

ANEXO A. PROGRAMAS DE OPTIMIZACIÓN.

A.1. Introducción.

Los archivos que componen el programa de optimización de primer orden son:

- *Ppal.txt*. Es el archivo principal para la optimización.
- *pe1.dv*. En este archivo se le asigna el valor de las variables de diseño en cada evaluación a las variables de diseño en el modelo.
- *Param.txt*. Aquí se establecen los parámetros del algoritmo de optimización y se realizan las definiciones de algunos vectores.
- *Definicion.txt*. Se establece el estimado inicial para la optimización y los intervalos de las variables de diseño.
- *Restricciones.txt*. En este archivo se establecen las restricciones del problema.
- *Fo.txt*. Aquí se calcula la función objetivo a minimizar.
- *Calcsmax.txt*. En este archivo se busca el máximo del parámetro s utilizado para la búsqueda monodimensional (Ecuación 3.4.1). Se busca imponiendo que las variables de diseño se mantengan dentro de su rango de variación.
- *Convergencia.txt*. Este archivo realiza la comparación entre el mínimo de la iteración actual y de la anterior para dar un valor que permita comprobar la convergencia.
- *Forest.txt*. En este archivo se penalizan las restricciones según el método de la penalización interior extendida.
- *Fototal.txt*. Aquí se evalúa la función objetivo penalizada.
- *Gradiente.txt*. En este archivo se calcula el gradiente.
- *Saurea.txt*. En este archivo se realiza la búsqueda monodimensional.

Para el programa de optimización de segundo orden se tiene el mismo esquema de archivos anteriores excepto el *ppal.txt* que pasa a llamarse *ppal2.txt*, además de los siguientes:

- *Calcms.txt*. Calcula la matriz S para aplicarla al gradiente según el método de Davidon-Fletcher-Powell.
- *Dirbusq.txt*. Obtiene la dirección de búsqueda usada en la optimización multidimensional.

A.2. Archivos del programa de optimización de primer orden.

A.2.1. Ppal.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Archivo principal de la optimización.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
/CLEAR, START
*get, hora, active, 0, time, wall
/input, param, txt
/input, definicion, txt

*do, externo, 1, Nmax, 1
    /input, convergencia, txt
    *if, dif, le, conv, then
        *exit
    *else
        interno=1
        /input, gradiente, txt

        *if, modgrad, le, tolgrad, then
            *exit
        *else
            *endif
!llamada a la función de cálculo de SMAX
        /input, calcsmax, txt
!llamada a saurea
        /input, saurea, txt
        *voper, smnDxQ, puntomin, mult, DxQ
        *voper, xmin, x0, add, smnDxQ
/output, estado%externo%%interno%, out
*stat, xmin
*stat, fobj
*stat, smax
/output
        *do, interno, 2, Nmax2, 1
            /input, convergencia, txt
            *if, dif, le, conv2, then
                *exit
            *else
                *vfun, x0, copy, xmin
!llamada al gradiente
                /input, gradiente, txt

                *if, modgrad, le, tolgrad, then
                    *exit
                *else
                    *endif
!llamada a la función de cálculo de SMAX
                /input, calcsmax, txt
!llamada a saurea
                /input, saurea, txt
                *voper, smnDxQ, puntomin, mult, DxQ
                *voper, xmin, x0, add, smnDxQ
            *endif
/output, estado%externo%%interno%, out
*stat, xmin
*stat, fobj
*stat, smax
/output
            *enddo
            landa=.1*landa
            epsilon=-Cte*landa**expon
        *endif
    *enddo
*get, hora2, active, 0, time, wall
hora3=(hora2-hora)*60
/output, resultado, out
*status
*STATUS, XMIN
/output

```

A.2.2. Param.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Definición de los parámetros de optimización
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
nvd=2      !Número de variables de diseño
nrest=1     !Número de restricciones
Nsaurea=4   !Número máximo de iteraciones en la búsqueda monodimensional
Nmax=5      !Número máximo de actualizaciones del parámetro de
penalización
Nmax2=3     !Número máximo de iteraciones del bucle interno de búsqueda
tolq=1e-4   !Tolerancia en la variable dependiente para la búsqueda
monodimensional
tolab=1e-4  !Tolerancia en la variable independiente para la búsqueda
monodimensional
conv=1e-2   !Tolerancia para la convergencia global
conv2=3e-2  !Tolerancia para la convergencia en el bucle interno de búsqueda

landa=1/100 !Valor inicial del parámetro de penalización

tolgrad=1e-3 !Tolerancia en el gradiente
tolfront=1e-2 !Tolerancia en la frontera

! Otros parámetros de la optimización
Cte=750
expon=.5
eps=1e-2
fmult=1
epsilon=-Cte*landa**expon

*dim,x0,,nvd
*dim,x00,,nvd
*dim,x1,,nvd
*dim,x2,,nvd
*dim,xmin,,nvd
*dim,DxQ,array,nvd
*dim,aDxQ,array,nvd
*dim,bDxQ,array,nvd
*dim,smDxQ,array,nvd
*dim,smnDxQ,array,nvd
*dim,xa,,nvd
*dim,xb,,nvd
*dim,xsmax,,nvd
*dim,xfo,,nvd
*dim,xfo1,,nvd
*dim,restones,,nrest

*dim,desvar0,,nvd
*dim,desvar1,,nvd
*dim,desvar2,,nvd
*dim,x0aux1,,nvd
*dim,x0aux2,,nvd

*dim,difv,,nvd

```

A.2.3. Definición.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Definición del estimado inicial
!! y de los intervalos factibles de
!! las variables de diseño
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!Estimado inicial
desvar0(1)=8,3

!limites inferiores
desvar1(1)=7.5,1

```

!limites superiores

desvar2(1)=15,6

¡ Se obtiene x0 como una variable de diseño adimensional (entre 0 y 1)

*voper,x0aux1,desvar0,sub,desvar1

*voper,x0aux2,desvar2,sub,desvar1

*voper,x0,x0aux1,div,x0aux2

*do,indxdes,1,nvd

 x1(indxdes)=0

 x2(indxdes)=1

*enddo

*vfun,x00,copy,x0

A.2.4. Pe1.dv

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Definición de las variables de diseño
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

r=xfo(1)*xaux2(1)+desvar0(1)

rcubo=xfo(2)*xaux2(2)+desvar0(2)

A.2.5. Fo.txt

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Evaluación de la función objetivo adimensionada
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

/input,modelo,inp

F0bj=sdmx1/0.3

A.2.6. Restricciones.txt

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Definición de las restricciones
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

restones(1)=((dejepd-1)*d/2-xfo(1))/4

A.2.7. Calcsmax.txt

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Cálculo de Smax (=frontera)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

*do,binic,1,nvd,1

 *if,DxQ(binic),eq,0,then

 *else

 smmm=(x1(binic)-x0(binic))/DxQ(binic)

 SMM=(x2(binic)-x0(binic))/DxQ(binic)

 *if,smmm,gt,0,then

 frontera=smmm

 *else

 frontera=SMM

 *endif

 *exit

 *endif

*enddo

*do,bact,binic,nvd,1

 *if,DxQ(bact),eq,0,then

 *else

 smmm=(x1(bact)-x0(bact))/DxQ(bact)

 SMM=(x2(bact)-x0(bact))/DxQ(bact)

 *if,smmm,gt,0,then

 *if,smmm,lt,frontera,then

 frontera=smmm

 *endif

 *else

 *if,SMM,lt,frontera,then

 frontera=SMM

```

                *endif
            *endif
        *endif
    *enddo
    frontera=frontera*fmult
    *if, frontera, lt, 0, then
        frontera=0
    *else
    *endif

```

A.2.8. Convergencia.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Obtención del valor que se use
!! como comparación en el criterio
!! de convergencia (dif)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
dif=0
*do, converg, 1, nvd, 1
    difv(converg)=abs(x0(converg)-xmin(converg))/x0(converg)
*enddo
*vscfun, dif, rms, difv

```

A.2.9. Forest.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Penalización de las restricciones
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*do, II, 1, nrest, 1
/output, restones, out, , append
*if, restones(II), le, epsilon, then
    restones(II)=-1/restones(II)
*else
    restones(II)=- (2*epsilon-restones(II))/epsilon**2
*endif
/output, ,
*enddo

```

A.2.10. Fototal.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Generación de la función objetivo penalizada
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
/input, fo, txt
/input, restricciones, txt
/input, forest, txt
ypenal=0
*vscfun, ypenal, sum, restones
/output, penalización, out, , append
*stat, ypenal
F0bj=F0bj+landa*ypenal
/output, ,

```

A.2.11. Gradiente.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Obtención del gradiente
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*vfun, xfo, copy, x0
/input, fototal, txt
qcentro=F0bj
*do, bgrad, 1, nvd, 1
    xfo1(bgrad)=x0(bgrad)+eps*(x2(bgrad)-x1(bgrad))
*enddo
*do, bgrad, 1, nvd, 1
    *vfun, xfo, copy, x0
    xfo(bgrad)=xfo1(bgrad)
/output, fototal, txt
DxQ(bgrad)=- (Fobj-qcentro)/(eps*(x2(bgrad)-x1(bgrad)))

! Evitar que el gradiente haga que se intente sobrepasar
! un límite del intervalo de una variable de diseño
*if, ((x0(bgrad)-x1(bgrad))/(x2(bgrad)-x1(bgrad))), lt, tolfrent, then

```

```

        *if, DxQ(bgrad), lt, 0, then
            DxQ(bgrad)=0
        *else
        *endif
    *elseif, ((x2(bgrad)-x0(bgrad))/(x2(bgrad)-x1(bgrad))), lt, tolfrent, then
        *if, DxQ(bgrad), lt, 0, then
            DxQ(bgrad)=0
        *else
        *endif
    *else
    *endif
    *enddo
    *vscfun, modgrad, rms, DxQ
    *voper, DxQ, DxQ, div, modgrad
    /output, gradiente, out, , append
    *stat
    *stat, Dxq
    *stat, x0
    /output, ,

```

A.2.12. Saurea.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Búsqueda monodimensional
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Au=(1+5*.5)/2
s1=0
puntoa=s1+(frontera-s1)/Au
puntob=s1+(frontera-s1)*(1-1/Au)

/output, SecAurea, out, , append
*voper, aDxQ, puntoa, mult, DxQ
*voper, bDxQ, puntob, mult, DxQ
*voper, smDxQ, frontera, mult, DxQ
*voper, xa, x0, add, aDxQ
*voper, xb, x0, add, bDxQ
*voper, xmax, x0, add, smDxQ

*vfun, xfo, copy, xa
/input, fototal, txt
Qa=Fobj

*vfun, xfo, copy, xb
/input, fototal, txt
Qb=Fobj

*vfun, xfo, copy, xmax
/input, fototal, txt
Qsmax=Fobj

*do, bsaurea, 1, Nsaurea, 1
/output, SecAurea, out, , append

inter=(puntoa-puntob)/frontera
*if, inter, le, tolab, then
*exit
*else
funcion=abs(Qa-Qb)
*if, funcion, le, tolq, then
*exit
*else
*if, Qa, ge, Qb, then
frontera=puntoa
puntoa=puntob
puntob=s1+(frontera-s1)*(1-1/Au)

*voper, aDxQ, puntoa, mult, DxQ
*voper, bDxQ, puntob, mult, DxQ
*voper, smDxQ, frontera, mult, DxQ
*voper, xa, x0, add, aDxQ
*voper, xb, x0, add, bDxQ
*voper, xmax, x0, add, smDxQ

Qsmax=Qa
Qa=Qb
*vfun, xfo, copy, xb

```

```

/input, fototal, txt
Qb=F0bj

*else
s1=puntob
puntob=puntoa
puntoa=s1+(frontera-s1)/Au

*voper, aDxQ, puntoa, mult, DxQ
*voper, bDxQ, puntob, mult, DxQ
*voper, smDxQ, frontera, mult, DxQ
*voper, xa, x0, add, aDxQ
*voper, xb, x0, add, bDxQ
*voper, xmax, x0, add, smDxQ

Qs1=Qb
Qb=Qa
*vfun, xfo, copy, xa
/input, fototal, txt
Qa=F0bj

*endif
*endif
*endif
/output, ,
*enddo

! Ajuste de los tres puntos por una parábola
dena=-puntob*puntoa+puntob*frontera+puntoa**2-frontera*puntoa
denb=puntob*puntoa-frontera*puntoa-puntob**2+puntob*frontera
dens=puntob*puntoa-frontera*puntoa+frontera**2-puntob*frontera

! Cálculo analítico del mínimo de la parábola
puntomin=(( (puntob+frontera)*Qa/dena)-
((puntoa+frontera)*Qb/denb)+((puntoa+puntob)*Qsmax/dens))/((2*Qa/dena)-
(2*Qb/denb)+(2*Qsmax/dens))
*if, puntomin, lt, s1, then
puntomin=puntob
*endif
*if, puntomin, gt, frontera, then
puntomin=frontera
*endif
/output, xmin, out, , append
*stat, xmin
/output, ,

```

A.3. Archivos del programa de optimización de primer orden.

Los archivos son los mismos que en el programa de primer orden exceptuando los que se listan a continuación:

A.3.1. Ppal2.txt

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Archivo principal de la optimización.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
/CLEAR,NOSTART
*get,hora,active,0,time,wall
!/output,pe1,aut
/input,param,txt
/input,definicion,txt

*do,externo,1,Nmax,1
    /input,convergencia,txt
    *if,dif,le,conv,then
        *exit
    *else
        interno=1
        *if,externo,ne,1,then
            *vfun,x0,copy,xmin      !!!novedad
        *else
            *endif
        /input,gradiente,txt

        *if,modgrad,le,tolgrad,then
            *exit
        *else
            *endif
        /input,dirbusq,txt
!llamada a la función de cálculo de SMAX
        /input,calcsmax,txt
!llamada a saurea
        /input,saurea,txt
        *voper,smnDxQ,puntomin,mult,dbusq
        *voper,xmin,x0,add,smnDxQ
/output,estado%externo%%interno%,out
*stat,xmin
*stat,fobj
*stat,smax
/output
        *vfun,dxq0,copy,dxq
        *do,interno,2,Nmax2,1
            /input,convergencia,txt
            *if,dif,le,conv2,then
                *exit
            *else
                *voper,pk,xmin,sub,x0
                *vfun,x0,copy,xmin
!llamada al gradiente
                /input,gradiente,txt

                *if,modgrad,le,tolgrad,then
                    *exit
                *else
                    *endif
                *voper,qk,dxq,sub,dxq0
                /input,calcms,txt
                /input,dirbusq,txt
!llamada a la función de cálculo de SMAX
                /input,calcsmax,txt
!llamada a saurea
                /input,saurea,txt
                *voper,smnDxQ,puntomin,mult,dbusq
                *voper,xmin,x0,add,smnDxQ
                /output,evolucion,out,,append
                *stat,xmin
                *stat
```

```

                                /output,,
                                *vfun,dxq0,copy,dxq
                            *endif
/output,estado%externo%%interno%,out
*stat,xmin
*stat,fobj
*stat,smax
/output
                                *enddo
                                landa=.1*landa
                                epsilon=-Cte*landa**expon
                            *endif
*enddo
*get,hora2,active,0,time,wall
hora3=(hora2-hora)*60
/output,solu,out,,append
*status
*STATUS,XMIN
/output,,

```

A.3.2. Param.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Definición de los parámetros de optimización
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
nvd=2          !Número de variables de diseño
nrest=1        !Número de restricciones
Nsaurea=4      !Número máximo de iteraciones en la búsqueda monodimensional
Nmax=5         !Número máximo de actualizaciones del parámetro de
penalización
Nmax2=3        !Número máximo de iteraciones del bucle interno de búsqueda
tolq=1e-4      !Tolerancia en la variable dependiente para la búsqueda
monodimensional
tolab=1e-4     !Tolerancia en la variable independiente para la búsqueda
monodimensional
conv=1e-2      !Tolerancia para la convergencia global
conv2=3e-2     !Tolerancia para la convergencia en el bucle interno de búsqueda

landa=1/100    !Valor inicial del parámetro de penalización

tolgrad=1e-3   !Tolerancia en el gradiente
tolfront=1e-2  !Tolerancia en la frontera

! Otros parámetros de la optimización
Cte=750
expon=.5
eps=1e-2
fmult=1
epsilon=-Cte*landa**expon

*dim,x0,,nvd
*dim,x00,,nvd
*dim,x1,,nvd
*dim,x2,,nvd
*dim,xmin,,nvd
*dim,DxQ,array,nvd
*dim,aDxQ,array,nvd
*dim,bDxQ,array,nvd
*dim,smDxQ,array,nvd
*dim,smnDxQ,array,nvd
*dim,xa,,nvd
*dim,xb,,nvd
*dim,xsmax,,nvd
*dim,xfo,,nvd
*dim,xfo1,,nvd
*dim,restones,,nrest

*dim,desvar0,,nvd
*dim,desvar1,,nvd
*dim,desvar2,,nvd
*dim,x0aux1,,nvd

```

```

*dim,x0aux2,,nvd
*dim,difv,,nvd

*dim,dbusq,,nvd
*dim,DxQ0,,nvd
*dim,qk,,nvd
*dim,pk,,nvd
*dim,qkt,,1,nvd
*dim,pkt,,1,nvd
*dim,vdaux2,,1,nvd

*dim,mats,,nvd,nvd
*dim,maux1,,nvd,nvd
*dim,maux2,,nvd,nvd
*dim,msuma,,nvd,nvd

! Matriz S inicial (identidad)
*do,indmatest,1,nvd,1
    mats(indmatest,indmatest)=1
*enddo

*dim,diaux1,,1
*dim,diaux2,,1

```

A.3.3. Calcsmax.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Cálculo de Smax (=frontera)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*do,binic,1,nvd,1
    *if,dbusq(binic),eq,0,then
    *else
        smmm=(x1(binic)-x0(binic))/dbusq(binic)
        SMM=(x2(binic)-x0(binic))/dbusq(binic)
        *if,smmm,gt,0,then
            frontera=smmm
        *else
            frontera=SMM
        *endif
        *exit
    *endif
*enddo
*do,bact,binic,nvd,1
    *if,dbusq(bact),eq,0,then
    *else
        smmm=(x1(bact)-x0(bact))/dbusq(bact)
        SMM=(x2(bact)-x0(bact))/dbusq(bact)
        *if,smmm,gt,0,then
            *if,smmm,lt,frontera,then
                frontera=smmm
            *endif
        *else
            *if,SMM,lt,frontera,then
                frontera=SMM
            *endif
        *endif
    *endif
*enddo
frontera=frontera*fmult
*if,frontera,lt,0,then
    frontera=0
*else
*endif

```

A.3.4. Saurea.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Búsqueda monodimensional
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Au=(1+5** .5)/2
s1=0
puntoa=s1+(frontera-s1)/Au
puntob=s1+(frontera-s1)*(1-1/Au)

```

```

/output, SecAurea, out, , append
*voper, aDxQ, puntoa, mult, dbusq
*voper, bDxQ, puntob, mult, dbusq
*voper, smDxQ, frontera, mult, dbusq
*voper, xa, x0, add, aDxQ
*voper, xb, x0, add, bDxQ
*voper, xmax, x0, add, smDxQ

*vfun, xfo, copy, xa
/input, fototal, txt
Qa=F0bj

*vfun, xfo, copy, xb
/input, fototal, txt
Qb=F0bj

*vfun, xfo, copy, xmax
/input, fototal, txt
Qsmax=F0bj

*do, bsaurea, 1, Nsaurea, 1
/output, SecAurea, out, , append

inter=(puntoa-puntob)/frontera
*if, inter, le, tolal, then
*exit
*else
funcion=abs(Qa-Qb)
*if, funcion, le, tolq, then
*exit
*else
*if, Qa, ge, Qb, then
frontera=puntoa
puntoa=puntob
puntob=s1+(frontera-s1)*(1-1/Au)

*voper, aDxQ, puntoa, mult, dbusq
*voper, bDxQ, puntob, mult, dbusq
*voper, smDxQ, frontera, mult, dbusq
*voper, xa, x0, add, aDxQ
*voper, xb, x0, add, bDxQ
*voper, xmax, x0, add, smDxQ

Qsmax=Qa
Qa=Qb
*vfun, xfo, copy, xb
/input, fototal, txt
Qb=F0bj

*else
s1=puntob
puntob=puntoa
puntoa=s1+(frontera-s1)/Au

*voper, aDxQ, puntoa, mult, dbusq
*voper, bDxQ, puntob, mult, dbusq
*voper, smDxQ, frontera, mult, dbusq
*voper, xa, x0, add, aDxQ
*voper, xb, x0, add, bDxQ
*voper, xmax, x0, add, smDxQ

Qs1=Qb
Qb=Qa
*vfun, xfo, copy, xa
/input, fototal, txt
Qa=F0bj

*endif
*endif
*endif
/output, ,
*enddo

! Ajuste de los tres puntos por una parábola
dena=-puntob*puntoa+puntob*frontera+puntoa**2-frontera*puntoa
denb=puntob*puntoa-frontera*puntoa-puntob**2+puntob*frontera

```

```

dens=puntob*puntoa-frontera*puntoa+frontera**2-puntob*frontera
! Cálculo analítico del mínimo de la parábola
puntomin=((puntob+frontera)*Qa/dena)-
((puntoa+frontera)*Qb/denb)+((puntoa+puntob)*Qsmax/dens))/((2*Qa/dena)-
(2*Qb/denb)+(2*Qsmax/dens))
*if, puntomin, lt, s1, then
puntomin=puntob
*endif
*if, puntomin, gt, frontera, then
puntomin=frontera
*endif
/output, xmin, out, , append
*stat, xmin
/output, ,

```

A.3.5. Dirbusq.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Obtenci de la direcci de b queda
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

*moper, dbusq, Mats, mult, DxQ
/output, dirbusqueda, out, , append
*stat
*stat, dxq
*stat, mats
*stat, dbusq
*stat, x0
/output, ,
/output, xmin, out
*stat, xmin
*stat, dbusq
/output, ,

```

A.3.6. CalcMS.txt

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Cálculo de la matriz S
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*mfun, pkt, tran, pk
*mfun, qkt, tran, qk
*moper, maux1, pk, mult, pkt

*moper, daux1, pkt, mult, pk
denaux1=daux1(1)

*voper, maux1, maux1, div, denaux1
*moper, maux2, qk, mult, qkt
*moper, maux2, maux2, mult, mats
*moper, maux2, mats, mult, maux2
*moper, vdaux2, qkt, mult, mats

*moper, daux2, vdaux2, mult, qk
denaux2=daux2(1)

*voper, maux2, maux2, div, denaux2
*voper, msuma, maux1, sub, maux2
*voper, mats, mats, add, msuma

```