```
function [ ] = plot_fiberSection(secDefFilePath, secTag, figNum, fibColor)
% SE 201B
% Fiber section plotter
% Angshuman Deb
%% INPUT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% secDefFilePath      : full path to file containing section definiton(s)
%                     (see note)
% secTag              : secTag of section to plot
% figNum              : Matlab figure number
% fibColor (optional) : Matrix of matTags along first column and
%                       r, g, b, alpha(optional) values along
%                       2nd, 3rd, 4th, 5th(optional) columns
%----------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note about secDefFilePath
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% For using this plotter, generate a text file with the
% full section definition as you would write in a tcl input file.
% Provide the full path of this text file to this plotter.
%----------------------------------------------------------------------
%% READ SECTION DATA
secDefFilePath = convertStringsToChars(secDefFilePath);
fid = fopen(fullfile(secDefFilePath), 'r');
RectPatch = [];
QuadPatch = [];
CircPatch = [];
Fiber = [];
StraightLayer = [];
CircLayer = [];

% RectPatch
% $matTag $numSubdivY $numSubdivZ $yI $zI $yJ $zJ

% QuadPatch
% $matTag $numSubdivIJ $numSubdivJK $yI $zI $yJ $zJ $yK $zK $yL $zL

% CircPatch
% $matTag $numSubdivCirc $numSubdivRad $yCenter $zCenter $intRad $extRad $startAng $endAng

% Fiber
% $matTag $yLoc $zLoc $A

% StraightLayer
% $matTag $numFiber $areaFiber $yStart $zStart $yEnd $zEnd

% CircLayer
% $matTag $numFiber $areaFiber $yCenter $zCenter $radius <$startAng $endAng>

while feof(fid) ~= 1
    currLine = fgetl(fid);
    currLine = strsplit(strtrim(strtok(currLine,';')));
    currLine = currLine(~cellfun('isempty',currLine));
    if ~isempty(currLine)
        if strcmp(currLine{1}, 'section') && strcmp(currLine{2},'Fiber')
            secTagCurr = str2double(currLine{3});
            if secTag == secTagCurr
                secDefEnd = false;
                while ~secDefEnd
                    currSecLine = fgetl(fid);
                    currSecLine = strsplit(strtrim(strtok(currSecLine,';')));
                    currSecLine = currSecLine(~cellfun('isempty',currSecLine));
                    if ~isempty(currSecLine)
                        if strcmp(currSecLine{1},'patch')
                            if strcmp(currSecLine{2},'rect')
                                RectPatch = [RectPatch;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9})];
                            elseif strcmp(currSecLine{2},'quad')
                                QuadPatch = [QuadPatch;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9}) str2double(currSecLine{10}) str2double(currSecLine{11}) str2double(currSecLine{12}) str2double(currSecLine{13})];
                            elseif strcmp(currSecLine{2},'circ')
                                CircPatch = [CircPatch;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9}) str2double(currSecLine{10}) str2double(currSecLine{11})];
                            end
                        elseif strcmp(currSecLine{1},'fiber')
                            Fiber = [Fiber;str2double(currSecLine{5}) str2double(currSecLine{2}) str2double(currSecLine{3}) str2double(currSecLine{4})];
                        elseif strcmp(currSecLine{1},'layer')
                            if strcmp(currSecLine{2},'straight')
                                StraightLayer = [StraightLayer;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9})];
                            elseif strcmp(currSecLine{2},'circ')
                                CircLayer = [CircLayer;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9}) str2double(currSecLine{10})];
                            end
                        elseif strcmp(currSecLine{1},'}')
                            secDefEnd = true;
                        end
                    end
                end
            end
        end
    end
end
fclose(fid);

if ~exist('fibColor', 'var')
    fibColor = NaN;
end
%% View Section
figure(figNum);hold on

for i1 = 1:size(RectPatch,1)
    matTag = RectPatch(i1,1);
    numSubdivY = RectPatch(i1,2);
    numSubdivZ = RectPatch(i1,3);
    yI = RectPatch(i1,4);
    zI = RectPatch(i1,5);
    yJ = RectPatch(i1,6);
    zJ = RectPatch(i1,7);

    yVec = linspace(yI,yJ,numSubdivY+1);
    zVec = linspace(zI,zJ,numSubdivZ+1);

    for i2 = 1:length(yVec)-1
        for i3 = 1:length(zVec)-1
            xPatch = [0,0,0,0];
            yPatch = [yVec(i2) yVec(i2+1) yVec(i2+1) yVec(i2)];
            zPatch = [zVec(i3) zVec(i3) zVec(i3+1) zVec(i3+1)];
            if ismember(matTag, fibColor(:,1))
                if size(fibColor,2) == 5
                    p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)-1));
                    p.FaceAlpha = fibColor(fibColor(:,1) == matTag,5);
                else
                    p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)));
                    p.FaceAlpha = 1;
                end
            else
                p = patch(xPatch,yPatch,zPatch,'k');
                p.FaceAlpha = 0.2;
            end
        end
    end

end

for i1 = 1:size(QuadPatch,1)
    matTag = QuadPatch(i1,1);
    numSubdivIJ = QuadPatch(i1,2);
    numSubdivJK = QuadPatch(i1,3);

    yI = QuadPatch(i1,4);
    zI = QuadPatch(i1,5);

    yJ = QuadPatch(i1,6);
    zJ = QuadPatch(i1,7);

    yK = QuadPatch(i1,8);
    zK = QuadPatch(i1,9);

    yL = QuadPatch(i1,10);
    zL = QuadPatch(i1,11);

    yIJ_vec = linspace(yI,yJ,numSubdivIJ+1);
    zIJ_vec = linspace(zI,zJ,numSubdivIJ+1);

    yJK_vec = linspace(yJ,yK,numSubdivJK+1);
    zJK_vec = linspace(zJ,zK,numSubdivJK+1);

    yLK_vec = linspace(yL,yK,numSubdivIJ+1);
    zLK_vec = linspace(zL,zK,numSubdivIJ+1);

    yIL_vec = linspace(yI,yL,numSubdivJK+1);
    zIL_vec = linspace(zI,zL,numSubdivJK+1);

    numLines_IJ = numSubdivIJ + 1;
    numLines_JK = numSubdivJK + 1;

    for i2 = 1:(numLines_IJ-1)
```

```
        for i3 = 1:(numLines_JK-1)

            if i2 == 1 && i3 == 1
                y1 = yIL_vec(i3);
                z1 = zIL_vec(i3);

                y2 = yIJ_vec(i2+1);
                z2 = zIJ_vec(i2+1);

                [y3,z3] = polyxpoly(...
                    [yIJ_vec(i2+1),yLK_vec(i2+1)],...
                    [zIJ_vec(i2+1),zLK_vec(i2+1)],...
                    [yIL_vec(i3+1),yJK_vec(i3+1)],...
                    [zIL_vec(i3+1),zJK_vec(i3+1)]);

                y4 = yIL_vec(i3+1);
                z4 = zIL_vec(i3+1);
            elseif i2 == 1 && i3 == (numLines_JK-1)
                y1 = yIL_vec(i3);
                z1 = zIL_vec(i3);

                [y2,z2] = polyxpoly(...
                    [yIJ_vec(i2+1),yLK_vec(i2+1)],...
                    [zIJ_vec(i2+1),zLK_vec(i2+1)],...
                    [yIL_vec(i3),yJK_vec(i3)],...
                    [zIL_vec(i3),zJK_vec(i3)]);

                y3 = yLK_vec(i2+1);
                z3 = zLK_vec(i2+1);

                y4 = yIL_vec(i3+1);
                z4 = zIL_vec(i3+1);
            elseif i2 == (numLines_IJ-1) && i3 == 1
                y1 = yIJ_vec(i2);
                z1 = zIJ_vec(i2);

                y2 = yIJ_vec(i2+1);
                z2 = zIJ_vec(i2+1);

                y3 = yJK_vec(i3+1);
                z3 = zJK_vec(i3+1);

                [y4,z4] = polyxpoly(...
                    [yIJ_vec(i2),yLK_vec(i2)],...
                    [zIJ_vec(i2),zLK_vec(i2)],...
                    [yIL_vec(i3+1),yJK_vec(i3+1)],...
                    [zIL_vec(i3+1),zJK_vec(i3+1)]);
            elseif i2 == (numLines_IJ-1) && i3 == (numLines_JK - 1)
                [y1,z1] = polyxpoly(...
                    [yIJ_vec(i2),yLK_vec(i2)],...
                    [zIJ_vec(i2),zLK_vec(i2)],...
                    [yIL_vec(i3),yJK_vec(i3)],...
                    [zIL_vec(i3),zJK_vec(i3)]);

                y2 = yJK_vec(i3);
                z2 = zJK_vec(i3);

                y3 = yJK_vec(i3+1);
                z3 = zJK_vec(i3+1);

                y4 = yLK_vec(i2);
                z4 = zLK_vec(i2);
            elseif i2 == 1 && ( 1 < i3 < (numLines_JK - 1))
                y1 = yIL_vec(i3);
                z1 = zIL_vec(i3);

                [y2,z2] = polyxpoly(...
                    [yIJ_vec(i2+1),yLK_vec(i2+1)],...
                    [zIJ_vec(i2+1),zLK_vec(i2+1)],...
                    [yIL_vec(i3),yJK_vec(i3)],...
                    [zIL_vec(i3),zJK_vec(i3)]);

                [y3,z3] = polyxpoly(...
                    [yIJ_vec(i2+1),yLK_vec(i2+1)],...
                    [zIJ_vec(i2+1),zLK_vec(i2+1)],...
                    [yIL_vec(i3+1),yJK_vec(i3+1)],...
                    [zIL_vec(i3+1),zJK_vec(i3+1)]);

                y4 = yIL_vec(i3+1);
                z4 = zIL_vec(i3+1);
            elseif i2 == (numLines_IJ-1) && ( 1 < i3 < (numLines_JK - 1))
                [y1,z1] = polyxpoly(...
                    [yIJ_vec(i2),yLK_vec(i2)],...
                    [zIJ_vec(i2),zLK_vec(i2)],...
                    [yIL_vec(i3),yJK_vec(i3)],...
                    [zIL_vec(i3),zJK_vec(i3)]);

                y2 = yJK_vec(i3);
                z2 = zJK_vec(i3);

                y3 = yJK_vec(i3+1);
                z3 = zJK_vec(i3+1);

                [y4,z4] = polyxpoly(...
                    [yIJ_vec(i2),yLK_vec(i2)],...
                    [zIJ_vec(i2),zLK_vec(i2)],...
                    [yIL_vec(i3+1),yJK_vec(i3+1)],...
                    [zIL_vec(i3+1),zJK_vec(i3+1)]);
            elseif (1 < i2 < (numLines_IJ-1)) && i3 == 1
                y1 = yIJ_vec(i2);
                z1 = zIJ_vec(i2);

                y2 = yIJ_vec(i2+1);
                z2 = zIJ_vec(i2+1);

                [y3,z3] = polyxpoly(...
                    [yIJ_vec(i2+1),yLK_vec(i2+1)],...
                    [zIJ_vec(i2+1),zLK_vec(i2+1)],...
                    [yIL_vec(i3+1),yJK_vec(i3+1)],...
                    [zIL_vec(i3+1),zJK_vec(i3+1)]);
                [y4,z4] = polyxpoly(...
                    [yIJ_vec(i2),yLK_vec(i2)],...
                    [zIJ_vec(i2),zLK_vec(i2)],...
                    [yIL_vec(i3+1),yJK_vec(i3+1)],...
                    [zIL_vec(i3+1),zJK_vec(i3+1)]);
            elseif (1 < i2 < (numLines_IJ-1)) && i3 == (numLines_JK-1)
                [y1,z1] = polyxpoly(...
                    [yIJ_vec(i2),yLK_vec(i2)],...
                    [zIJ_vec(i2),zLK_vec(i2)],...
                    [yIL_vec(i3),yJK_vec(i3)],...
                    [zIL_vec(i3),zJK_vec(i3)]);

                [y2,z2] = polyxpoly(...
                    [yIJ_vec(i2+1),yLK_vec(i2+1)],...
                    [zIJ_vec(i2+1),zLK_vec(i2+1)],...
                    [yIL_vec(i3),yJK_vec(i3)],...
                    [zIL_vec(i3),zJK_vec(i3)]);

                y3 = yLK_vec(i2+1);
                z3 = zLK_vec(i2+1);

                y4 = yLK_vec(i2);
                z4 = zLK_vec(i2);
            else
                [y1,z1] = polyxpoly(...
                    [yIJ_vec(i2),yLK_vec(i2)],...
                    [zIJ_vec(i2),zLK_vec(i2)],...
                    [yIL_vec(i3),yJK_vec(i3)],...
                    [zIL_vec(i3),zJK_vec(i3)]);
                [y2,z2] = polyxpoly(...
                    [yIJ_vec(i2+1),yLK_vec(i2+1)],...
                    [zIJ_vec(i2+1),zLK_vec(i2+1)],...
                    [yIL_vec(i3),yJK_vec(i3)],...
                    [zIL_vec(i3),zJK_vec(i3)]);
                [y3,z3] = polyxpoly(...
                    [yIJ_vec(i2+1),yLK_vec(i2+1)],...
                    [zIJ_vec(i2+1),zLK_vec(i2+1)],...
                    [yIL_vec(i3+1),yJK_vec(i3+1)],...
                    [zIL_vec(i3+1),zJK_vec(i3+1)]);
                [y4,z4] = polyxpoly(...
                    [yIJ_vec(i2),yLK_vec(i2)],...
                    [zIJ_vec(i2),zLK_vec(i2)],...
                    [yIL_vec(i3+1),yJK_vec(i3+1)],...
                    [zIL_vec(i3+1),zJK_vec(i3+1)]);
            end

            yPatch = [y1,y2,y3,y4];
            zPatch = [z1,z2,z3,z4];
            xPatch = zeros(size(yPatch));

            if ismember(matTag, fibColor(:,1))
                if size(fibColor,2) == 5
                    p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)-1));
                    p.FaceAlpha = fibColor(fibColor(:,1) == matTag,5);
                else
                    p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)));
                    p.FaceAlpha = 1;
                end
            else
                p = patch(xPatch,yPatch,zPatch,'k');
                p.FaceAlpha = 0.2;
            end
        end
    end
end

for i1 = 1:size(CircPatch,1)
    matTag = CircPatch(i1,1);
    numSubdivCirc = CircPatch(i1,2);
    numSubdivRad = CircPatch(i1,3);
    yCenter = CircPatch(i1,4);
    zCenter = CircPatch(i1,5);
    intRad = CircPatch(i1,6);
    extRad = CircPatch(i1,7);
    startAng = CircPatch(i1,8)*pi/180;
    endAng = CircPatch(i1,9)*pi/180;

    rVec = linspace(intRad,extRad,numSubdivRad+1);
    thetaVec = linspace(startAng,endAng,numSubdivCirc+1);

    for i2 = 1:length(rVec)-1
        for i3 = 1:length(thetaVec)-1
            xPatch = [0,0,0,0];
            yPatch = yCenter + [rVec(i2)*cos(thetaVec(i3)) rVec(i2+1)*cos(thetaVec(i3)) rVec(i2+1)*cos(thetaVec(i3+1)) rVec(i2)*cos(thetaVec(i3+1))];
            zPatch = zCenter + [rVec(i2)*sin(thetaVec(i3)) rVec(i2+1)*sin(thetaVec(i3)) rVec(i2+1)*sin(thetaVec(i3+1)) rVec(i2)*sin(thetaVec(i3+1))];
            if ismember(matTag, fibColor(:,1))
                if size(fibColor,2) == 5
                    p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)-1));
                    p.FaceAlpha = fibColor(fibColor(:,1) == matTag,5);
                else
                    p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)));
                    p.FaceAlpha = 1;
                end
            else
                p = patch(xPatch,yPatch,zPatch,'k');
                p.FaceAlpha = 0.2;
            end
        end
    end
end

for i1 = 1:size(CircLayer,1)
    matTag = CircLayer(i1,1);
    numFiber = CircLayer(i1,2);
    areaFiber = CircLayer(i1,3);
    yCenter = CircLayer(i1,4);
    zCenter = CircLayer(i1,5);
    radius = CircLayer(i1,6);

    if length(CircLayer(i1,:)) > 6
        startAng = CircLayer(i1,7)*pi/180;
        endAng = CircLayer(i1,8)*pi/180;
    else
        startAng = 0.0;
        endAng = (360 - 360/numFiber)*pi/180;
    end

    rVec = linspace(radius,radius,numFiber);
    thetaVec = linspace(startAng,endAng,numFiber);
    yVec = yCenter + rVec.*cos(thetaVec);
    zVec = zCenter + rVec.*sin(thetaVec);

    for i2 = 1:length(yVec)
        if ismember(matTag, fibColor(:,1))
            filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100, fibColor(fibColor(:,1) == matTag,2:size(fibColor,2)));
        else
            filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100,'k');
        end
    end
end

for i1 = 1:size(StraightLayer,1)
    matTag = StraightLayer(i1,1);
    numFiber = StraightLayer(i1,2);
    areaFiber = StraightLayer(i1,3);
    yStart = StraightLayer(i1,4);
    zStart = StraightLayer(i1,5);
    yEnd = StraightLayer(i1,6);
    zEnd = StraightLayer(i1,7);

    yVec = linspace(yStart,yEnd,numFiber);
    zVec = linspace(zStart,zEnd,numFiber);

    for i2 = 1:length(yVec)
        if ismember(matTag, fibColor(:,1))
            filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100, fibColor(fibColor(:,1) == matTag,2:size(fibColor,2)));
        else
            filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100,'k');
        end
    end
end

for i1 = 1:size(Fiber,1)
    matTag = Fiber(i1,1);
    yVec = Fiber(i1,2);
    zVec = Fiber(i1,3);
    areaFiber = Fiber(i1,4);
    if ismember(matTag, fibColor(:,1))
        filledCircle([0,yVec,zVec],sqrt(areaFiber/pi),100, fibColor(fibColor(:,1) == matTag,2:size(fibColor,2)));
    else
        filledCircle([0,yVec,zVec],sqrt(areaFiber/pi),100,'k');
    end
end

ylabel('y')
zlabel('z')
view([1,0,0])
box on;
axis square;
axis equal;

end

```

```
% Helper function to plot filled circle
function h = filledCircle(center,r,N,color)
if isnumeric(color) && length(color) == 4
    alpha = color(4);
    color = color(1:3);
else
    alpha = 1;
end
THETA=linspace(0,2*pi,N);
RHO=ones(1,N)*r;
[Y,Z] = pol2cart(THETA,RHO);
Y=Y+center(2);
Z=Z+center(3);
X = zeros(size(Y));
h=patch(X,Y,Z,color);
h.FaceAlpha = alpha;
end

```