```
!-----------------------------------------------------------------------------------------------------------------
!                              CRIANDO DOMÍNIO DE ANÁLISE (BLOCO)
!-----------------------------------------------------------------------------------------------------------------
/PREP7
*AFUN, DEG
!Dimensões do bloco

L_block = 10 ! Lado [m]
x1 = 0
y1 = 0
z1 = 0
x2 = L_block
y2 =( L_block/10)*3
z2 = L_block/10

BLOCK, X1, X2, Y1, Y2, Z1, Z2  !Criando bloco

!-----------------------------------------------------------------------------------------------------------------
!                       DISCRETIZANDO DOMÍNIO + PROPRIEDADES DOS MATERIAIS
!-----------------------------------------------------------------------------------------------------------------
E _0 =  1
E_min = 1E-13
Poiss = 0.3
mass_dens = 7850
```

S_ypla = 24.8E6

S_ult = 46E6

ET, *1,SOLID185* **!Descrevendo tipo de elemento utilizado, Solid185 (elemento 3D 8nós, hexaedro)**

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

MP, *DENS*, *1*, mass_dens **!Definindo massa específica do elemento**

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

save

Elem_size = L_block/10

ESIZE,*Elem_size* **!Definindo tamanho dos elemento pelo número de divisões nas !arestas do !bloco**

***ASK**, *Confirma*, *Gerar malha?,* **!Checkpoint**

VMESH, *1,* **!Gerar malha no volume 1**

!----------

! CRIANDO VARIÁVEL DE DESENHO

!----------

***ASK**, *obj_massa*, *redução de massa,0.5* **!Definindo redução de massa alvo da análise**

!Tolerância para convergência da análise

***ASK**, *tol_conv*, *tolerância para ser considerada convergência,0.01*

!Fator de penalização

***ASK**, *pnalz*, *fator de penalização para o SIMP,3*

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*,

***DIM**, *rho*, *ARRAY*, *Num_elem* **!Dimensionando vetor da variável de desenho**
***DIM**, *rho_new*, *ARRAY*, *Num_elem*
***DIM**, *rho_f*, *ARRAY*, *Num_elem*
!*VFILL,*rho_f*, *Rand*, *0*, *1*
!rho_f(1:Num_elem) =
0.6820,0.6539,0.5465,0.5424,0.6347,0.3414,0.3277,0.4511,0.5503,0.4074,0.7261,0.6653,0.5494,0.3287,0.0942
***DO**, *i*, *1*, *Num_elem*, **!Atribuindo valor inicial para as variáveis de desenho**
rho_f(i) = obj_massa
!rho_f(i) = 1
rho(i) = rho_f(i)
***ENDDO**

!----------
! PREPARANDO FILTRO DE MASSA ESPECÍFICA
!----------

***ASK**, *R_V*, *Raio da vizinhança (vezes o tamanho do elemento), 1*

R_viz = (Elem_size)(R_v)* **!definindo raio que caracteriza a vizinhança de cada elemento**

*R_factor = 1.5*Elem_size*

***GET, Time_ini_viz, ACTIVE, 0, *TIME*, *WALL*,**

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

viz_alloc = 15

DIM**, *vizin*, *ARRAY*, *Num_elem*, ***viz_alloc **!dimensionando vetor da vizinhança**

***DIM**, *viz_e_size*, *ARRAY*, *Num_elem*,

DIM**, *H_e*, *ARRAY*, *Num_elem*,viz_alloc***

DIM**, *H_eAR*, *ARRAY*, ***viz_alloc

***DIM**, *H_eS*, *ARRAY*, *Num_elem*,

***DIM**, *H_eMax*, *ARRAY*, *Num_elem*,

***DIM**, *B_ETRACK*, *ARRAY*, *200*,Num_elem

****VGET,*** *locX_ei*, ***ELEM, 1, CENT, X, , ,0***

****VGET,*** *locY_ei*, ***ELEM, 1, CENT, Y, , ,0***

****VGET,*** *locZ_ei*, ***ELEM, 1, CENT, Z, , ,0***

!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

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

!loop para determinação da vizinhança
elem_mesh = 1
*DO, e_N, 1,Num_elem

!Gerando caminho de seleção
k = 1
*DO, phii, 1,21,
*DO, tetaa, 1,3,

x_path = locX_ei(e_N) + R_viz*cos(ang_phi(phii))*sin(ang_teta(tetaa))
y_path = locY_ei(e_N) + R_viz*sin(ang_phi(phii))
z_path = locZ_ei(e_N) + R_viz*cos(ang_phi(phii))*cos(ang_teta(tetaa))

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

k = k+1

***ENDDO**

***ENDDO**

!/PBC,PATH,1

!/REPLOT

!/PBC,PATH,0

!Selecionando elementos atravessados pelo caminho

ESEL, ***S***, ***path***,*ngbr*,

!EPLOT

!*ASK, *Confirma*, *Path ok?,*

***GET**,*Num_E_viz,ELEM,0,COUNT* **!Contando elementos selecionados pelo caminho**

viz_e_size(e_n) = Num_E_viz

!Salvando os números (rótulos) dos elementos

Sel_Elem = 0 **!contador para iniciar a seleção**

***DO**,*i,1,Num_E_viz*

```
 Elem_num = ELNEXT(Sel_Elem)
        Vizin(e_N,i) = Elem_num
        Dist_ij = ((CENTRX(Elem_num)-locX_ei(e_N))**2 + (CENTRY(Elem_num)-locY_ei(e_N))**2 +
(CENTRZ(Elem_num)-locZ_ei(e_N))**2)**(1/2)

        H_e(e_N,i) = R_factor - Dist_ij
        H_check = H_e(e_N,i)
*IF,H_check,LT,0,THEN
        H_e(e_N,i) = 0
*ENDIF
        H_eAR(i) = H_e(e_N,i)

*IF,i,EQ,Num_E_viz,THEN
i = i+1
*DO,cln,i,viz_alloc
        H_eAR(cln) = 0
*ENDDO
*ENDIF

        Sel_Elem = Elem_num

*ENDDO
```

*VSCFUN, HeS, SUM, H_eAR

*VSCFUN, HeMax, Max, H_eAR

H_eS(e_N) = HeS

H_eMax(e_N) = HeMax

*ENDDO

*GET, Time_end_viz, ACTIVE, 0, TIME, WALL,

Time_viz = Time_end_viz - Time_ini_viz

/COM,Tempo de criação da vizinhança: %Time_viz%

!---

! CARREGAMENTOS E FIXAÇÕES

!---

yd_max = 0.05*L_Block

PATH, cargas, 2,30, 50

PPATH, 1, ,L_block, 0, 0,1

PPATH, 2, ,L_block, 0,z2,1

ESEL, S, path,cargas,

NSLE, S,POS,2

NSLE,A,POS,6

!NSEL, S, LOC, X, L_block

F, ALL,FY, -1

NSEL, *S*, *LOC*, *X*, *0*

D, *ALL,ALL,0*

!----------

! ANÁLISE DE OTIMIZAÇÃO (LOOP EXTERNO)

!----------

***DIM**, *rho_it*, *ARRAY*, *200*,Num_elem

***DIM**, *e_ene_it*, *ARRAY*, *200*,Num_elem

***DIM**, *Be_it*, *ARRAY*, *200*,Num_elem

total_ene(1:200) = 0

it_sum_evol(1:200) = 0

delem_ene(1:Num_elem) = 0

rho_diff(1:Num_elem) = 0

delem_vol(1:Num_elem) = 0

sens_R(1:Num_elem) = 0

sens_Rho(1:Num_elem) = 0

obj_vol(1:Num_elem) = 0

sens_V(1:Num_elem) = 0

sum_hrv(1:Num_elem) = 0

sum_hv(1:Num_elem) = 0

e_volm(1:Num_elem) = 0

B_e(1:Num_elem) = 0

n_damping = 0.5

ene_node(1:Num_node) = 0

rho_lowlim(1:2) = 0

rho_uplim(1:2) = 0

rho_min = 1e-2

mass_densmin = 1e-4

STOP_OTIM = 1

J = 1

loop = 1

max_loop = 100

change = 0

Check_ene = 0

DOWHILE**, ***STOP_OTIM

```
ESEL, ALL

/SOLU  !Entrando no ambiente de solução
NCNV, 1, 1E20
!OUTRES, VENG
!TRNOPT,  ,  ,  ,  ,  ,  ,  , YES

!Atribuindo variável de desenho aos elementos da malha
*DO,i,1,Num_elem

E_i = E_min + (E_0 - E_min)*(rho(i)**(pnalz))
mass_dens = mass_densmin + (rho(i)**(pnalz))*(mass_dens - mass_densmin)
MP, EX, i, E_i  !Definindo Módulo de elasticidade
MP, DENS, i, mass_dens !Definindo massa específica do elemento
MP, PRXY, i,Poiss
MPCHG, i, i

*ENDDO
*VSCFUN, rho_max, MAX, rho
rho_check = rho_max

ALLSEL,ALL
SOLVE
```

```
/POST1
!/COM,Iteracao numero : %loop% ultima mudanca de :%change%

ETABLE, Energy,SEND,ELASTIC
*VGET, Elem_ene, ELEM, 1, ETAB, Energy, , ,0

ETABLE, Volume,VOLU,
*VGET, Elem_vol, ELEM, 1, ETAB, Volume, , ,0

*VSCFUN, sum_ene, SUM, Elem_ene,
*VSCFUN, sum_vol, SUM, Elem_vol
total_ene(j) = sum_ene
*IF, LOOP,EQ,1,THEN
SET,FIRST
!*
/EFACET,1
PLNSOL, SEND,ELASTIC, 0,1.0
*ASK, Confirma, tirou foto?,
*ENDIF

*DO,i,1,Num_elem
v_size = viz_e_size(i)
*DO,e_v,1,v_size
ele_num = Vizin(i,e_v)
sens_R(e_v) = (H_e(i,e_v)/H_eS(e_v))
```

```
sens_V(e_v) = (H_e(i,e_v)/H_eS(e_v))

*IF,e_v,EQ,v_size,THEN
e_v =e_v +1
*DO,cln,e_v,viz_alloc
        sens_R(cln) = 0
        sens_V(cln) = 0
*ENDDO
*ENDIF

*ENDDO

*VSCFUN, sum_sens_R, SUM, Sens_R
*VSCFUN, sum_sens_V, SUM, Sens_V
delem_ene(i) = sum_sens_R*((-pnalz*(rho(i)**(pnalz-1)))*Elem_ene(i))
delem_vol(i) = elem_vol(i)*sum_sens_V
*ENDDO

L_stop = 1
L_1 = 0
L_2 = 1e16
dens_move = 0.2
opt_count = 0
```

```
*DOWHILE,L_stop

L_mid = 0.5*(L_2+L_1)
*DO,i,1,Num_elem
B_e(i) = (-delem_ene(i))/(L_mid*delem_vol(i))
B_ETRACK(j,i) = B_e(i)
rhoB_e = rho_f(i)*(B_e(i)**(n_damping))
rho_up = rho_f(i)+ dens_move
rho_down = rho_f(i) - dens_move
rhoBe_rhoup(1:2) = rhoB_e, rho_up
*VSCFUN, min1, MIN, rhoBe_rhoup
min1_1(1:2) = 1, min1
*VSCFUN, min2, MIN, min1_1
maxrd_min2(1:2) = rho_down, min2
*VSCFUN, max1, MAX, maxrd_min 2
max0_1(1:2) = rho_min, max1
*VSCFUN, max2, MAX, max0_1

rho_new(i) = max2

rho_diff(i) = ((rho_new(i) - rho_f(i))**(2))**(1/2)
*ENDDO

!-----------------------------------------------------------------------------------------------------------------------
```

```
! FILTRANDO DENSIDADE
*DO,i,1,Num_elem
v_size = viz_e_size(i)
*DO,e_v,1,v_size
ele_num = Vizin(i,e_v)
sum_hrv(e_v) = (H_e(i,e_v)/H_eS(e_v))
sum_hv(e_v)  = H_e(i,e_v)*elem_vol(ele_num)
sens_Rho(e_v) = (H_e(i,e_v)/H_eS(e_v))
*IF,e_v,EQ,v_size,THEN
e_v =e_v +1
*DO,cln,e_v,viz_alloc
            sum_hrv(cln) = 0
           sum_hv(cln) = 0
sens_Rho(cln) = 0
*ENDDO
*ENDIF

*ENDDO

*VSCFUN, sumhrv, SUM, sum_hrv
*VSCFUN, sumhv, SUM, sum_hv
*VSCFUN, sum_sens_Rho, SUM, Sens_Rho

rho(i) = sumhrv*rho_new(i)
```

```
r_new = rho_new(i)
Be_ = B_e(i)
e_volm(i) = elem_vol(i)*rho(i)
!obj_vol(i) = elem_vol(i)*obj_massa
e_ene = elem_ene(i)
d_eene = delem_ene(i)
d_vol = delem_vol(i)
/COM, rho_new: %i% %r_new%, dene: %d_eene%,Be = %Be_%
*ENDDO
!-----------------------------------------------------------------------------------------------------------------------

*VSCFUN, sum_evol, SUM, e_volm
!*VSCFUN, sum_objvol, SUM, obj_vol
it_sum_evol(j) = sum_evol
vol_ratio = sum_evol/sum_vol

*IF,vol_ratio,GT,obj_massa,THEN
L_1 = L_mid
*ELSE
L_2 = L_mid
*ENDIF

L_check = (L_2-L_1)/(L_1+L_2)
/COM, L1 : %L_1%, L2:%L_2%, L_mid: %L_mid%, vol_ratio: %vol_ratio%
```

```
*IF,L_check,LT,0.001,THEN
L_stop = 0
*ENDIF

opt_count = opt_count + 1
*ENDDO


*VSCFUN, change, MAX, rho_diff
!*VSCFUN, rhonew_max, min, rho_new

!j_1 = j-1
!*IF,j,GT,1,THEN
!change = ((total_ene(j)- total_ene(j_1))**(2))**(1/2)
*IF,change,LE,1e-2,THEN
stop_otim = 0
*ENDIF
!*ENDIF



*IF,loop,GE,max_loop,THEN
stop_otim = 0
```

```
*ENDIF

*DO,i,1,Num_elem
rho_f(i) = rho_new(i)
!rho_fis = rho(i)
rho_it(j,i) = rho(i)
e_ene_it(j,i)  = elem_ene(i)
Be_it(j,i)  = B_e(i)

!/COM, rho_new: %i% %r_new%, dene: %d_eene%,Be = %Be_%
*ENDDO

!*CFOPEN,rho_it_ansys,txt,
!*VWRITE,rho(1),
!(1F6.4)
!*CFCLOS

!*CFOPEN,rhonew_it_ansys,txt,
!*VWRITE,rho_new(1),
!(1F6.4)
!*CFCLOS

!*CFOPEN,ce_it_ansys,txt,
!*VWRITE,elem_ene(1),
!(1F10.4)
```

```
!*CFCLOS

!*CFOPEN,dc_it_ansys,txt,
!*VWRITE,delem_ene(1),
!(1F10.4)
!*CFCLOS


!*CFOPEN,Be_it_ansys,txt,
!*VWRITE,B_e(1),
!(1F6.4)
!*CFCLOS

/COM,Iteracao numero : %loop% ultima mudanca de :%change%
!*IF,Check_ene,EQ,0,THEN
!*ASK, Check_ene,Compare energias,
!*ENDIF
loop = loop + 1
j = j + 1
*ENDDO

*CFOPEN,rho_it_ansys,txt,
*VWRITE,rho_it(1,1),rho_it(1,2),rho_it(1,3),rho_it(1,4),rho_it(1,5),rho_it(1,6),rho_it(1,7),rho_it(1,8),rho_it(1,9),rho_it(1,10),rho_it(1,11),rho_it(1,12),rho_it(1,13),rho_it(
1,14),rho_it(1,15)
```

```
%G      %G     %G     %G     %G     %G     %G     %G     %G     %G     %G     %G     %G     %G     %G
*CFCLOS

*CFOPEN,ce_it_ansys,txt,
*VWRITE,e_ene_it(1,1),e_ene_it(1,2),e_ene_it(1,3),e_ene_it(1,4),e_ene_it(1,5),e_ene_it(1,6),e_ene_it(1,7),e_ene_it(1,8),e_ene_it(1,9),e_ene_it(1,10),e_ene_it(1,11),e_ene_it(1,12),e_ene_it(1,13),e_ene_it(1,14),e_ene_it(1,15)
%G      %G     %G     %G     %G     %G     %G     %G     %G     %G     %G     %G     %G     %G     %G
*CFCLOS

!-----------------------------------------------------------------------

ESEL,NONE
*DO,i,1,Num_elem
r_ele =rho_F(i)
*IF,r_ele,GE,0.5,THEN
ESEL,A ,elem,,i
*ENDIF
*ENDDO

H_Hs(1:Num_elem) = 0
*DO,i,1,Num_elem
v_size = viz_e_size(i)
*DO,e_v,1,v_size
ele_num = Vizin(i,e_v)
```

```
sens_R(e_v) = (H_e(i,e_v)/H_eS(e_v))
sens_V(e_v) = H_e(i,e_v)/H_eS(e_v)
*IF,e_v,EQ,v_size,THEN
e_v =e_v +1
*DO,cln,e_v,viz_alloc
        sens_R(cln) = 0
        sens_V(cln) = 0
*ENDDO
*ENDIF

*ENDDO
*VSCFUN, HHs, SUM, sens_R
H_Hs(i) = HHs**(0.5)

*ENDDO

i_ene(1:loop) = 0
*do, i,1,loop
i_ene(i) = i
*enddo
```