

```

1 function [ ] = plot_fiberSection(secDefFilePath, secTag, figNum, fibColor)
2 % SE 201B
3 % Fiber section plotter
4 % Angshuman Deb
5 %% INPUT
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 % secDefFilePath : full path to file containing section definition(s)
8 % (see note)
9 % secTag : secTag of section to plot
10 % figNum : Matlab figure number
11 % fibColor (optional) : Matrix of matTags along first column and
12 % r, g, b, alpha(optional) values along
13 % 2nd, 3rd, 4th, 5th(optional) columns
14 %-----
15
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17 % Note about secDefFilePath
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19 % For using this plotter, generate a text file with the
20 % full section definition as you would write in a tcl input file.
21 % Provide the full path of this text file to this plotter.
22 %-----
23 %% READ SECTION DATA
24 secDefFilePath = convertStringsToChars(secDefFilePath);
25 fid = fopen(fullfile(secDefFilePath), 'r');
26 RectPatch = [];
27 QuadPatch = [];
28 CircPatch = [];
29 Fiber = [];
30 StraightLayer = [];
31 CircLayer = [];
32
33 % RectPatch
34 % $matTag $numSubdivY $numSubdivZ $yI $zI $yJ $zJ
35
36 % QuadPatch
37 % $matTag $numSubdivIJ $numSubdivJK $yI $zI $yJ $zJ $yK $zK $yL $zL
38
39 % CircPatch
40 % $matTag $numSubdivCirc $numSubdivRad $yCenter $zCenter $intRad $extRad $startAng $endAng
41
42 % Fiber
43 % $matTag $yLoc $zLoc $A
44
45 % StraightLayer
46 % $matTag $numFiber $areaFiber $yStart $zStart $yEnd $zEnd
47
48 % CircLayer
49 % $matTag $numFiber $areaFiber $yCenter $zCenter $radius <$startAng $endAng>
50
51 while feof(fid) ~= 1
52     currLine = fgetl(fid);
53     currLine = strsplit(strtrim(strtok(currLine,',')));
54     currLine = currLine(~cellfun('isempty',currLine));
55     if ~isempty(currLine)
56         if strcmp(currLine{1}, 'section') && strcmp(currLine{2}, 'Fiber')
57             secTagCurr = str2double(currLine{3});

```

```

58     if secTag == secTagCurr
59         secDefEnd = false;
60         while ~secDefEnd
61             currSecLine = fgetl(fid);
62             currSecLine = strsplit(strtrim(strtok(currSecLine,',')));
63             currSecLine = currSecLine(~cellfun('isempty',currSecLine));
64             if ~isempty(currSecLine)
65                 if strcmp(currSecLine{1},'patch')
66                     if strcmp(currSecLine{2},'rect')
67                         RectPatch = [RectPatch;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(
currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9})];
68                     elseif strcmp(currSecLine{2},'quad')
69                         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})];
70                     elseif strcmp(currSecLine{2},'circ')
71                         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})];
72                     end
73                     elseif strcmp(currSecLine{1},'fiber')
74                         Fiber = [Fiber;str2double(currSecLine{5}) str2double(currSecLine{2}) str2double(currSecLine{3}) str2double(currSecLine{4})];
75                     elseif strcmp(currSecLine{1},'layer')
76                         if strcmp(currSecLine{2},'straight')
77                             StraightLayer = [StraightLayer;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5})
str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9})];
78                         elseif strcmp(currSecLine{2},'circ')
79                             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})];
80                         end
81                     elseif strcmp(currSecLine{1},'')
82                         secDefEnd = true;
83                     end
84                 end
85             end
86         end
87     end
88 end
89 end
90 fclose(fid);
91
92 if ~exist('fibColor','var')
93     fibColor = NaN;
94 end
95 %% View Section
96 figure(figNum);hold on
97
98 for i1 = 1:size(RectPatch,1)
99     matTag = RectPatch(i1,1);
100     numSubdivY = RectPatch(i1,2);
101     numSubdivZ = RectPatch(i1,3);
102     y1 = RectPatch(i1,4);
103     z1 = RectPatch(i1,5);
104     yJ = RectPatch(i1,6);
105     zJ = RectPatch(i1,7);
106
107     yVec = linspace(y1,yJ,numSubdivY+1);

```

```

108 zVec = linspace(zl,zJ,numSubdivZ+1);
109
110 for i2 = 1:length(yVec)-1
111     for i3 = 1:length(zVec)-1
112         xPatch = [0,0,0,0];
113         yPatch = [yVec(i2) yVec(i2+1) yVec(i2+1) yVec(i2)];
114         zPatch = [zVec(i3) zVec(i3) zVec(i3+1) zVec(i3+1)];
115         if ismember(matTag, fibColor(:,1))
116             if size(fibColor,2) == 5
117                 p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)-1));
118                 p.FaceAlpha = fibColor(fibColor(:,1) == matTag,5);
119             else
120                 p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)));
121                 p.FaceAlpha = 1;
122             end
123         else
124             p = patch(xPatch,yPatch,zPatch,'k');
125             p.FaceAlpha = 0.2;
126         end
127     end
128 end
129
130 end
131
132 for i1 = 1:size(QuadPatch,1)
133     matTag = QuadPatch(i1,1);
134     numSubdivIJ = QuadPatch(i1,2);
135     numSubdivJK = QuadPatch(i1,3);
136
137     yI = QuadPatch(i1,4);
138     zI = QuadPatch(i1,5);
139
140     yJ = QuadPatch(i1,6);
141     zJ = QuadPatch(i1,7);
142
143     yK = QuadPatch(i1,8);
144     zK = QuadPatch(i1,9);
145
146     yL = QuadPatch(i1,10);
147     zL = QuadPatch(i1,11);
148
149     yIJ_vec = linspace(yI,yJ,numSubdivIJ+1);
150     zIJ_vec = linspace(zI,zJ,numSubdivIJ+1);
151
152     yJK_vec = linspace(yJ,yK,numSubdivJK+1);
153     zJK_vec = linspace(zJ,zK,numSubdivJK+1);
154
155     yLK_vec = linspace(yL,yK,numSubdivIJ+1);
156     zLK_vec = linspace(zL,zK,numSubdivIJ+1);
157
158     yIL_vec = linspace(yI,yL,numSubdivJK+1);
159     zIL_vec = linspace(zI,zL,numSubdivJK+1);
160
161     numLines_IJ = numSubdivIJ + 1;
162     numLines_JK = numSubdivJK + 1;
163
164     for i2 = 1:(numLines_IJ-1)

```

```
165     for i3 = 1:(numLines_JK-1)
166
167         if i2 == 1 && i3 == 1
168             y1 = yIL_vec(i3);
169             z1 = zIL_vec(i3);
170
171             y2 = yIJ_vec(i2+1);
172             z2 = zIJ_vec(i2+1);
173
174             [y3,z3] = polyxpoly(...
175                 [yIJ_vec(i2+1),yLK_vec(i2+1)],...
176                 [zIJ_vec(i2+1),zLK_vec(i2+1)],...
177                 [yIL_vec(i3+1),yJK_vec(i3+1)],...
178                 [zIL_vec(i3+1),zJK_vec(i3+1)]);
179
180             y4 = yIL_vec(i3+1);
181             z4 = zIL_vec(i3+1);
182         elseif i2 == 1 && i3 == (numLines_JK-1)
183             y1 = yIL_vec(i3);
184             z1 = zIL_vec(i3);
185
186             [y2,z2] = polyxpoly(...
187                 [yIJ_vec(i2+1),yLK_vec(i2+1)],...
188                 [zIJ_vec(i2+1),zLK_vec(i2+1)],...
189                 [yIL_vec(i3),yJK_vec(i3)],...
190                 [zIL_vec(i3),zJK_vec(i3)]);
191
192             y3 = yLK_vec(i2+1);
193             z3 = zLK_vec(i2+1);
194
195             y4 = yIL_vec(i3+1);
196             z4 = zIL_vec(i3+1);
197         elseif i2 == (numLines_IJ-1) && i3 == 1
198             y1 = yIJ_vec(i2);
199             z1 = zIJ_vec(i2);
200
201             y2 = yIJ_vec(i2+1);
202             z2 = zIJ_vec(i2+1);
203
204             y3 = yJK_vec(i3+1);
205             z3 = zJK_vec(i3+1);
206
207             [y4,z4] = polyxpoly(...
208                 [yIJ_vec(i2),yLK_vec(i2)],...
209                 [zIJ_vec(i2),zLK_vec(i2)],...
210                 [yIL_vec(i3+1),yJK_vec(i3+1)],...
211                 [zIL_vec(i3+1),zJK_vec(i3+1)]);
212         elseif i2 == (numLines_IJ-1) && i3 == (numLines_JK - 1)
213             [y1,z1] = polyxpoly(...
214                 [yIJ_vec(i2),yLK_vec(i2)],...
215                 [zIJ_vec(i2),zLK_vec(i2)],...
216                 [yIL_vec(i3),yJK_vec(i3)],...
217                 [zIL_vec(i3),zJK_vec(i3)]);
218
219             y2 = yJK_vec(i3);
220             z2 = zJK_vec(i3);
221
```

```

222     y3 = yJK_vec(i3+1);
223     z3 = zJK_vec(i3+1);
224
225     y4 = yLK_vec(i2);
226     z4 = zLK_vec(i2);
227     elseif i2 == 1 && ( 1 < i3 < (numLines_JK - 1))
228         y1 = yLJ_vec(i3);
229         z1 = zLJ_vec(i3);
230
231         [y2,z2] = polyxpoly(...
232             [yLJ_vec(i2+1),yLK_vec(i2+1)],...
233             [zLJ_vec(i2+1),zLK_vec(i2+1)],...
234             [yL_vec(i3),yJK_vec(i3)],...
235             [zL_vec(i3),zJK_vec(i3)]);
236
237         [y3,z3] = polyxpoly(...
238             [yLJ_vec(i2+1),yLK_vec(i2+1)],...
239             [zLJ_vec(i2+1),zLK_vec(i2+1)],...
240             [yL_vec(i3+1),yJK_vec(i3+1)],...
241             [zL_vec(i3+1),zJK_vec(i3+1)]);
242
243         y4 = yLJ_vec(i3+1);
244         z4 = zLJ_vec(i3+1);
245     elseif i2 == (numLines_LJ-1) && ( 1 < i3 < (numLines_JK - 1))
246         [y1,z1] = polyxpoly(...
247             [yLJ_vec(i2),yLK_vec(i2)],...
248             [zLJ_vec(i2),zLK_vec(i2)],...
249             [yL_vec(i3),yJK_vec(i3)],...
250             [zL_vec(i3),zJK_vec(i3)]);
251
252         y2 = yJK_vec(i3);
253         z2 = zJK_vec(i3);
254
255         y3 = yJK_vec(i3+1);
256         z3 = zJK_vec(i3+1);
257
258         [y4,z4] = polyxpoly(...
259             [yLJ_vec(i2),yLK_vec(i2)],...
260             [zLJ_vec(i2),zLK_vec(i2)],...
261             [yL_vec(i3+1),yJK_vec(i3+1)],...
262             [zL_vec(i3+1),zJK_vec(i3+1)]);
263     elseif (1 < i2 < (numLines_LJ-1)) && i3 == 1
264         y1 = yLJ_vec(i2);
265         z1 = zLJ_vec(i2);
266
267         y2 = yLJ_vec(i2+1);
268         z2 = zLJ_vec(i2+1);
269
270         [y3,z3] = polyxpoly(...
271             [yLJ_vec(i2+1),yLK_vec(i2+1)],...
272             [zLJ_vec(i2+1),zLK_vec(i2+1)],...
273             [yL_vec(i3+1),yJK_vec(i3+1)],...
274             [zL_vec(i3+1),zJK_vec(i3+1)]);
275         [y4,z4] = polyxpoly(...
276             [yLJ_vec(i2),yLK_vec(i2)],...
277             [zLJ_vec(i2),zLK_vec(i2)],...
278             [yL_vec(i3+1),yJK_vec(i3+1)],...

```

```

279         [zL_vec(i3+1),zJK_vec(i3+1)]];
280     elseif (1 < i2 < (numLines_IJ-1)) && i3 == (numLines_JK-1)
281         [y1,z1] = polyxpoly(...
282             [yIJ_vec(i2),yLK_vec(i2)],...
283             [zIJ_vec(i2),zLK_vec(i2)],...
284             [yIL_vec(i3),yJK_vec(i3)],...
285             [zIL_vec(i3),zJK_vec(i3)]);
286
287         [y2,z2] = polyxpoly(...
288             [yIJ_vec(i2+1),yLK_vec(i2+1)],...
289             [zIJ_vec(i2+1),zLK_vec(i2+1)],...
290             [yIL_vec(i3),yJK_vec(i3)],...
291             [zIL_vec(i3),zJK_vec(i3)]);
292
293         y3 = yLK_vec(i2+1);
294         z3 = zLK_vec(i2+1);
295
296         y4 = yLK_vec(i2);
297         z4 = zLK_vec(i2);
298     else
299         [y1,z1] = polyxpoly(...
300             [yIJ_vec(i2),yLK_vec(i2)],...
301             [zIJ_vec(i2),zLK_vec(i2)],...
302             [yIL_vec(i3),yJK_vec(i3)],...
303             [zIL_vec(i3),zJK_vec(i3)]);
304         [y2,z2] = polyxpoly(...
305             [yIJ_vec(i2+1),yLK_vec(i2+1)],...
306             [zIJ_vec(i2+1),zLK_vec(i2+1)],...
307             [yIL_vec(i3),yJK_vec(i3)],...
308             [zIL_vec(i3),zJK_vec(i3)]);
309         [y3,z3] = polyxpoly(...
310             [yIJ_vec(i2+1),yLK_vec(i2+1)],...
311             [zIJ_vec(i2+1),zLK_vec(i2+1)],...
312             [yIL_vec(i3+1),yJK_vec(i3+1)],...
313             [zIL_vec(i3+1),zJK_vec(i3+1)]);
314         [y4,z4] = polyxpoly(...
315             [yIJ_vec(i2),yLK_vec(i2)],...
316             [zIJ_vec(i2),zLK_vec(i2)],...
317             [yIL_vec(i3+1),yJK_vec(i3+1)],...
318             [zIL_vec(i3+1),zJK_vec(i3+1)]);
319     end
320
321     yPatch = [y1,y2,y3,y4];
322     zPatch = [z1,z2,z3,z4];
323     xPatch = zeros(size(yPatch));
324
325     if ismember(matTag, fibColor(:,1))
326         if size(fibColor,2) == 5
327             p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)-1));
328             p.FaceAlpha = fibColor(fibColor(:,1) == matTag,5);
329         else
330             p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)));
331             p.FaceAlpha = 1;
332         end
333     else
334         p = patch(xPatch,yPatch,zPatch, 'k');
335         p.FaceAlpha = 0.2;

```

```

336     end
337 end
338 end
339 end
340
341 for i1 = 1:size(CircPatch,1)
342     matTag = CircPatch(i1,1);
343     numSubdivCirc = CircPatch(i1,2);
344     numSubdivRad = CircPatch(i1,3);
345     yCenter = CircPatch(i1,4);
346     zCenter = CircPatch(i1,5);
347     intRad = CircPatch(i1,6);
348     extRad = CircPatch(i1,7);
349     startAng = CircPatch(i1,8)*pi/180;
350     endAng = CircPatch(i1,9)*pi/180;
351
352     rVec = linspace(intRad,extRad,numSubdivRad+1);
353     thetaVec = linspace(startAng,endAng,numSubdivCirc+1);
354
355     for i2 = 1:length(rVec)-1
356         for i3 = 1:length(thetaVec)-1
357             xPatch = [0,0,0,0];
358             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))];
359             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))];
360             if ismember(matTag, fibColor(:,1))
361                 if size(fibColor,2) == 5
362                     p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)-1));
363                     p.FaceAlpha = fibColor(fibColor(:,1) == matTag,5);
364                 else
365                     p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)));
366                     p.FaceAlpha = 1;
367                 end
368             else
369                 p = patch(xPatch,yPatch,zPatch,'k');
370                 p.FaceAlpha = 0.2;
371             end
372         end
373     end
374 end
375
376 for i1 = 1:size(CircLayer,1)
377     matTag = CircLayer(i1,1);
378     numFiber = CircLayer(i1,2);
379     areaFiber = CircLayer(i1,3);
380     yCenter = CircLayer(i1,4);
381     zCenter = CircLayer(i1,5);
382     radius = CircLayer(i1,6);
383
384     if length(CircLayer(i1,:)) > 6
385         startAng = CircLayer(i1,7)*pi/180;
386         endAng = CircLayer(i1,8)*pi/180;
387     else
388         startAng = 0.0;
389         endAng = (360 - 360/numFiber)*pi/180;
390     end

```

```
391
392 rVec = linspace(radius,radius,numFiber);
393 thetaVec = linspace(startAng,endAng,numFiber);
394 yVec = yCenter + rVec.*cos(thetaVec);
395 zVec = zCenter + rVec.*sin(thetaVec);
396
397 for i2 = 1:length(yVec)
398     if ismember(matTag, fibColor(:,1))
399         filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100, fibColor(fibColor(:,1) == matTag,2:size(fibColor,2)));
400     else
401         filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100,'k');
402     end
403 end
404 end
405
406 for i1 = 1:size(StraightLayer,1)
407     matTag = StraightLayer(i1,1);
408     numFiber = StraightLayer(i1,2);
409     areaFiber = StraightLayer(i1,3);
410     yStart = StraightLayer(i1,4);
411     zStart = StraightLayer(i1,5);
412     yEnd = StraightLayer(i1,6);
413     zEnd = StraightLayer(i1,7);
414
415     yVec = linspace(yStart,yEnd,numFiber);
416     zVec = linspace(zStart,zEnd,numFiber);
417
418     for i2 = 1:length(yVec)
419         if ismember(matTag, fibColor(:,1))
420             filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100, fibColor(fibColor(:,1) == matTag,2:size(fibColor,2)));
421         else
422             filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100,'k');
423         end
424     end
425 end
426
427 for i1 = 1:size(Fiber,1)
428     matTag = Fiber(i1,1);
429     yVec = Fiber(i1,2);
430     zVec = Fiber(i1,3);
431     areaFiber = Fiber(i1,4);
432     if ismember(matTag, fibColor(:,1))
433         filledCircle([0,yVec,zVec],sqrt(areaFiber/pi),100, fibColor(fibColor(:,1) == matTag,2:size(fibColor,2)));
434     else
435         filledCircle([0,yVec,zVec],sqrt(areaFiber/pi),100,'k');
436     end
437 end
438
439 ylabel('y')
440 xlabel('z')
441 view([1,0,0])
442 box on;
443 axis square;
444 axis equal;
445
446 end
447
```



```
448 % Helper function to plot filled circle
449 function h = filledCircle(center,r,N,color)
450 if isnumeric(color) && length(color) == 4
451     alpha = color(4);
452     color = color(1:3);
453 else
454     alpha = 1;
455 end
456 THETA=linspace(0,2*pi,N);
457 RHO=ones(1,N)*r;
458 [Y,Z] = pol2cart(THETA,RHO);
459 Y=Y+center(2);
460 Z=Z+center(3);
461 X = zeros(size(Y));
462 h=patch(X,Y,Z,color);
463 h.FaceAlpha = alpha;
464 end
465
```