```
%-------------------------------------------------------------------
%%%% "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');
```

```
 63 fprintf(fid,'{\n');
 64 fprintf(fid,'var planeXY  = agb.GetXYPlane();\n');
 65 fprintf(fid,'var newPlane = agb.PlaneFromPlane(planeXY);\n');
 66 fprintf(fid,'newPlane.Name = Name;\n');
 67 fprintf(fid,'newPlane.AddTransform(agc.XformZOffset, Offset);\n');
 68 fprintf(fid,'agb.regen();\n');
 69 fprintf(fid,'return newPlane;\n');
 70 fprintf(fid,'}\n');
 71
 72 % Planos - Criar
 73 for j=1:(nprofiles+t)
 74     fprintf(fid,['Plane',num2str(j),' = doPlane("Plane',num2str(j),'",',num2str(Ri(j)),');↙
\n']);
 75 end
 76 % Sketch - Funcao
 77 fprintf(fid,'function doSketch(plane,Name,splineX,splineY)\n');
 78 fprintf(fid,'{\n');
 79 fprintf(fid,'p = new Object();\n');
 80 fprintf(fid,'agb.SetActivePlane (plane);\n');
 81 fprintf(fid,'p.Plane  = agb.GetActivePlane();\n');
 82 fprintf(fid,'p.Origin = p.Plane.GetOrigin();\n');
 83 fprintf(fid,'p.XAxis  = p.Plane.GetXAxis();\n');
 84 fprintf(fid,'p.YAxis  = p.Plane.GetYAxis();\n');
 85 fprintf(fid,'p.Sk1 = p.Plane.NewSketch();\n');
 86 fprintf(fid,'p.Sk1.Name = Name;\n');
 87 fprintf(fid,'with (p.Sk1)\n');
 88 fprintf(fid,'{\n');
 89 fprintf(fid,'p.Sp1 = SplineBegin();\n');
 90 fprintf(fid,'with (p.Sp1)\n');
 91 fprintf(fid,'{\n');
 92 fprintf(fid,'SplineFlexibility = agc.Yes;\n');
 93
 94 fprintf(fid,'for (itr in splineX)\n');
 95 fprintf(fid,'{\n');
 96 fprintf(fid,'SplineXY( splineX[itr], splineY[itr]);\n');
 97 fprintf(fid,'}\n');
 98
 99 fprintf(fid,'SplineFitPtEnd();\n');
100 fprintf(fid,'}\n');
101 fprintf(fid,'}\n');
102 fprintf(fid,'p.Plane.EvalDimCons();\n');
103 fprintf(fid,'return p;\n');
104 fprintf(fid,'}\n');
105
106 for j=1:(nprofiles+t)
107     % Sketch - Splines
108     fprintf(fid,['var splineX = new Array(',num2str(npoints),');\n']);
109     fprintf(fid,['var splineY = new Array(',num2str(npoints),');\n']);
110     for s=1:npoints
111         fprintf(fid,['splineX[',num2str(s-1),']=',num2str(FXr(j,s)),';\n']);
112         fprintf(fid,['splineY[',num2str(s-1),']=',num2str(FYr(j,s)),';\n']);
113     end
114     % Sketch - Criar
115     fprintf(fid,['skPlane',num2str(j),' = doSketch(Plane',num2str(j),',"Sketch',num2str↙
(j),'",splineX,splineY',');\n']);
116
117     fprintf(fid,['ag.selectedFeature = ag.gui.TreeviewFeature(p.Sk1.Name, 0);\n']);
118     fprintf(fid,['var SSk1 = ag.gui.CreateSurfSk();\n']);
119     % fprintf(fid,['SSk1.Name = "name";\n']);
120     fprintf(fid,['SSk1.Operation = ag.c.Frozen;\n']);
121     fprintf(fid,['SSk1.WithPlaneNormal = ag.c.Yes;\n']);
122     fprintf(fid,['ag.listview.ActivateItem("Thickness (>=0)");\n']);
123     fprintf(fid,['ag.listview.ItemValue = "0,1";\n']);
124 end
125 fprintf(fid,['function planeSketchesOnly(p)\n']);
```

```
126 fprintf(fid,['{\n']);
127 fprintf(fid,['agb.SetActivePlane (Plane1)\n']);
128 fprintf(fid,['p.Plane  = agb.GetActivePlane();\n']);
129 fprintf(fid,['p.Origin = p.Plane.GetOrigin();\n']);
130 fprintf(fid,['p.XAxis  = p.Plane.GetXAxis();\n']);
131 fprintf(fid,['p.YAxis  = p.Plane.GetYAxis();\n']);
132 fprintf(fid,['p.Sk2 = p.Plane.NewSketch();\n']);
133 fprintf(fid,['p.Sk2.Name ="longarina";\n']);
134 fprintf(fid,['with (p.Sk2)\n']);
135 fprintf(fid,['{\n']);
136 raio=num2str(d1/2,'%f');
137 raio = strsplit(raio,'.');
138 raio = strjoin(raio,',');
139 fprintf(fid,['p.Circ1 = Circle(',num2str(xc),',',num2str(yc),',"',raio,'");\n']);
140 %fprintf(fid,['p.Circ2 = Circle(',num2str(xc),',',num2str(yc),',"',num2str(d2/2),'");↙
\n']);
141 fprintf(fid,['}\n']);
142 fprintf(fid,['p.Sk3 = p.Plane.NewSketch();\n']);
143 fprintf(fid,['p.Sk3.Name ="longarina_remove";\n']);
144 fprintf(fid,['with (p.Sk3)\n']);
145 fprintf(fid,['{\n']);
146 fprintf(fid,['p.Circ1 = Circle(',num2str(xc),',',num2str(yc),',"',raio,'");\n']);
147 fprintf(fid,['}\n']);
148 fprintf(fid,['p.Plane.EvalDimCons();\n']);
149 fprintf(fid,['return p;\n']);
150 fprintf(fid,['}\n']);
151
152 fprintf(fid,['var Skin1 = agb.Skin(agc.Add, agc.Yes, 0.0, 0.0);\n']);
153 fprintf(fid,['Skin1.Name = "Skin";\n']);
154
155 for j=1:(nprofiles+t)
156     fprintf(fid,['Skin1.AddBaseObject(skPlane',num2str(j),'.Sk1);\n']);
157 end
158 fprintf(fid,['agb.Regen();\n']);
159
160 fprintf(fid,['for (var i = 1; i<=',num2str(nprofiles-1),'; i++)\n']);
161 fprintf(fid,['{\n']);
162 fprintf(fid,['var namedSelection = ag.gui.CreateSelectionSet();\n']);
163 fprintf(fid,['ag.gui.Commit();\n']);
164 fprintf(fid,['namedSelection.Name = "Sup" +i\n']);
165 fprintf(fid,['var face = ag.m.ModelFaces(i+',num2str(nprofiles+2*t),');\n']);
166 fprintf(fid,['agb.AddSelect(agc.TypeFace, face);\n']);
167 fprintf(fid,['ag.listview.ActivateItem("Geometry");\n']);
168 fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
169 fprintf(fid,['ag.Regen;\n']);
170 fprintf(fid,['}\n']);
171
172
173 fprintf(fid,['var namedSelection = ag.gui.CreateSelectionSet();\n']);
174 fprintf(fid,['ag.gui.Commit();\n']);
175 fprintf(fid,['namedSelection.Name = "SupTotal";\n']);
176 fprintf(fid,['for (var i = 1; i<=',num2str(t+nprofiles-1),'; i++)\n']);
177 fprintf(fid,['{\n']);
178 fprintf(fid,['var face = ag.m.ModelFaces(i+',num2str(nprofiles+t),');\n']);
179 fprintf(fid,['agb.AddSelect(agc.TypeFace, face);\n']);
180 fprintf(fid,['}\n']);
181 fprintf(fid,['ag.listview.ActivateItem("Geometry");\n']);
182 fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
183 fprintf(fid,['ag.Regen;\n']);
184
185
186 fprintf(fid,['var namedSelection = ag.gui.CreateSelectionSet();\n']);
187 fprintf(fid,['ag.gui.Commit();\n']);
188 fprintf(fid,['namedSelection.Name = "Support";\n']);
189 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']);
```

```
253 fprintf(fid,['ad2 = ag.fm.Body(',num2str(nprofiles+t),');\n']);
254 fprintf(fid,['agb.AddSelect(agc.TypeBody, ad2);\n']);
255 fprintf(fid,['ag.listview.ItemValue = "Apply";\n']);
256 fprintf(fid,['ag.listview.ActivateItem("Preserve Tool Bodies?");\n']);
257 fprintf(fid,['ag.listview.ItemValue = "Yes";\n']);
258 fprintf(fid,['agb.Regen();\n']);
259
260
261 fprintf(fid,['var Prt1 = agb.FormNewPartFromAllBodies();\n']);
262 fprintf(fid,['Prt1.Name ="Part1";\n']);
263 fprintf(fid,['agb.Regen();\n']);
264 fclose(fid);
265 %--------------------------------------------------------------------
266 function create_model(model_name,load_name,vel_rotation)
267
268 tb =load(load_name);
269 nprofiles=size(tb,1);
270
271 fid = fopen(model_name,'w+');
272
273 % %malha
274 fprintf(fid,['var DS = WB.AppletList.Applet("DSApplet").App;\n']);
275 fprintf(fid,['var ListView = DS.Script.lv;\n']);
276
277 fprintf(fid,['var cont = DS.Tree.FirstActiveModel.ContactGroup;\n']);
278 fprintf(fid,['var parc = cont.Children.Item(1);\n']) ;
279 fprintf(fid,['DS.Script.changeActiveObject(parc.ID);\n']) ;
280 fprintf(fid,['DS.Script.Delete();\n']);
281
282 fprintf(fid,['var Mesh_Mod = DS.Tree.FirstActiveBranch.MeshControlGroup;\n']);
283
284 fprintf(fid,['DS.Script.SelectItems(""+Mesh_Mod.ID);\n']);
285
286 fprintf(fid,['DS.Script.doInsertMeshMappedMeshing(1)\n']);
287 fprintf(fid,['ListView.ActivateItem("Scoping Method");\n']);
288 fprintf(fid,['ListView.ItemValue = "Named Selection" ;\n']);
289 fprintf(fid,['ListView.ActivateItem("Named Selection");\n']);
290 fprintf(fid,['ListView.ItemValue = "SupTotal" ;\n']);
291
292
293 %condicoes de contorno
294 fprintf(fid,['var Env = DS.Tree.FirstActiveBranch.Environment;\n']);
295 fprintf(fid,['DS.Script.SelectItems(""+Env.ID);\n']);
296 for j=1:(nprofiles-1)
297     fprintf(fid,['DS.Script.doInsertEnvironmentForce(1)\n']);
298     fprintf(fid,['ListView.ActivateItem("Scoping Method");\n']);
299     fprintf(fid,['ListView.ItemValue = "Named Selection" ;\n']);
300     fprintf(fid,['ListView.ActivateItem("Named Selection");\n']);
301     fprintf(fid,['ListView.ItemValue = "Sup',num2str(j),'" ;\n']);
302     fprintf(fid,['ListView.ActivateItem("Define By");\n']);
303     fprintf(fid,['ListView.ItemValue = "Components" ; \n']);
304     fprintf(fid,['ListView.ActivateItem("X Component");\n']);
305     fprintf(fid,['ListView.ItemValue = "0" \n']);
306     fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
307     fprintf(fid,['ListView.ActivateItem("Y Component");\n']);
308
309     carga=num2str(tb(j,1),'%f');
310     carga = strsplit(carga,'.');
311     carga = strjoin(carga,',');
312
313     fprintf(fid,['ListView.ItemValue = "',carga,'" \n']);
314     fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
315     fprintf(fid,['ListView.ActivateItem("Z Component");\n']);
316     fprintf(fid,['ListView.ItemValue = "0" \n']);
317     fprintf(fid,['ListView.SelectedItem.IsChecked="false"\n']);
```

```
318
319     fprintf(fid,['DS.Script.doInsertEnvironmentForce(1)\n']);
320     fprintf(fid,['ListView.ActivateItem("Scoping Method");\n']);
321     fprintf(fid,['ListView.ItemValue = "Named Selection" ;\n']);
322     fprintf(fid,['ListView.ActivateItem("Named Selection");\n']);
323     fprintf(fid,['ListView.ItemValue = "Sup',num2str(j),'" ;\n']);
324     fprintf(fid,['ListView.ActivateItem("Define By");\n']);
325     fprintf(fid,['ListView.ItemValue = "Components" ; \n']);
326     fprintf(fid,['ListView.ActivateItem("X Component");\n']);
327
328     carga=num2str(tb(j,2),'%f');
329     carga = strsplit(carga,'.');
330     carga = strjoin(carga,',');
331
332     fprintf(fid,['ListView.ItemValue = "-',carga,'" \n']);
333     fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
334     fprintf(fid,['ListView.ActivateItem("Y Component");\n']);
335     fprintf(fid,['ListView.ItemValue = "0" \n']);
336     fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
337     fprintf(fid,['ListView.ActivateItem("Z Component");\n']);
338     fprintf(fid,['ListView.ItemValue = "0" \n']);
339     fprintf(fid,['ListView.SelectedItem.IsChecked="false"\n']);
340
341     fprintf(fid,['DS.Script.doInsertEnvironmentFreeMoment(1)\n']);
342     fprintf(fid,['ListView.ActivateItem("Scoping Method");\n']);
343     fprintf(fid,['ListView.ItemValue = "Named Selection" ;\n']);
344     fprintf(fid,['ListView.ActivateItem("Named Selection");\n']);
345     fprintf(fid,['ListView.ItemValue = "Sup',num2str(j),'" ;\n']);
346     fprintf(fid,['ListView.ActivateItem("Define By");\n']);
347     fprintf(fid,['ListView.ItemValue = "Components" ; \n']);
348     fprintf(fid,['ListView.ActivateItem("X Component");\n']);
349     fprintf(fid,['ListView.ItemValue = "0" \n']);
350     fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
351     fprintf(fid,['ListView.ActivateItem("Y Component");\n']);
352     fprintf(fid,['ListView.ItemValue = "0" \n']);
353     fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
354     fprintf(fid,['ListView.ActivateItem("Z Component");\n']);
355
356     carga=num2str(tb(j,3),'%f');
357     carga = strsplit(carga,'.');
358     carga = strjoin(carga,',');
359
360     fprintf(fid,['ListView.ItemValue = "',carga,'" \n']);
361     fprintf(fid,['ListView.SelectedItem.IsChecked="false"\n']);
362 end
363
364 fprintf(fid,['DS.Script.doInsertEnvironmentFixedDisplacement(1)\n']);
365 fprintf(fid,['ListView.ActivateItem("Scoping Method");\n']);
366 fprintf(fid,['ListView.ItemValue = "Named Selection" ;\n']);
367 fprintf(fid,['ListView.ActivateItem("Named Selection");\n']);
368 fprintf(fid,['ListView.ItemValue = "Support" ;\n']);
369
370 fprintf(fid,['DS.Script.doInsertEnvironmentRotationalVelocity(1)\n']);
371 fprintf(fid,['ListView.ActivateItem("Define By");\n']);
372 fprintf(fid,['ListView.ItemValue = "Components" ; \n']);
373 fprintf(fid,['ListView.ActivateItem("X Component");\n']);
374 fprintf(fid,['ListView.ItemValue = "0" \n']);
375 fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
376 rot=num2str(vel_rotation,'%f');
377 rot = strsplit(rot,'.');
378 rot = strjoin(rot,',');
379 fprintf(fid,['ListView.ActivateItem("Y Component");\n']);
380 fprintf(fid,['ListView.ItemValue = "-',rot,'" \n']);
381 fprintf(fid,['ListView.SelectedItem.IsChecked="false" \n']);
382 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;
```

```
448
449 xx = spline(x0,XiYi(:,1)-xm,x);%%%%correction
450 yy = spline(y0,XiYi(:,2)-ym,y);%%%%correction
451
452
453 %.............................................................
454 % ESTABELECE AS COORDENADAS DOS PONTOS DO PERFIL EM CADA ESTAÇAO
455 %-------------------------------------------------------------
456 theta =linspace(0,2*pi,npoints);
457 FX = [[(el(:,1)/1000)]*cos(theta);Ci*xx];
458 FY = [[(el(:,2)/1000)]*sin(theta);Ci*yy];
459
460
461
462 %.............................................................
463 % PROVOCA A ROTAÇAO DOS VETORES QUE FORMAM AS ESTAÇOES DA PA
464 %-------------------------------------------------------------
465
466 for i = 1:npoints
467     FXr(:,i) = cos(Bi).*FX(:,i) - sin(Bi).*FY(:,i);
468     FYr(:,i) = sin(Bi).*FX(:,i) + cos(Bi).*FY(:,i);
469 end
470
471 FXr = 1000*FXr;
472 FYr = 1000*FYr;
473 Ri = 1000*Ri;
474
475
476 z = linspace(minr,maxr,nr);
477
478 r = sqrt(FXr.*FXr + FYr.*FYr);
479
480 theta = atan2(FYr,FXr);
481
482 Ax = r.*cos(theta);
483 Ay = r.*sin(theta);
484 Az = Ri;
485
486
487
488 function [xm,ym] = baric_perfil(filename)
489 %-------------------------------
490 XiYi = load(filename);
491 %-------------------------------
492 nx = length(XiYi(:,1));
493 ny = length(XiYi(:,2));
494 %-------------------------------
495
496 x = XiYi(:,1);
497 y = XiYi(:,2);
498
499 for i = 1:nx-1
500     Ax(i)  = (x(i+1)-x(i))*y(i);
501     Ay(i)  = (y(i+1)-y(i))*x(i);
502     xA(i) = (x(i+1)+x(i))/2*Ax(i);
503     yA(i) = (y(i+1)+y(i))/2*Ay(i);
504 end
505
506 xm = sum(xA)/sum(Ax);
507 ym = sum(yA)/sum(Ay);
```