```
function [ fig, undefPlot, defPlot ] = plot_modeShape(modelDefFilePath, modeShapeDirPath, mode, scale, nDim, crdTransfMatrix, undefColor, undefLineWidth, defColor, defLineWidth, figNum)
%% DESCRIPTION

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INPUT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% modelDefFilePath        : full path to file containing model definition
%                           (see note)
% modeShapeDirPath        : full path to directory containing
%                           modeShape_$mode.txt file (see note)
% mode                    : mode number to plot
% scale                   : scale factor for mode shape
% nDim                    : number of dimensions (2 or 3)
% crdTransfMatrix         : coordinate transformation matrix (see note)
% undefColor              : color for undeformed shape
% undefLineWidth          : line width for undeformed shape
% defColor                : color for deformed shape
% defLineWidth            : line width for deformed shape
% figNum                  : Matlab figure number
%-------------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note about modelDefFilePath
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% For using this plotter, generate a text output of your model as you
% write the .tcl input file. This text output should at least have all
% the nodal and element information. As you go on adding nodes and
% elements in the .tcl input file, it is required to write the command
% lines for adding nodes and elements to this text file. Provide the
% full path of this text file.
%-------------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note about modeShape_$mode.txt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Save modeShape_$mode.txt for mode number = $mode. This file should have
% 1+6 columns for nDim = 3.
% 1+3 columns for nDim = 2.
% Column 1 should have tags of all nodes in the model
% Columns 2:end should have the node eigenvectors of all nodes in the model
% for mode number = $mode.
%-------------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note about crdTransfMatrix (= R):
% m: MATLAB
% o: OpenSees
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [oX;oY;oZ] = R*[mX;mY;mZ]
% [mX;mY;mZ] = R'*[oX;oY;oZ]
% Assume (mX, mY, mZ) as basis
% 1st row of R = oX in (mX, mY, mZ)
% 2nd row of R = oY in (mX, mY, mZ)
% 3rd row of R = oZ in (mX, mY, mZ)
%-------------------------------------------------------------------------
%% READ MODEL DATA
```

```
57
58 [modelDefFilePath, modeShapeDirPath] = convertStringsToChars(modelDefFilePath, modeShapeDirPath);
59
60 if ~ismember(nDim,[2,3])
61     error('Incorrect dimension! Should be 2 or 3.')
62 end
63
64 modelDataFile_fid = fopen(fullfile(modelDefFilePath,'modelData.txt'),'r');
65 str = textscan(modelDataFile_fid,'%s');
66 nodeCount = sum(ismember(str{:},'node'));
67 eleCount = sum(ismember(str{:},'element')) + sum(ismember(str{:},'rigidLink'));
68 nodeData = zeros(nodeCount, nDim + 1);
69 eleData = zeros(eleCount, 3);
70 eleTypes = cell(eleCount, 1);
71 nodeCtr = 0;
72 eleCtr = 0;
73 frewind(modelDataFile_fid);
74
75 while ~feof(modelDataFile_fid)
76     currLine = fgetl(modelDataFile_fid);
77     currLine = strtrim(strtok(currLine,';'));
78     currLine = strsplit(currLine);
79
80     if size(currLine,2) > 1 && strcmp(currLine{1},'node') == 1
81         nodeCtr = nodeCtr + 1;
82         nodeNum = str2double(currLine{2});
83         coordinates = arrayfun(@(x) str2double(currLine{x}), 3:length(currLine), 'UniformOutput', 1);
84         nodeData(nodeCtr,:) = [nodeNum coordinates];
85     end
86
87     if (size(currLine,2) > 1 && strcmp(currLine{1},'element') == 1) || (size(currLine,2) > 1 && strcmp(currLine{1},'rigidLink') == 1)
88         eleCtr = eleCtr + 1;
89         if strcmp(currLine{1},'rigidLink') == 1
90             eleNum = 0;
91             connectingNodes = [str2double(currLine{4}),str2double(currLine{3})];
92             eleType = currLine{1};
93         else
94             eleNum = str2double(currLine{3});
95             connectingNodes = [str2double(currLine{4}),str2double(currLine{5})];
96             eleType = currLine{2};
97         end
98         eleData(eleCtr,:) = [eleNum connectingNodes];
99         eleTypes{eleCtr} = eleType;
100    end
101 end
102 fclose(modelDataFile_fid);
103
104 nodeData(:,2:size(nodeData,2)) = (crdTransfMatrix'*nodeData(:,2:size(nodeData,2))')';
105
106 %% PLOT UNDEFORMED SHAPE
107 fig = figure(figNum);
108 axis equal
109 hold on
110 grid on
111 box on
112 [~,element_iNode] = ismember(eleData(:,2),nodeData(:,1));
113 [~,element_jNode] = ismember(eleData(:,3),nodeData(:,1));
```

```
114
115 if nDim == 3
116     for i = 1:size(eleData,1)
117         undefPlot = ...
118             plot3([nodeData(element_iNode(i),2) nodeData(element_jNode(i),2)],...
119             [nodeData(element_iNode(i),3) nodeData(element_jNode(i),3)],...
120             [nodeData(element_iNode(i),4) nodeData(element_jNode(i),4)],...
121             '--','Color',undefColor,'LineWidth',undefLineWidth);hold on
122     end
123 elseif nDim == 2
124     for i = 1:size(eleData,1)
125         undefPlot = ...
126             plot([nodeData(element_iNode(i),2) nodeData(element_jNode(i),2)],...
127             [nodeData(element_iNode(i),3) nodeData(element_jNode(i),3)],...
128             '--','Color',undefColor,'LineWidth',undefLineWidth);hold on
129     end
130 end
131
132 %% PLOT MODE SHAPE
133 %% READ MODE SHAPE INFORMATION
134 nodeDataDeformed = zeros(size(nodeData));
135 nodeEigenVector = load(fullfile(modeShapeDirPath,['modeShape_' num2str(mode) '.txt']));
136
137 for i = 1:size(nodeData,1)
138     [~, ind] = ismember(nodeData(i,1), nodeEigenVector(:,1));
139     if nDim == 2
140         nodeDataDeformed(i,:) = [nodeData(i,1) nodeEigenVector(ind,2:size(nodeEigenVector,2)-1)];
141     elseif nDim == 3
142         nodeDataDeformed(i,:) = [nodeData(i,1) nodeEigenVector(ind,2:size(nodeEigenVector,2)-3)];
143     end
144 end
145
146 nodeDataDeformed(:,2:size(nodeDataDeformed,2)) = (crdTransfMatrix'*nodeDataDeformed(:,2:size(nodeDataDeformed,2))')';
147 nodeDataDeformed(:,2:size(nodeDataDeformed,2)) = scale*nodeDataDeformed(:,2:size(nodeDataDeformed,2)) + nodeData(:,2:size↙
(nodeData,2));
148 %% PLOT MODE SHAPE
149 figure(figNum)
150 if nDim == 3
151     for i = 1:size(eleData,1)
152         defPlot = ...
153             plot3([nodeDataDeformed(element_iNode(i),2) nodeDataDeformed(element_jNode(i),2)],...
154             [nodeDataDeformed(element_iNode(i),3) nodeDataDeformed(element_jNode(i),3)],...
155             [nodeDataDeformed(element_iNode(i),4) nodeDataDeformed(element_jNode(i),4)],...
156             'LineStyle','-','Color',defColor,'LineWidth',defLineWidth);
157     end
158     plot3(nodeDataDeformed(:,2),nodeDataDeformed(:,3),nodeDataDeformed(:,4),'ks','LineWidth',1,'MarkerFaceColor',[0.5 0.5 0.5])
159 elseif nDim == 2
160     for i = 1:size(eleData,1)
161         defPlot = ...
162             plot([nodeDataDeformed(element_iNode(i),2) nodeDataDeformed(element_jNode(i),2)],...
163             [nodeDataDeformed(element_iNode(i),3) nodeDataDeformed(element_jNode(i),3)],...
164             'LineStyle','-','Color',defColor,'LineWidth',defLineWidth);
165     end
166     plot(nodeDataDeformed(:,2),nodeDataDeformed(:,3),'ks','LineWidth',1,'MarkerFaceColor',[0.5 0.5 0.5])
167 end
168
169 end
```