

A Main_Coupled.m

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %                               UPC                               %
3 %       Prasad ADHAV - FEF Assignment 2018 - Part 3           %
4 %This code solves the Part 3 of the Assignment which is Coupled problem %
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7
8
9 clear all;
10 close all;
11 clc
12
13 addpath('Func_ReferenceElement')
14 %%
15 %Given Material Parameters in Problem
16 sigma_F = 0.25; %s^-1
17 sigma_G = 2;    %s^-1
18 sigma_GF = 0.5; %s^-1
19 D_F = 5;        %micro.m/s
20 D_G = 15;       %micro.m/s
21 mu = 1000;
22
23 Nr = cinput('No. of Nodes in Radial =',20);
24 Ntheta = cinput('No. of Nodes in Radial =',10);
25
26 timef = cinput('End time',2*pi);
27 ntime = cinput('No. of Time Steps',200);
28 dt = timef/ntime;
29
30 % Selection of Element type and interpolation degree
31 % For Our case we are using Biinear Quadrilateral elements, so we choose
32 % elemV = 0 and the degree for velocity is 2(Quadratic) while degree for
33 % pressure is 1(Linear)
34 % (0: quadrilaterals, 1: triangles, 11: triangles with bubble function)
35 elemV = 0;
36 degreeV = 1;
37 degreeP = 1;
38 elem = 0;
39 elemP = elemV;
40
41 %% Creating and plotting mesh
42
43 referenceElement = SetReferenceElementStokes(elemV,degreeV,elemP,degreeP);
44 Xe_ref = referenceElement.Xe_ref;
45
46 %Pressure Mesh
47 [XP,TP] = createMesh(Nr, Ntheta,degreeP);
48 figure(1)
49 PlotMesh(TP,XP,elemV,'b-');
50
51 %Velocity Mesh
52 [X,T,theta] = createMesh(Nr,Ntheta,degreeV)
53 figure(2);
```

```

54 PlotMesh(T,X,elemP,'r-');
55
56 dom = [XP(1,1), XP(size(XP,1),1), XP(1,2), XP(size(XP,1),2)]
57
58 %% Matrices
59
60 % Matrices arising from the discretization
61 %[K,G,f] = StokesSystem(X,T,XP,TP,referenceElement);
62 [K,G,f] = StokesSystem2(X,T,XP,TP,referenceElement);
63 K = mu*K;
64 [ndofP,ndofV] = size(G);
65
66
67 % Quadrature
68 ngaus=4;
69 [pospg,wpg] = Quadrature(elemV,ngaus)
70 % Shape Functions
71 [N,Nxi,Neta] = ShapeFunc(elemV,degreeV,pospg);
72
73 [Tf,Tu] = boundaryMatrices(X,T,elem,degreeP,Xe_ref)
74 M = CreMassMat(X,T,pospg,wpg,N,Nxi,Neta);
75 %C = CreConvMat(X,T,Conv,pospg,wpg,N,Nxi,Neta);
76 K = CreStiffMat(X,T,pospg,wpg,N,Nxi,Neta);
77
78 %%
79 %convert no. of elements to numbers of nodes
80 nnr = Nr + 1;
81 nNtheta = Ntheta + 1;
82
83 %Find Dirichlet nodes
84 Y1=1:(nNtheta);
85 Y2=(nnr-1)*nNtheta+1:(nNtheta)*(nnr);
86 Y1=Y1';
87 Y2=Y2';
88
89 Nodes_Dir = [Y1; Y2];
90 Dir_dof = 2*length(Nodes_Dir);
91
92
93
94 C = [2*Nodes_Dir - 1; 2*Nodes_Dir];
95 Cstep = reshape(C, Dir_dof/2, 2);
96 Cbal = reshape(Cstep', Dir_dof, 1);
97 A = zeros(Dir_dof,ndofV);
98 A(:,Cbal) = eye(Dir_dof);
99
100 r = linspace(15,25,Nr+1);
101 theta_max = 10*pi/180 ;
102 theta = linspace(-theta_max,theta_max,Ntheta+1) ;
103
104 b = [(-0.3*cos((pi/2)-(theta)))';
105      (-0.15*cos(pi/2-(theta)))';
106      (-0.3*sin((pi/2-(theta))))';
107      (-0.15*sin(pi/2-(theta))))'];
108
109 b_Dir = reshape(b', Dir_dof,1);

```

```

110 vel_Dir = [Cbal b_Dir];
111
112 %% F Solution
113
114 %BC for F
115 nPt = size(X,1);
116 F = zeros(nPt, ntime + 1);
117 Dir_val =80;
118 coun = 0;
119
120 for i = 1:size(X,1)
121     Fdir = abs(X(i,1)^2 + X(i,2)^2 - 25^2);
122     if Fdir <= 10^-6
123         coun = coun + 1;
124         F(i,:) = Dir_val;
125     end
126 end
127
128
129 confined = 1;
130 if confined
131     nunkP = ndofP-1;
132     G(1,:) = [];
133 else
134     nunkP = ndofP;
135 end
136
137 nD = [(Ntheta + 1)*(Nr + 1) - Ntheta:(Ntheta + 1)*(Nr + 1)]';
138 Cdir = [nD, zeros(length(nD),1)];
139 nDirF = size(Cdir,1);
140
141 AccdF = zeros(nDirF, nPt);
142 AccdF(:, Cdir(:,1)) = eye(nDirF);
143 bccdF = Cdir(:,2);
144
145 vel(:,1) = zeros(ndofV,1);
146 vel(vel_Dir(:,1)) = vel_Dir(:,2);
147
148 %Af = M + 0.5*C*dt + 0.5*D_F*K*dt + 0.5*sigma_F*M*dt;
149 %Bf = M + 0.5*(-C*dt - D_F*K*dt - sigma_F*M*dt);
150 ff = zeros(nPt,1);
151
152 %% G Solution
153
154 G = zeros(nPt, ntime + 1);
155 Ag = M + 0.5*D_G*K*dt + 0.5*sigma_G*M*dt;
156 Bg = -D_G*K*dt - sigma_G*M*dt;
157 Cg = sigma_GF*M*dt;
158 fg = zeros(nPt,1);
159
160 Atot = Ag;
161 [l2, u2] = lu(Atot);
162
163
164
165 %% Postprocess

```

```

166 %
167 % for n= 1:ntime
168 %
169 %     %initialize
170 %     veli = 1;  Fmat = 1;
171 %     icoun = 0;
172 %     g = F(:,n);
173 %
174 %     tol = 10^-6
175 %     while veli>=tol
176 %         icoun = icoun + 1;
177 %         fu = -Tf*g(:,icoun);
178 %         for i =1:size(K,1)
179 %             fu_tot(i,1) = fu(i,1) - K(i,vel_Dir(:,1))*Cbal;
180 %         end
181 %
182 %         for i = 1:Dir_dof
183 %             fu_tot(vel_Dir(i,1)) = vel_Dir(i,2);
184 %         end
185 %
186 %         vel(:,icoun+1) = A\fu_tot;
187 %         vel_Dir = vel(:,icoun+1);
188 %
189 %         Conv = reshape(vel,2,[])';
190 %         C = CreConvMat(X,T,Conv,postpg,wpg,N,Nxi,Neta);
191 %
192 %         %Solution of F
193 %         Af = M + 0.5*C*DT + 0.5*D_F*K*dt + 0.5*sigma_F*M*DT;
194 %         Bf = M + 0.5*(-C*DT - D_F*K*dt - sigma_F*M*DT);
195 %         Ff = zeros(nPt,1);
196 %
197 %         Afloop = [Af Accd_F';Accd_F zeros(nDir_F,nDir_F)];
198 %         [L1,U1] = lu(Afloop);
199 %
200 %         Bfloop = [Bf*F(:,n)+ Ff; bccd_F];
201 %
202 %         Lag_1 = U1\(L1\Bfloop);
203 %         g(:,icoun+1) = Lag_1(1:nPt);
204 %     end
205 %
206 %     vel2(:,n) = vel(:,icoun+1);
207 %     vel = vel2(:,n);
208 %     Conv = reshape(vel,2,[])';
209 %     C = CreConvMat(X,T,Conv,postpg,wpg,N,Nxi,Neta);
210 %
211 %     Af = M + 0.5*C*dt + 0.5*D_F*K*dt + 0.5*sigma_F*M*dt;
212 %     Bf = (-C*dt - D_F*K*dt - sigma_F*M*dt);
213 %     Ff = zeros(nPt,1);
214 %
215 %     Afloop = [Af Accd_F';Accd_F zeros(nDir_F,nDir_F)];
216 %     [L1,U1] = lu(Afloop);
217 %     Bfloop = [Bf*F(:,n)+ Ff; bccd_F];
218 %     aux_f = U1\(L1\Bfloop);
219 %     aux_F = U1\(L1\Bfloop);
220 %     F(:,n+1) = F(:,n) + aux_F(1:nPt);
221 %

```

```

222 %     btot = [Bg*G(:,n) - 0.5*C_G*aux_f(1:nPt) + 0.5*Cg*F(:,n+1) + fg];
223 %     aux_G = U2\ (L2\btot);
224 %     G(:,n+1) = G(:,n) + aux_G(1:nPt);
225 %
226 % end
227 %
228 %
229 %
230 % figure(1)
231 % nPt = size(X,1);
232 % figure;
233 % quiver(X(1:nPt,1),X(1:nPt,2),Conv(1:nPt,1),Conv(1:nPt,2));
234 % hold on
235 % axis equal; axis tight
236 %
237 %
238 %
239 % figure(2)
240 % PlotResults(X,T,Conv(:,1),referenceElement.elemV,referenceElement.degreeV
    )
241 % figure(3)
242 % PlotResults(X,T,Conv(:,2),referenceElement.elemV,referenceElement.degreeV
    )
243 % v = sqrt((Conv(:,1).^2)+(Conv(:,2).^2));
244 % figure(4)
245 % PlotResults(X,T,v,referenceElement.elemV,referenceElement.degreeV)
246 %
247 %
248 %
249 % if max(F(:,ntime + 1))<100 && min(F(:,ntime + 1))>-100
250 %
251 %     figure(5); clf;
252 %
253 %     [xx,yy,sol] = MatSol(X,Ntheta,Nr,F(:,ntime + 1));
254 %     surface(xx,yy,sol);
255 %     view([40,30])
256 %     axis auto
257 %
258 %     grid on;
259 %
260 %
261 %     figure(6); clf;
262 %     set(gca,'FontSize',12);
263 %     [C,h]=contour(xx,yy,sol);
264 %
265 %     clabel(C,h);
266 %     axis auto
267 %
268 %     figure(10); clf;
269 %     peli = moviein(ntime + 1);
270 %     axis auto
271 %
272 %     for n=1:ntime + 1
273 %         [xx,yy,sol] = MatSol(X,Ntheta,Nr,F(:,n));
274 %         surf(xx,yy,sol);
275 %         zlim([0 90])

```

```

276 %     ylim([13 25])
277 %     pause(0.0001)
278 %         peli(:,n) = getframe;
279 %     end
280 % end
281 % pause(0.1);
282
283 if max(G(:,ntime + 1))<100 && min(G(:,ntime + 1))>-100
284
285     figure(11); clf;
286
287     [xx,yy,sol] = MatSol(X,Ntheta,Nr,G(:,ntime + 1));
288     surface(xx,yy,sol);
289     view([40,30])
290
291     grid on;
292
293
294     figure(12); clf;
295     set(gca, 'FontSize',12);
296     [C,h]=contour(xx,yy,sol);
297
298     clabel(C,h);
299
300 %     figure(20); clf;
301 %     peli = moviein(ntime + 1);
302 %
303 %     for n=1:ntime + 1
304 %         [xx,yy,sol] = MatSol(X,Ntheta,Nr,G(:,n));
305 %         surf(xx,yy,sol);
306 %         zlim([0 15])
307 %         ylim([13 25])
308 %         pause(0.0001)
309 %         peli(:,n) = getframe;
310 %     end
311 end

```