resisting Force (kip)"); legend("Location","Southeast");
377 end
378 function print_file(no, name)
379     print("figures\"+string(no)+" "+ strjoin(name),'-dsvg','-PMicrosoft Print to PDF','-r600','-painters');
380 end
```