/PREP7

***GET**, Tempo_0, ACTIVE, 0, *TIME*, *WALL*,
***AFUN**,DEG !unidade de ângulos

!*ASK,p_tol,espessura mínima de parede,0.002
kp = 1 !Contador de Kpts
k_l = 0 !Contador de Linhas
Area_count = 0 !Contador de Areas
***DIM**,Blade_info,ARRAY,15,4

Blade_info(1,1) =
0.15,0.218,0.286,0.354,0.421,0.489,0.557,0.625,0.693,0.761,0.829,0.896,0.964,1.032,1.1
Blade_info(1,2) = 0.29,0.38,0.44,0.49,0.52,0.53,0.54,0.53,0.5,0.47,0.43,0.38,0.31,0.22,0.1
Blade_info(1,3) =
43.69,40.68,37.86,35.22,32.73,30.41,28.23,26.19,24.29,22.52,20.87,19.33,17.9,16.57,15.34
Blade_info(1,4) =
0.102,0.134,0.157,0.174,0.185,0.190,0.190,0.187,0.179,0.168,0.154,0.135,0.112,0.08,0.033

***DIM**,Area_info,ARRAY,5,2,15 ! Matriz de propriedades das áreas que formam a superfície do !sólido
***DIM**,Blade_points_X,ARRAY,101,3,15
***DIM**,Blade_points_Y,ARRAY,101,3,15
***DIM**,Blade_points_Z,ARRAY,101,3,15
***DIM**,Blade_points_N,ARRAY,101,3,15
***DIM**,Blade_lines_N,ARRAY,1,3,15

***VEC,**raiolim,D,ALLOC,4 !**Criando vetor dos raios limites da esfera a ser criada**
***VEC,**vecrprev,D,ALLOC,1

stop = 1
***DOWHILE**,stop

/INQUIRE,file_valid,EXIST,txt_node_data,txt,,
***IF**,file_valid,EQ,0,THEN
! Do-loop para gerar o solido por loft ou skinning, cria-se o sketch de cada sessão e então ! !
! aplica-se o skinning para criar a área lateral

***DO**,i,1,15
/VIEW,1,1,1,1

```
/ANG,1
/REP,FAST
!Perfil NACA 4415
Corda = Blade_info(i,2)
Area_info(3,1,i) =  Blade_info(i,2)
Area_info(3,2,i) =  Blade_info(i,2)
m = (4/100)
p = (4/10)
t = (15/100)

x_c = 0
x_cinc = 0.01

z_coord = Blade_info(i,1)
Area_info(1,1,i) = Blade_info(i,1)
Area_info(1,2,i) = Blade_info(i,1)
teta_blade = Blade_info(i,3)
dist_cent = Blade_info(i,4)
x_desloc = -corda*0.33
y_desloc = 0

*DO,j,1,101

*IF,x_c,GE,0,AND,x_c,LE,p,THEN
y_c = (m/p**2)*(2*p*x_c-x_c**2)
teta_surf = ATAN( (2*m*(p-x_c))/(p**2))
Area_info(4,1,i) = teta_surf
Area_info(4,2,i) = teta_surf

*ELSE
y_c = (m/(1-p)**2)*(1-(2*p) + 2*p*x_c - x_c**2)
teta_surf = ATAN(((2*m)/((1-p)**2))*(p-x_c))
Area_info(4,1,i) = teta_surf
Area_info(4,2,i) = teta_surf

*ENDIF

y_t = (5*t)*(0.2969*(x_c**(1/2))-0.126*x_c-0.3516*(x_c**2)+
0.2843*(x_c**3)-0.1015*(x_c**4))
```

```
Area_info(5,1,i) = y_t
Area_info(5,2,i) = y_t

x_u = x_c - y_t*sin(teta_surf)
y_u = y_c + y_t*cos(teta_surf)

x_l = x_c + y_t*sin(teta_surf)
y_l = y_c - y_t*cos(teta_surf)

x_cs = (x_c*corda + x_desloc)
y_cs = (y_c*corda + y_desloc)
x_cr = x_cs*cos(teta_blade) - y_cs*sin(teta_blade)
y_cr = x_cs*sin(teta_blade) + y_cs*cos(teta_blade)

x_us = (x_u*corda + x_desloc)
y_us = (y_u*corda + y_desloc)
x_ur = x_us*cos(teta_blade) - y_us*sin(teta_blade)
y_ur = x_us*sin(teta_blade) + y_us*cos(teta_blade)

x_ls = ( x_l*corda + x_desloc)
y_ls = (y_l*corda + y_desloc)
x_lr = x_ls*cos(teta_blade) - y_ls*sin(teta_blade)
y_lr = x_ls*sin(teta_blade) + y_ls*cos(teta_blade)

*IF,x_c,EQ,1,THEN
x_ur = x_cr
y_ur = y_cr
x_lr = x_cr
y_lr = y_cr
*ENDIF
!KPLOT
K,kp,x_cr,y_cr,z_coord
Blade_points_N(j,2,i) = kp
Blade_points_X(j,2,i) = x_cr
Blade_points_Y(j,2,i) = y_cr
Blade_points_Z(j,2,i) = z_coord
kp = kp +1

K,kp,x_ur,y_ur,z_coord
```

```
Blade_points_N(j,1,i) = kp
Blade_points_X(j,1,i) = x_ur
Blade_points_Y(j,1,i) = y_ur
Blade_points_Z(j,1,i) = z_coord
kp = kp + 1

K,kp,x_lr,y_lr,z_coord
Blade_points_N(j,3,i) = kp
Blade_points_X(j,3,i) = x_lr
Blade_points_Y(j,3,i) = y_lr
Blade_points_Z(j,3,i) = z_coord

kp = kp + 1
*IF,x_c,EQ,0,THEN
Blade_points_N(j,1,i) = Blade_points_N(j,2,i)
Blade_points_N(j,3,i) = Blade_points_N(j,2,i)
*ELSEIF,x_c,EQ,1,THEN
Blade_points_N(j,1,i) = Blade_points_N(j,2,i)
Blade_points_N(j,3,i) = Blade_points_N(j,2,i)
*ENDIF
x_c = x_c + x_cinc


*ENDDO

*ENDDO


*DO,i,1,15
j = 1
splines_up = 0
splines_c = 0
splines_low = 0
spline_done = 1
FLST,3,101,3
*DO,j,1,101
FITEM,3,Blade_points_N(j,1,i)
*ENDDO
BSPLIS, ,P51X
```

```
k_l =k_l + 1
Blade_lines_N(1,1,i) = k_l

!FLST,3,101,3
!*DO,j,1,101
!FITEM,3,Blade_points_N(j,2,i)
!*ENDDO
!BSPLIS, ,P51X
!k_l  = k_l + 1
!Blade_lines_N(1,2,i) = k_l

FLST,3,101,3
*DO,j,1,101
FITEM,3,Blade_points_N(j,3,i)
*ENDDO
BSPLIS, ,P51X
k_l  = k_l  + 1
Blade_lines_N(1,3,i) = k_l
KPLOT
LPLOT
*ENDDO

*DO,i,1,14
*IF,i,EQ,1,THEN
AL,Blade_lines_N(1,1,i),Blade_lines_N(1,3,i)
Area_count = Area_count + 1
*ENDIF
ASKIN,Blade_lines_N(1,1,i), Blade_lines_N(1,1,i+1)
Area_count = Area_count + 1
Area_info(2,1,i) = Area_count
ASKIN,Blade_lines_N(1,3,i), Blade_lines_N(1,3,i+1)
Area_count =Area_count+ 1
Area_info(2,2,i) = Area_count

*IF,i+1,EQ,15,THEN
AL,Blade_lines_N(1,1,i+1),Blade_lines_N(1,3,i+1)
Area_count = Area_count + 1
*EXIT
*ENDIF
```

!KPLOT

!APLOT

***ENDDO**

Area_info(2,1,15) = Area_info(2,1,14)

Area_info(2,2,15) = Area_info(2,2,14)

VA,ALL !Criando Sólido a partir das áreas criadas no loop de desenho

VPLOT !Plotando o volume criado*/*

***GET, Tempo_1, ACTIVE, 0, *TIME*, *WALL*,**

T_desenho_pa = Tempo_1 - Tempo_0

!*ASK,Confirme,tirou foto?

!*GET,vol_ini,VOLU,0,NUM,MAXD

!VSEL,S,VOLU,,vol_ini,,,1

!IGESOUT,Vol_ini,IGES,,0

ET,1,solid285 !Descrevendo tipo de elemento utilizado Solid285 (elemento 3D 4 nós, tetrae)

MP,ex,1,2e11 !Definindo Módulo de elasticidade

MP,prxy,1,0.3 !Definindo Razão de Poisson

save

!

ESIZE,0.1,

!*ASK,Confirm,gerar malha?,

VMESH,1,

Elem_size = 0.0173

***GET, Tempo_2, ACTIVE, 0, *TIME*, *WALL*,**

T_malha = Tempo_2 - Tempo_1

!----------

! DISCRETIZANDO DOMÍNIO

!----------

E _0= 2E11

Poiss = 0.3

ET, *2,SOLID285*

ET, *3,SOLID285*

MP, *EX*, *1*, *E_0* **!Definindo Módulo de elasticidade**

MP, *PRXY*, *1*,*Poiss* **!Definindo Razão de Poisson**

```
MP, EX, 2, E_0  !Definindo Módulo de elasticidade
MP, PRXY, 2,Poiss  !Definindo Razão de Poisson

MP, EX, 3, E_0  !Definindo Módulo de elasticidade
MP, PRXY, 3,Poiss  !Definindo Razão de Poisson

save

ESEL, ALL, !Selecionando todos os elementos criados

*GET, Num_elem, ELEM, 0, NUM, MAX,  !Obtendo número total de elementos e salvando na !variável Num_elem

*GET, Num_node, NODE, 0, NUM, MAX,

!Definindo superfície externa como restrição para a criação de esferas
ASEL, ALL
NSLA, S, 1
ESLN, S, 0, ALL
EMODIF, ALL, TYPE, 2

!PATH, restric, 303,30, 50
!Definindo estruturas internas fictícias como restrição para criação de esferas
!/UIS, MSGPOP, 3
!ESEL, NONE,
!*DO,i,1,5
!pp = 1
!*DO,j,1,101
!*DO,k,1,3
!stripe = i*3
        !x_path = Blade_points_X(j,k,stripe)
        !y_path = Blade_points_Y(j,k,stripe)
        !z_path = Blade_points_Z(j,k,stripe)
        ! PPATH, pp, , x_path, y_path, z_path,1

 !       pp = pp + 1
!!*ENDDO
!*ENDDO
```

```
!ESEL,A, path,restric,
!*ENDDO
!EMODIF, ALL, TYPE, 3


VSEL, S, VOLU,  , 1,
NSLV, S, 1
ASEL, ALL
NSLA, U, 1
!NSLE, U,

*GET, Num_node_in, NODE, 0, COUNT,
*DIM, Node_data, ARRAY, Num_node_in, 5
*DIM, cav_center, ARRAY, Num_node_in, 5

ESEL, S, TYPE,,2
!ESEL, A, TYPE,,3
*GET, Num_e_surf, ELEM, 0, COUNT,

node_n = 0
*DO,i,1,Num_node_in

 node_num = NDNEXT(node_n)
Node_data(i,1) = node_num
Node_data(i,2) = NX(node_num)
Node_data(i,3) = NY(node_num)
Node_data(i,4) = NZ(node_num)
node_n = node_num
 *ENDDO


!Definindo pontos para o caminho de seleção de elementos da vizinhança
ang_phi(1:21) = 0
*DO, i, 2,21,
ang_phi(i) = ang_phi(i-1) + 18
*ENDDO
ang_teta(1:3) = 0,120,240
npts_ppath = 21*3
```

/UIS, ***MSGPOP***, *3* **!suprimindo avisos**

PATH, *ngbr*, *npts_ppath*,*30*, *50* **!Alocando espaço para o caminho de seleção de elementos da !vizinhança**

***DO**, *N*, *1*,*Num_node_in*

locX_Ni = Node_data(N,2)

locY_Ni = Node_data(N,3)

locZ_Ni = Node_data(N,4)

R_viz = (Elem_size)/2

R_viz_it = (Elem_size)/2

ESEL, *S*, *TYPE*,,2

!ESEL, *A*, *TYPE*,,3

r_stop = 1

total_path = 1

DOWHILE**, ***r_stop

k = 1

***DO**, *phii*, *1*,*21*,

***DO**, *tetaa*, *1*,*3*,

*x_path = locX_Ni + R_viz_it*cos(ang_phi(phii))*sin(ang_teta(tetaa))*

*y_path = locY_Ni + R_viz_it*sin(ang_phi(phii))*

*z_path = locZ_Ni + R_viz_it*cos(ang_phi(phii))*cos(ang_teta(tetaa))*

PPATH, *k*, , *x_path, y_path, z_path,1*

k = k+1

***ENDDO**

***ENDDO**

```
ESEL,U, path,ngbr,
*GET, path_check, ELEM, 0, COUNT,

*IF,path_check, LT,Num_e_surf,THEN
 R_viz_it = R_viz_it - R_viz*0.5
Node_data(N,5) = R_viz_it
r_stop = 0
*EXIT
*ELSE
R_viz_it = R_viz_it + R_viz*0.5
total_path = total_path+1
 *ENDIF

*ENDDO
*ENDDO


*GET, Tempo_3, ACTIVE, 0, TIME, WALL,
        T_espaco_no = Tempo_3 - Tempo_2


*MFUN, ND_RSORT, COPY, NODE_DATA
*MOPER, PER_ND, ND_RSORT, SORT,,5

Space_Check = 1
cav_count = 0
node_disp = 1
*DOWHILE, SPACE_CHECK

i_a = ND_RSORT(Num_node_in,1)
X_a = ND_RSORT(Num_node_in,2)
Y_a = ND_RSORT(Num_node_in,3)
Z_a = ND_RSORT(Num_node_in,4)
R_max_a = ND_RSORT(Num_node_in,5)
Num_node_in2 = 0
*DO, i, node_disp,Num_node_in
```

```
DIST_ai = ((X_a - Node_data(i,2))**(2) + (Y_a - Node_data(i,3))**(2)
+ (Z_a - Node_data(i,4))**(2))**(1/2)
print_wall_tol = DIST_ai - R_max_a
*IF,print_wall_tol,LE,0.002,THEN
Node_data(i,1) = 0
Node_data(i,2) = 0
Node_data(i,3) = 0
Node_data(i,4) = 0
Node_data(i,5) = 0
Num_node_in2 = Num_node_in2 + 1

*ELSE
R_max = Node_data(i,5)
R_max_new = DIST_ai - R_max_a - 0.0006
*IF,R_max_new,LE,R_max,THEN
Node_data(i,5) = R_max_new
*ENDIF
*ENDIF

*ENDDO

*MOPER, PER_ND1, Node_data, SORT,,1
*MFUN, ND_RSORT, COPY, NODE_DATA
*MOPER, PER_ND, ND_RSORT, SORT,,5
node_disp = node_disp + Num_node_in2


cav_count = cav_count+1
cav_center(cav_count,1) = i_a
cav_center(cav_count,2) = X_a
cav_center(cav_count,3) = Y_a
cav_center(cav_count,4) = Z_a
cav_center(cav_count,5) = R_max_a
```

```
/COM, num no:%i_a% R_max: %R_max_a% cavidades = %cav_count%

*IF,node_disp, GE,Num_node_in,THEN
*EXIT
*ENDIF

*IF,R_max_a, LE,0.001,THEN
*EXIT
*ENDIF

*ENDDO
*GET, Tempo_4, ACTIVE, 0, TIME, WALL,
        T_cav_def = Tempo_4 - Tempo_3

*DO, i, 1,cav_count

locX_Ni = cav_center(i,2)

locY_Ni = cav_center(i,3)
locZ_Ni = cav_center(i,4)
R_viz_it = cav_center(i,5)

WPLANE,-1,locX_Ni,locY_Ni,locZ_Ni,locX_Ni+1,locY_Ni,locZ_Ni,locX_Ni,locY_N
i+1,locZ_Ni
SPHERE,R_viz_it !Esfera de medição criada para interceptar a superfície externa
*ENDDO

ASEL,ALL
VSEL,S,VOLU,,1
ASLV,U
APLOT,ALL
*GET, Tempo_5, ACTIVE, 0, TIME, WALL,
        T_cav_creat = Tempo_5 - Tempo_4

/post1
```

```
ETABLE, Volume,VOLU,
*VGET, Elem_vol, ELEM, 1, ETAB, Volume, , ,0
*VSCFUN, vol_total, SUM, Elem_vol

total_sph_vol = 0

*DO, i, 1,cav_count

R_viz_it = cav_center(i,5)
Sphere_vol = (4/3)*3.1415*(R_viz_it**3)
total_sph_vol = total_sph_vol + Sphere_vol
*ENDDO
```