A *Main_Coupled.m*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                              UPC                                        %
%         Prasad ADHAV - FEF Assignment 2018 - Part 3                     %
%This code solves the Part 3 of the Assignment which is Coupled problem   %
%                                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


clear all;
close all;
clc

addpath('Func_ReferenceElement')
%%
%Given Material Parameters in Problem
sigma_F = 0.25; %s^-1
sigma_G = 2;    %s^-1
sigma_GF = 0.5; %s^-1
D_F = 5;        %micro.m/s
D_G = 15;       %micro.m/s
mu = 1000;

Nr  = cinput('No. of Nodes in Radial =',20);
Ntheta  = cinput('No. of Nodes in Radial =',10);

timef = cinput('End time',2*pi);
ntime = cinput('No. of Time Steps',200);
dt = timef/ntime;

% Selection of Element type and interpolation degree
% For Our case we are using Biinear Quadrilateral elements, so we choose
% elemV = 0 and the degree for velocity is 2(Quadratic) while degree for
% pressure is 1(Linear)
% (0: quadrilaterals, 1: triangles, 11: triangles with bubble function)
elemV = 0;
degreeV = 1;
degreeP = 1;
elem = 0;
elemP = elemV;

%% Creating and plotting mesh

referenceElement = SetReferenceElementStokes(elemV,degreeV,elemP,degreeP);
Xe_ref = referenceElement.Xe_ref;

%Pressure Mesh
[XP,TP] = createMesh(Nr, Ntheta,degreeP);
figure(1)
PlotMesh(TP,XP,elemV,'b-');

%Velocity Mesh
[X,T,theta] = createMesh(Nr,Ntheta,degreeV)
figure(2);
```

```
PlotMesh(T,X,elemP,'r-');

dom = [XP(1,1), XP(size(XP,1),1), XP(1,2), XP(size(XP,1),2)]

%% Matrices

% Matrices arising from the discretization
%[K,G,f] = StokesSystem(X,T,XP,TP,referenceElement);
[K,G,f] = StokesSystem2(X,T,XP,TP,referenceElement);
 K = mu*K;
[ndofP,ndofV] = size(G);


% Quadrature
 ngaus=4;
[pospg,wpg] = Quadrature(elemV,ngaus)
% Shape Functions
[N,Nxi,Neta] = ShapeFunc(elemV,degreeV,pospg);

[Tf,Tu] = boundaryMatrices(X,T,elem,degreeP,Xe_ref)
M = CreMassMat(X,T,pospg,wpg,N,Nxi,Neta);
%C = CreConvMat(X,T,Conv,pospg,wpg,N,Nxi,Neta);
K = CreStiffMat(X,T,pospg,wpg,N,Nxi,Neta);

%%
%convert no. of elements to numbers of nodes
nnr = Nr + 1;
nNtheta = Ntheta + 1;

%Find Diritchlet nodes
Y1=1:(nNtheta);
Y2=(nnr-1)*nNtheta+1:(nNtheta)*(nnr);
Y1=Y1';
Y2=Y2';

Nodes_Dir = [Y1; Y2];
Dir_dof = 2*length(Nodes_Dir);



C = [2*Nodes_Dir - 1; 2*Nodes_Dir];
Cstep = reshape(C, Dir_dof/2, 2);
Cbal = reshape(Cstep', Dir_dof, 1);
A = zeros(Dir_dof,ndofV);
A(:,Cbal) = eye(Dir_dof);

r = linspace(15,25,Nr+1);
theta_max = 10*pi/180 ;
theta = linspace(-theta_max,theta_max,Ntheta+1) ;

b = [(-0.3*cos((pi/2)-(theta)))';
    (-0.15*cos(pi/2-(theta)))';
    (-0.3*sin((pi/2-(theta))))';
    (-0.15*sin((pi/2-(theta))))'];

b_Dir = reshape(b', Dir_dof,1);
```

```
vel_Dir = [Cbal b_Dir];

%% F Solution

%BC for F
nPt = size(X,1);
F = zeros(nPt, ntime + 1);
Dir_val =80;
coun = 0;

for i = 1:size(X,1)
    Fdir = abs(X(i,1)^2 + X(i,2)^2 - 25^2);
    if Fdir <= 10^-6
        coun = coun + 1;
        F(i,:) = Dir_val;
    end
end


confined = 1;
if confined
   nunkP = ndofP-1;
   G(1,:) = [];
else
   nunkP = ndofP;
end

nD = [(Ntheta + 1)*(Nr + 1) - Ntheta:(Ntheta + 1)*(Nr + 1)]';
Cdir = [nD, zeros(length(nD),1)];
nDirF = size(Cdir,1);

AccdF = zeros(nDirF, nPt);
AccdF(:, Cdir(:,1)) = eye(nDirF);
bccdF = Cdir(:,2);

vel(:,1) = zeros(ndofV,1);
vel(vel_Dir(:,1)) = vel_Dir(:,2);

%Af = M + 0.5*C*dt + 0.5*D_F*K*dt + 0.5*sigma_F*M*dt;
%Bf = M + 0.5*(-C*dt - D_F*K*dt - sigma_F*M*dt);
ff = zeros(nPt,1);

%% G Solution

G = zeros(nPt, ntime + 1);
Ag = M + 0.5*D_G*K*dt + 0.5*sigma_G*M*dt;
Bg = -D_G*K*dt - sigma_G*M*dt;
Cg = sigma_GF*M*dt;
fg = zeros(nPt,1);

Atot = Ag;
[l2, u2] = lu(Atot);



%% Postprocess
```

```
%
% for n= 1:ntime
%
%     %initialize
%     veli = 1;  Fmat = 1;
%     icoun = 0;
%     g = F(:,n);
%
%     tol = 10^-6
%     while veli>=tol
%         icoun = icoun + 1;
%         fu = -Tf*g(:,icoun);
%         for i =1:size(K,1)
%             fu_tot(i,1) = fu(i,1) - K(i,vel_Dir(:,1))*Cbal;
%         end
%
%         for i = 1:Dir_dof
%             fu_tot(vel_Dir(i,1)) = vel_Dir(i,2);
%         end
%
%         vel(:,icoun+1) = A\fu_tot;
%         vel_Dir = vel(:,icoun+1);
%
%         Conv = reshape(vel,2,[])';
%         C = CreConvMat(X,T,Conv,pospg,wpg,N,Nxi,Neta);
%
%         %Solution of F
%         Af = M + 0.5*C*DT + 0.5*D_F*K*dt + 0.5*sigma_F*M*DT;
%         Bf = M + 0.5*(-C*DT - D_F*K*dt - sigma_F*M*DT);
%         Ff = zeros(nPt,1);
%
%         Afloop = [Af Accd_F';Accd_F zeros(nDir_F,nDir_F)];
%         [L1,U1] = lu(Afloop);
%
%         Bfloop = [Bf*F(:,n)+ Ff; bccd_F];
%
%         Lag_1 = U1\(L1\Bfloop);
%         g(:,icoun+1) =  Lag_1(1:nPt);
%     end
%
%     vel2(:,n) = vel(:,icoun+1);
%     vel = vel2(:,n);
%     Conv = reshape(vel,2,[])';
%     C = CreConvMat(X,T,Conv,pospg,wpg,N,Nxi,Neta);
%
%     Af = M + 0.5*C*dt + 0.5*D_F*K*dt + 0.5*sigma_F*M*dt;
%     Bf = (-C*dt - D_F*K*dt - sigma_F*M*dt);
%     Ff = zeros(nPt,1);
%
%     Afloop = [Af Accd_F';Accd_F zeros(nDir_F,nDir_F)];
%     [L1,U1] = lu(Afloop);
%     Bfloop = [Bf*F(:,n)+ Ff; bccd_F];
%     aux_f =  U1\(L1\Bfloop);
%     aux_F = U1\(L1\Bfloop);
%     F(:,n+1) = F(:,n) + aux_F(1:nPt);
%
```

```
%     btot = [Bg*G(:,n) - 0.5*C_G*aux_f(1:nPt) + 0.5*Cg*F(:,n+1) + fg];
%     aux_G  = U2\(L2\btot);
%     G(:,n+1) = G(:,n) + aux_G(1:nPt);
%
% end
%
%
%
% figure(1)
% nPt = size(X,1);
% figure;
% quiver(X(1:nPt,1),X(1:nPt,2),Conv(1:nPt,1),Conv(1:nPt,2));
% hold on
% axis equal; axis tight
%
%
%
% figure(2)
% PlotResults(X,T,Conv(:,1),referenceElement.elemV,referenceElement.degreeV
  )
% figure(3)
% PlotResults(X,T,Conv(:,2),referenceElement.elemV,referenceElement.degreeV
  )
% v = sqrt((Conv(:,1).^2)+(Conv(:,2).^2));
% figure(4)
% PlotResults(X,T,v,referenceElement.elemV,referenceElement.degreeV)
%
%
%
% if max(F(:,ntime + 1))<100 &&  min(F(:,ntime + 1))>-100
%
%     figure(5); clf;
%
%     [xx,yy,sol] = MatSol(X,Ntheta,Nr,F(:,ntime + 1));
%     surface(xx,yy,sol);
%     view([40,30])
%     axis auto
%
%     grid on;
%
%
%     figure(6); clf;
%     set(gca,'FontSize',12);
%     [C,h]=contour(xx,yy,sol);
%
%     clabel(C,h);
%     axis auto
%
%     figure(10); clf;
%     peli = moviein(ntime + 1);
%     axis auto
%
%     for n=1:ntime + 1
%         [xx,yy,sol] = MatSol(X,Ntheta,Nr,F(:,n));
%         surf(xx,yy,sol);
%         zlim([0 90])
```

```
%      ylim([13 25])
%      pause(0.0001)
%          peli(:,n) = getframe;
%      end
% end
% pause(0.1);

if max(G(:,ntime + 1))<100 &&  min(G(:,ntime + 1))>-100

    figure(11); clf;

    [xx,yy,sol] = MatSol(X,Ntheta,Nr,G(:,ntime + 1));
    surface(xx,yy,sol);
    view([40,30])

    grid on;


    figure(12); clf;
    set(gca,'FontSize',12);
    [C,h]=contour(xx,yy,sol);

    clabel(C,h);

%      figure(20); clf;
%      peli = moviein(ntime + 1);
%
%      for n=1:ntime + 1
%          [xx,yy,sol] = MatSol(X,Ntheta,Nr,G(:,n));
%          surf(xx,yy,sol);
%          zlim([0 15])
%      ylim([13 25])
%      pause(0.0001)
%          peli(:,n) = getframe;
%      end
end
```