```matlab
%--------------------------------------------------------------------
%%%% "TURBINE_ANSYS.m"
%%%% IMPLEMENTED BY SERGIO CUSTODIO - engsergiocustodio@gmail.com
%%%% CONTRIBUTIONS OF HELDER SANTANA
%--------------------------------------------------------------------

function TURBINE_ANSYS(journal_name,geom_name,model_name,shape_name,root_name,perfil_name,npontos_perfil,xlongarina,ylongarina,dlongarina,perfil_fim_longarina,load_name,vel_rotation)

create_journal(journal_name,geom_name,model_name);
create_geom(geom_name,shape_name,root_name,perfil_name,npontos_perfil,xlongarina,ylongarina,dlongarina,perfil_fim_longarina);
create_model(model_name,load_name,vel_rotation);

%--------------------------------------------------------------------
function create_journal(journal_name,geom_name,model_name)

fid = fopen(journal_name,'w+');

fprintf(fid,['template1 = GetTemplate(\n']);
fprintf(fid,['  TemplateName="Static Structural",\n']);
fprintf(fid,['  Solver="ANSYS")\n']);
fprintf(fid,['system1 = template1.CreateSystem()\n']);
fprintf(fid,['geometry1 = system1.GetContainer(ComponentName="Geometry")\n']);
fprintf(fid,['geometry1.Edit()\n']);
bld = [cd,'/',geom_name];
bld = strsplit(bld,'\');
bld = strjoin(bld,'/');
fprintf(fid,['script = open(''',bld,''',''r'')\n']);
fprintf(fid,['geometry1.SendCommand(Command=script.read())\n']);
fprintf(fid,['geometry1.Exit()\n']);
fprintf(fid,['component1 = system1.GetComponent(Name="Model")\n']);
fprintf(fid,['component1.Refresh()\n']);
fprintf(fid,['model1 = system1.GetContainer(ComponentName="Model")\n']);
fprintf(fid,['model1.Edit()\n']);
str = [cd,'/',model_name];
str = strsplit(str,'\');
str = strjoin(str,'/');
fprintf(fid,['script = open(''',str,''',''r'')\n']);
fprintf(fid,['model1.SendCommand(Command=script.read())\n']);
fclose(fid);
%--------------------------------------------------------------------
function create_geom(geom_name,shape_name,root_name,perfil_name,npontos_perfil,xlongarina,ylongarina,dlongarina,perfil_fim_longarina)

bladeshape=load(shape_name);
rootshape=load(root_name);
nprofiles=size(bladeshape,1);
t=size(rootshape,1);
npoints=npontos_perfil;
r=bladeshape(:,1);
c=bladeshape(:,2);
b=bladeshape(:,3);
el =rootshape;el = el([1:t],[1:4]);
[FXr,FYr,Ri] = plot_shape3Da(perfil_name,npontos_perfil,nprofiles,r,c,b,el);
xc=xlongarina;
yc=ylongarina;
d1=dlongarina;
l=perfil_fim_longarina;

fid = fopen(geom_name,'w+');
fprintf(fid,'ag.gui.setUnits(ag.c.UnitMillimeter, ag.c.UnitDegree, ag.c.No);\n');

% Planos - Funcao
fprintf(fid,'function doPlane(Name,Offset)\n');
fprintf(fid,'{\n');
fprintf(fid,'var planeXY  = agb.GetXYPlane();\n');
fprintf(fid,'var newPlane = agb.PlaneFromPlane(planeXY);\n');
fprintf(fid,'newPlane.Name = Name;\n');
fprintf(fid,'newPlane.AddTransform(agc.XformZOffset, Offset);\n');
fprintf(fid,'agb.regen();\n');
fprintf(fid,'return newPlane;\n');
fprintf(fid,'}\n');

% Planos - Criar
for j=1:(nprofiles+t)
    fprintf(fid,['Plane',num2str(j),' = doPlane("Plane',num2str(j),'",',num2str(Ri(j)),');\n']);
end
% Sketch - Funcao
fprintf(fid,'function doSketch(plane,Name,splineX,splineY)\n');
fprintf(fid,'{\n');
fprintf(fid,'p = new Object();\n');
fprintf(fid,'agb.SetActivePlane (plane);\n');
fprintf(fid,'p.Plane  = agb.GetActivePlane();\n');
fprintf(fid,'p.Origin = p.Plane.GetOrigin();\n');
fprintf(fid,'p.XAxis  = p.Plane.GetXAxis();\n');
fprintf(fid,'p.YAxis  = p.Plane.GetYAxis();\n');
fprintf(fid,'p.Sk1 = p.Plane.NewSketch();\n');
fprintf(fid,'p.Sk1.Name = Name;\n');
fprintf(fid,'with (p.Sk1)\n');
fprintf(fid,'{\n');
fprintf(fid,'p.Sp1 = SplineBegin();\n');
fprintf(fid,'with (p.Sp1)\n');
fprintf(fid,'{\n');
fprintf(fid,'SplineFlexibility = agc.Yes;\n');

fprintf(fid,'for (itr in splineX)\n');
fprintf(fid,'{\n');
fprintf(fid,'SplineXY( splineX[itr], splineY[itr]);\n');
fprintf(fid,'}\n');

fprintf(fid,'SplineFitPtEnd();\n');
fprintf(fid,'}\n');
fprintf(fid,'}\n');
fprintf(fid,'p.Plane.EvalDimCons();\n');
fprintf(fid,'return p;\n');
fprintf(fid,'}\n');

for j=1:(nprofiles+t)
    % Sketch - Splines
    fprintf(fid,['var splineX = new Array(',num2str(npoints),');\n']);
    fprintf(fid,['var splineY = new Array(',num2str(npoints),');\n']);
    for s=1:npoints
        fprintf(fid,['splineX[',num2str(s-1),']=',num2str(FXr(j,s)),';\n']);
        fprintf(fid,['splineY[',num2str(s-1),']=',num2str(FYr(j,s)),';\n']);
    end
    % Sketch - Criar
    fprintf(fid,['skPlane',num2str(j),' = doSketch(Plane',num2str(j),',"Sketch',num2str(j),'",splineX,splineY',');\n']);

    fprintf(fid,['ag.selectedFeature = ag.gui.TreeviewFeature(p.Sk1.Name, 0);\n']);
    fprintf(fid,['var SSk1 = ag.gui.CreateSurfSk();\n']);
    % fprintf(fid,['SSk1.Name = "name";\n']);
    fprintf(fid,['SSk1.Operation = ag.c.Frozen;\n']);
    fprintf(fid,['SSk1.WithPlaneNormal = ag.c.Yes;\n']);
    fprintf(fid,['ag.listview.ActivateItem("Thickness (>=0)");\n']);
    fprintf(fid,['ag.listview.ItemValue = "0,1";\n']);
end
fprintf(fid,['function planeSketchesOnly(p)\n']);
fprintf(fid,['{\n']);
fprintf(fid,['agb.SetActivePlane (Plane1)\n']);
fprintf(fid,['p.Plane  = agb.GetActivePlane();\n']);
fprintf(fid,['p.Origin = p.Plane.GetOrigin();\n']);
fprintf(fid,['p.XAxis  = p.Plane.GetXAxis();\n']);
fprintf(fid,['p.YAxis  = p.Plane.GetYAxis();\n']);
fprintf(fid,['p.Sk2 = p.Plane.NewSketch();\n']);
fprintf(fid,['p.Sk2.Name ="longarina";\n']);
fprintf(fid,['with (p.Sk2)\n']);
fprintf(fid,['{\n']);
raio=num2str(d1/2,'%f');
raio = strsplit(raio,'.');
raio = strjoin(raio,',');
fprintf(fid,['p.Circ1 = Circle(',num2str(xc),',',num2str(yc),',"',raio,'");\n']);
%fprintf(fid,['p.Circ2 = Circle(',num2str(xc),',',num2str(yc),',"',num2str(d2/2),'");\n']);
fprintf(fid,['}\n']);
fprintf(fid,['p.Sk3 = p.Plane.NewSketch();\n']);
fprintf(fid,['p.Sk3.Name ="longarina_remove";\n']);
fprintf(fid,['with (p.Sk3)\n']);
fprintf(fid,['{\n']);
fprintf(fid,['p.Circ1 = Circle(',num2str(xc),',',num2str(yc),',"',raio,'");\n']);
fprintf(fid,['}\n']);
fprintf(fid,['p.Plane.EvalDimCons();\n']);
fprintf(fid,['return p;\n']);
fprintf(fid,['}\n']);

fprintf(fid,['var Skin1 = agb.Skin(agc.Add, agc.Yes, 0.0, 0.0);\n']);
fprintf(fid,['Skin1.Name = "Skin";\n']);

for j=1:(nprofiles+t)
    fprintf(fid,['Skin1.AddBaseObject(skPlane',num2str(j),'.Sk1);\n']);
end
fprintf(fid,['agb.Regen();\n']);

fprintf(fid,['for (var i = 1; i<=',num2str(nprofiles-1),'; i++)\n']);
fprintf(fid,['{\n']);
fprintf(fid,['var namedSelection = ag.gui.CreateSelectionSet();\n']);
fprintf(fid,['ag.gui.Commit();\n']);
fprintf(fid,['namedSelection.Name = "Sup" +i\n']);
fprintf(fid,['var face = ag.m.ModelFaces(i+',num2str(nprofiles+2*t),');\n']);
fprintf(fid,['agb.AddSelect(agc.TypeFace, face);\n']);
fprintf(fid,['ag.listview.ActivateItem("Geometry");\n']);
fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
fprintf(fid,['ag.Regen;\n']);
fprintf(fid,['}\n']);


fprintf(fid,['var namedSelection = ag.gui.CreateSelectionSet();\n']);
fprintf(fid,['ag.gui.Commit();\n']);
fprintf(fid,['namedSelection.Name = "SupTotal";\n']);
fprintf(fid,['for (var i = 1; i<=',num2str(t+nprofiles-1),'; i++)\n']);
fprintf(fid,['{\n']);
fprintf(fid,['var face = ag.m.ModelFaces(i+',num2str(nprofiles+t),');\n']);
fprintf(fid,['agb.AddSelect(agc.TypeFace, face);\n']);
fprintf(fid,['}\n']);
fprintf(fid,['ag.listview.ActivateItem("Geometry");\n']);
fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
fprintf(fid,['ag.Regen;\n']);


fprintf(fid,['var namedSelection = ag.gui.CreateSelectionSet();\n']);
fprintf(fid,['ag.gui.Commit();\n']);
fprintf(fid,['namedSelection.Name = "Support";\n']);
fprintf(fid,['var face = ag.m.ModelFaces(1);\n']);
fprintf(fid,['agb.AddSelect(agc.TypeFace, face);\n']);
fprintf(fid,['ag.listview.ActivateItem("Geometry");\n']);
fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
fprintf(fid,['ag.Regen;\n']);


fprintf(fid,['var ps1 = planeSketchesOnly (new Object());\n']);
fprintf(fid,['var ext2 = agb.Extrude(agc.Add, ps1.Sk3, agc.DirNormal, agc.ExtendFixed, ',num2str(Ri(l+t)-Ri(1)),', agc.ExtendFixed, 0.0, agc.yes, 0.0, 0.0);\n']);
fprintf(fid,['agb.Regen();\n']);
fprintf(fid,['var Pat = ag.gui.CreateBoolean();\n']);
fprintf(fid,['ag.listview.ActivateItem("Operation");\n']);
fprintf(fid,['ag.listview.ItemValue = "Subtract";\n']);
fprintf(fid,['ag.listview.ActivateItem("Target Bodies");\n']);
fprintf(fid,['ag.bodyPick;\n']);
fprintf(fid,['for (var i = 0; i<=',num2str(nprofiles+t-1),'; i++)\n']);
fprintf(fid,['{\n']);
fprintf(fid,['ad1 = ag.fm.Body(i);\n']);
fprintf(fid,['agb.AddSelect(agc.TypeBody, ad1);\n']);
fprintf(fid,['}\n']);
fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
fprintf(fid,['ag.listview.ActivateItem("Tool Bodies");\n']);
fprintf(fid,['ag.bodyPick;\n']);
fprintf(fid,['ad2 = ag.fm.Body(',num2str(nprofiles+t+1),');\n']);
fprintf(fid,['agb.AddSelect(agc.TypeBody, ad2);\n']);
fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
fprintf(fid,['agb.Regen();\n']);

fprintf(fid,['var ext1 = agb.Extrude(agc.Add, ps1.Sk2, agc.DirNormal, agc.ExtendFixed, ',num2str(Ri(l+t)-Ri(1)),', agc.ExtendFixed, 0.0, agc.Yes, 0.0, 0.0);\n']);
fprintf(fid,['agb.Regen();\n']);

fprintf(fid,['var Pat = ag.gui.CreateBoolean();\n']);
fprintf(fid,['ag.listview.ActivateItem("Operation");\n']);
fprintf(fid,['ag.listview.ItemValue = "Imprint Faces";\n']);
fprintf(fid,['ag.listview.ActivateItem("Tool Bodies");\n']);
fprintf(fid,['ag.bodyPick;\n']);
fprintf(fid,['for (var i = 0; i<=',num2str(nprofiles+t-1),'; i++)\n']);
fprintf(fid,['{\n']);
fprintf(fid,['ad1 = ag.fm.Body(i);\n']);
fprintf(fid,['agb.AddSelect(agc.TypeBody, ad1);\n']);
fprintf(fid,['}\n']);
fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
fprintf(fid,['ag.listview.ActivateItem("Target Bodies");\n']);
fprintf(fid,['ag.bodyPick;\n']);
fprintf(fid,['ad2 = ag.fm.Body(',num2str(nprofiles+t+1),');\n']);
fprintf(fid,['agb.AddSelect(agc.TypeBody, ad2);\n']);
fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
fprintf(fid,['ag.listview.ActivateItem("Preserve Tool Bodies?");\n']);
fprintf(fid,['ag.listview.ItemValue = "Yes";\n']);
fprintf(fid,['agb.Regen();\n']);

fprintf(fid,['var Pat = ag.gui.CreateBoolean();\n']);
fprintf(fid,['ag.listview.ActivateItem("Operation");\n']);
fprintf(fid,['ag.listview.ItemValue = "Imprint Faces";\n']);
fprintf(fid,['ag.listview.ActivateItem("Tool Bodies");\n']);
fprintf(fid,['ag.bodyPick;\n']);
fprintf(fid,['for (var i = 0; i<=',num2str(nprofiles+t-1),'; i++)\n']);
fprintf(fid,['{\n']);
fprintf(fid,['ad1 = ag.fm.Body(i);\n']);
fprintf(fid,['agb.AddSelect(agc.TypeBody, ad1);\n']);
fprintf(fid,['}\n']);
fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
fprintf(fid,['ag.listview.ActivateItem("Target Bodies");\n']);
fprintf(fid,['ag.bodyPick;\n']);
fprintf(fid,['ad2 = ag.fm.Body(',num2str(nprofiles+t),');\n']);
fprintf(fid,['agb.AddSelect(agc.TypeBody, ad2);\n']);
fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
fprintf(fid,['ag.listview.ActivateItem("Preserve Tool Bodies?");\n']);
fprintf(fid,['ag.listview.ItemValue = "Yes";\n']);
fprintf(fid,['agb.Regen();\n']);


fprintf(fid,['var Prt1 = agb.FormNewPartFromAllBodies();\n']);
fprintf(fid,['Prt1.Name ="Part1";\n']);
fprintf(fid,['agb.Regen();\n']);
fclose(fid);
%----------------------------------------------------------------------
function create_model(model_name,load_name,vel_rotation)

tb =load(load_name);
nprofiles=size(tb,1);

fid = fopen(model_name,'w+');

% %malha
fprintf(fid,['var DS = WB.AppletList.Applet("DSApplet").App;\n']);
fprintf(fid,['var ListView = DS.Script.lv;\n']);

fprintf(fid,['var cont = DS.Tree.FirstActiveModel.ContactGroup;\n']);
fprintf(fid,['var parc = cont.Children.Item(1);\n']) ;
fprintf(fid,['DS.Script.changeActiveObject(parc.ID);\n']) ;
fprintf(fid,['DS.Script.Delete();\n']);

fprintf(fid,['var Mesh_Mod = DS.Tree.FirstActiveBranch.MeshControlGroup;\n']);

fprintf(fid,['DS.Script.SelectItems(""+Mesh_Mod.ID);\n']);

fprintf(fid,['DS.Script.doInsertMeshMappedMeshing(1)\n']);
fprintf(fid,['ListView.ActivateItem("Scoping Method");\n']);
fprintf(fid,['ListView.ItemValue = "Named Selection" ;\n']);
fprintf(fid,['ListView.ActivateItem("Named Selection");\n']);
fprintf(fid,['ListView.ItemValue = "SupTotal" ;\n']);


%condicoes de contorno
fprintf(fid,['var Env = DS.Tree.FirstActiveBranch.Environment;\n']);
fprintf(fid,['DS.Script.SelectItems(""+Env.ID);\n']);
for j=1:(nprofiles-1)
    fprintf(fid,['DS.Script.doInsertEnvironmentForce(1)\n']);
    fprintf(fid,['ListView.ActivateItem("Scoping Method");\n']);
    fprintf(fid,['ListView.ItemValue = "Named Selection" ;\n']);
    fprintf(fid,['ListView.ActivateItem("Named Selection");\n']);
    fprintf(fid,['ListView.ItemValue = "Sup',num2str(j),'" ;\n']);
    fprintf(fid,['ListView.ActivateItem("Define By");\n']);
    fprintf(fid,['ListView.ItemValue = "Components" ; \n']);
    fprintf(fid,['ListView.ActivateItem("X Component");\n']);
    fprintf(fid,['ListView.ItemValue = "0" \n']);
    fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
    fprintf(fid,['ListView.ActivateItem("Y Component");\n']);

    carga=num2str(tb(j,1),'%f');
    carga = strsplit(carga,'.');
    carga = strjoin(carga,',');

    fprintf(fid,['ListView.ItemValue = "',carga,'" \n']);
    fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
    fprintf(fid,['ListView.ActivateItem("Z Component");\n']);
    fprintf(fid,['ListView.ItemValue = "0" \n']);
    fprintf(fid,['ListView.SelectedItem.IsChecked="false"\n']);
```

```matlab

    fprintf(fid,['DS.Script.doInsertEnvironmentForce(1)\n']);
    fprintf(fid,['ListView.ActivateItem("Scoping Method");\n']);
    fprintf(fid,['ListView.ItemValue = "Named Selection" ;\n']);
    fprintf(fid,['ListView.ActivateItem("Named Selection");\n']);
    fprintf(fid,['ListView.ItemValue = "Sup',num2str(j),'" ;\n']);
    fprintf(fid,['ListView.ActivateItem("Define By");\n']);
    fprintf(fid,['ListView.ItemValue = "Components" ; \n']);
    fprintf(fid,['ListView.ActivateItem("X Component");\n']);

    carga=num2str(tb(j,2),'%f');
    carga = strsplit(carga,'.');
    carga = strjoin(carga,',');

    fprintf(fid,['ListView.ItemValue = "-',carga,'" \n']);
    fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
    fprintf(fid,['ListView.ActivateItem("Y Component");\n']);
    fprintf(fid,['ListView.ItemValue = "0" \n']);
    fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
    fprintf(fid,['ListView.ActivateItem("Z Component");\n']);
    fprintf(fid,['ListView.ItemValue = "0" \n']);
    fprintf(fid,['ListView.SelectedItem.IsChecked="false"\n']);

    fprintf(fid,['DS.Script.doInsertEnvironmentFreeMoment(1)\n']);
    fprintf(fid,['ListView.ActivateItem("Scoping Method");\n']);
    fprintf(fid,['ListView.ItemValue = "Named Selection" ;\n']);
    fprintf(fid,['ListView.ActivateItem("Named Selection");\n']);
    fprintf(fid,['ListView.ItemValue = "Sup',num2str(j),'" ;\n']);
    fprintf(fid,['ListView.ActivateItem("Define By");\n']);
    fprintf(fid,['ListView.ItemValue = "Components" ; \n']);
    fprintf(fid,['ListView.ActivateItem("X Component");\n']);
    fprintf(fid,['ListView.ItemValue = "0" \n']);
    fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
    fprintf(fid,['ListView.ActivateItem("Y Component");\n']);
    fprintf(fid,['ListView.ItemValue = "0" \n']);
    fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
    fprintf(fid,['ListView.ActivateItem("Z Component");\n']);

    carga=num2str(tb(j,3),'%f');
    carga = strsplit(carga,'.');
    carga = strjoin(carga,',');

    fprintf(fid,['ListView.ItemValue = "',carga,'" \n']);
    fprintf(fid,['ListView.SelectedItem.IsChecked="false"\n']);
end

fprintf(fid,['DS.Script.doInsertEnvironmentFixedDisplacement(1)\n']);
fprintf(fid,['ListView.ActivateItem("Scoping Method");\n']);
fprintf(fid,['ListView.ItemValue = "Named Selection" ;\n']);
fprintf(fid,['ListView.ActivateItem("Named Selection");\n']);
fprintf(fid,['ListView.ItemValue = "Support" ;\n']);

fprintf(fid,['DS.Script.doInsertEnvironmentRotationalVelocity(1)\n']);
fprintf(fid,['ListView.ActivateItem("Define By");\n']);
fprintf(fid,['ListView.ItemValue = "Components" ; \n']);
fprintf(fid,['ListView.ActivateItem("X Component");\n']);
fprintf(fid,['ListView.ItemValue = "0" \n']);
fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
rot=num2str(vel_rotation,'%f');
rot = strsplit(rot,'.');
rot = strjoin(rot,',');
fprintf(fid,['ListView.ActivateItem("Y Component");\n']);
fprintf(fid,['ListView.ItemValue = "-',rot,'" \n']);
fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
fprintf(fid,['ListView.ActivateItem("Z Component");\n']);
fprintf(fid,['ListView.ItemValue = "0" \n']);
fprintf(fid,['ListView.SelectedItem.IsChecked="false"\n']);

fprintf(fid,['DS.Script.doInsertEnvironmentGravity(1)\n']);
fprintf(fid,['ListView.ActivateItem("Direction");\n']);
fprintf(fid,['ListView.ItemValue = "-X Direction" \n']);

fclose(fid);

%-------------------------------------------------------------------
function [Ax,Ay,Ri] = plot_shape3Da(filename,npoints,nprofiles,radius,chord,twist,el)
XiYi = load(filename);

% Ajuste do bordo de fuga
np = 5;ni = 20;
xs = XiYi((length(XiYi(:,1))-np),1);
dxi0 = (max(XiYi(:,1))-xs)/np; xi0 = xs:dxi0:max(XiYi(:,1));
dxi = (max(XiYi(:,1))-xs)/ni; xi = xs:dxi:max(XiYi(:,1));
vecx = XiYi([(length(XiYi(:,1))-np):length(XiYi(:,1))],1)';

ys = XiYi((length(XiYi(:,2))-np),2);
dyi0 = (max(XiYi(:,2))-ys)/np; yi0 = ys:dyi0:max(XiYi(:,2));
dyi = (max(XiYi(:,2))-ys)/ni; yi = ys:dyi:max(XiYi(:,2));
vecy = XiYi([(length(XiYi(:,2))-np):length(XiYi(:,2))],2)';

Xi = spline(xi0,vecx,xi);
Yi = spline(yi0,vecy,yi);

XiYi([(length(XiYi(:,1))-np):length(XiYi(:,1))],:) = [];
XiYi = [XiYi;[Xi',Yi']];

%%%%%%%%%%%%%%%%%%%%%%%%%%

R = radius;
C = chord;
B = twist;  %Radianos

dr = (R(end)-R(1))/(nprofiles-1);
Ri = R(1):dr:R(end);
Ci = spline(R,C,Ri);
Bi = spline(R,B,Ri);
Ri=[(el(:,3)/1000);Ri'];
Ci=Ci';
Bi=[(el(:,4)*pi/180);Bi'];
%---------------------------------------------------------------
nx = length(XiYi(:,1));         nc = length(Ci);
ny = length(XiYi(:,2));         nr = length(Ri);
%---------------------------------------------------------------
minr = min(Ri);
maxr = max(Ri);
%..............................................................
[xm,ym] = baric_perfil(filename);

%..............................................................
%..............................................................
% INTERPOLA AS COORDENADAS DOS PONTOS DO PERFIL EM CADA ESTAÇAO
%--------------------------------------------------------------
xmin = min(XiYi(:,1)-xm);       ymin = min(XiYi(:,2)-ym);
xmax = max(XiYi(:,1)-xm);       ymax = max(XiYi(:,2)-ym);

dx0 = (xmax-xmin)/(nx-1);       x0 = xmin:dx0:xmax;
dy0 = (ymax-ymin)/(ny-1);       y0 = ymin:dy0:ymax;

dx = (xmax-xmin)/(npoints-1);        x = xmin:dx:xmax;
dy = (ymax-ymin)/(npoints-1);        y = ymin:dy:ymax;
```

```
xx = spline(x0,XiYi(:,1)-xm,x);%%%%correction
yy = spline(y0,XiYi(:,2)-ym,y);%%%%correction


%...............................................................
% ESTABELECE AS COORDENADAS DOS PONTOS DO PERFIL EM CADA ESTAÇAO
%---------------------------------------------------------------
theta =linspace(0,2*pi,npoints);
FX = [[(el(:,1)/1000)]*cos(theta);Ci*xx];
FY = [[(el(:,2)/1000)]*sin(theta);Ci*yy];



%.............................................................
% PROVOCA A ROTAÇAO DOS VETORES QUE FORMAM AS ESTAÇOES DA PA
%-------------------------------------------------------------

for i = 1:npoints
    FXr(:,i) = cos(Bi).*FX(:,i) - sin(Bi).*FY(:,i);
    FYr(:,i) = sin(Bi).*FX(:,i) + cos(Bi).*FY(:,i);
end

FXr = 1000*FXr;
FYr = 1000*FYr;
Ri = 1000*Ri;


z = linspace(minr,maxr,nr);

r = sqrt(FXr.*FXr + FYr.*FYr);

theta = atan2(FYr,FXr);

Ax = r.*cos(theta);
Ay = r.*sin(theta);
Az = Ri;



function [xm,ym] = baric_perfil(filename)
%-------------------------------
XiYi = load(filename);
%-------------------------------
nx = length(XiYi(:,1));
ny = length(XiYi(:,2));
%-------------------------------

x = XiYi(:,1);
y = XiYi(:,2);

for i = 1:nx-1
    Ax(i)   = (x(i+1)-x(i))*y(i);
    Ay(i)   = (y(i+1)-y(i))*x(i);
    xA(i)  = (x(i+1)+x(i))/2*Ax(i);
    yA(i)  = (y(i+1)+y(i))/2*Ay(i);
end

xm = sum(xA)/sum(Ax);
ym = sum(yA)/sum(Ay);
```