
ANEXO

I

CÓDIGO DE OPTIMIZACIÓN

En este *Anexo I* se incluye el código de las tres funciones más relevantes del programa **optimiza 3D**: la propia **optimiza 3D**, la de cálculo del mallado, **mallado 3D** y la que realiza los cálculos anuales **performance 3D**.

El fin de incluir estos códigos es facilitar la comprensión del mismo, para la persona que esté interesada en programar en Matlab o en analizar detalladamente los cálculos que se realizan en ellas.

Función optimiza 3D

function

```
[MAXPOW, MINCOST, THTOPTIM, TILTOPTIM, INDSOLUCION, SOLUCION]=optimiza3D(THTMIN,
THTMAX, TILTMIN, TILTMAX, DESIGNPOW, DESIGNDAY, DESIGNHOUR, Ndays, Nhours, RADIACION,
LATITUDE, MEANFLUX, SIGMA, ALTURAH, ANCHURAH, PEDESTH, RTOWER, XV, YV, CRITERIO)
```

%optimizafin.m--> función de optimización de campos con geometría compleja

%

```
THTMIN=75;           % altura mínima de la torre;
THTMAX=125;          % altura máxima de la torre
TILTMIN=25;          % ángulo de inclinación del receptor mínimo;
TILTMAX=50;          % ángulo de inclinación del receptor máximo;
DESIGNPOW=25;         % potencia en el punto de diseño;
DESIGNDAY=81;         % día del punto de diseño
DESIGNHOUR=12;        % hora del punto de diseño
LATITUDE=37.22;      % latitud
MEANFLUX=600;         % flujo medio en el receptor
SIGMA=2.9;            % error total sol+tracking +etc
ALTURAH=8.87;         % altura del helióstato
ANCHURAH=6.65;       % anchura del helióstato
PEDESTH=5;           % altura del pedestal
RTOWER=10;           % radio de la torre en m
RADIACION=0.869;
```

```
PENDIENTETERRENO=-pi/12;
```

```
%XV=[ -280  0  250   200   300    0    0  -200];
%YV=[   0  0 -250  -300  -400  -450  -400  -300];
```

```
XV=[350 -350 -350 350];
YV=[0 0 -500 -500];
```

```
%Ndays vector con el número de los días donde se estudia el performance
%Nhours vector con el número de horas donde se estudia el performance
```

```
%Ndays=[354 35 81 126 172];
%Nhours=[9 10 11 12 13 14 15];

Ndays=[172];
Nhours=[9 15];

%para ver el ángulo mínimo para que no de sombra en la parcela; ya lo tenemos
en cuenta en el performance

%minPendienteTerreno=MinPendienteMio(LATITUDE,Ndays,Nhours,TILTPLANE);

%CRITERIO nos da el criterio de optimización

%CRITERIO=['Maxpotanual'];
CRITERIO=['Mincosteporpotanual'];
%
%MAXPOW tiene sentido si se ha elegido el criterio de máxima potencia
%MINCOST tiene sentido si se ha elegido el criterio de mínimo coste
%THTOPTIM altura óptima
%TILTOPTIM inclinación óptima
%INDSOLUCION índice de la variable solución donde se encuentra el óptimo
%SOLUCION es un struct-array donde se han almacenado todos los resultados de
%todos los performances de todos los campos probados

AreaRec=DESIGNPOW*1000/MEANFLUX;
Rrec=sqrt(AreaRec/pi);

%inicializo para las comparaciones en la optimización

MAXPOW=0;
MINCOST=1e20;

%la optimización se hace en grupos de 3 alturas X 3 ángulos de receptor (9)
%posibilidades de hay tomamos las mejores y seguimos dividiendo
%es un sistema de optimización tipo Newton simple
%
INTENTONUM=0;           %Contador del número de campos calculados
MEJORA=1;              %mejora será igual a uno si existe mejora entre el
grupo de (9) posibilidades y el siguiente grupo de 9 intervalos más pequeños

MINCOSTBUCLE9=MINCOST;
MAXPOWBUCLE9=MAXPOW;
THTMINTOTAL=THTMIN;
THTMAXTOTAL=THTMAX;
TILTMINTOTAL=TILTMIN;
TILTMAXTOTAL=TILTMAX;

while MEJORA==1;

    THT=THTMIN:(THTMAX-THTMIN)/2:THTMAX
    TILTANG=TILTMIN:(TILTMAX-TILTMIN)/2:TILTMAX

%para estos ejemplos supondremos la altura de la torre igual al target

    [AZIMUT,ELEVATION]= solpos(LATITUDE,DESIGNDAY,DESIGNHOUR+0.01);
```

```

[cosinc]= cosdirinc(AZIMUT, ELEVATION);
cosinc=squeeze(cosinc)';

for jj=1:3           %índice relativo a las alturas de torre

    [CAMPO(jj).THT,Area]=malla3D(THT(jj),ALTURAH,ANCHURAH,XV,YV,PEDESTH,
    ALTURARECEPTOR,PENDIENTETERRENO);
    AIM(jj).THT=[0 0 THT(jj)];
    %CAMPO(jj).THT(:,3)=CAMPO(jj).THT(:,3)+PEDESTH;

    parejasvecinos=vecinos3D(CAMPO(jj).THT(:,1),CAMPO(jj).THT(:,2),
    CAMPO(jj).THT(:,3),10);

    distanaim=sqrt((CAMPO(jj).THT(:,1)-
    AIM(jj).THT(1,1)).^2+(CAMPO(jj).THT(:,2)-
    AIM(jj).THT(1,2)).^2+(CAMPO(jj).THT(:,3)-AIM(jj).THT(1,3)).^2);

    vectoraim=[(AIM(jj).THT(1,1)-CAMPO(jj).THT(:,1))./distanaim
                (AIM(jj).THT(1,2)-CAMPO(jj).THT(:,2))./distanaim
                (AIM(jj).THT(1,3)-CAMPO(jj).THT(:,3))./distanaim];

    [ATENUACION]=atenuacion(distanaim);

    [XN,YN,ZN,Normalhel,Cosrefhel]=vertices
    (CAMPO(jj).THT,ALTURAH,ANCHURAH,cosinc,AIM(jj).THT);

    %dibujaHelióstatos(CAMPO(jj).THT,THT(jj),XN,YN,ZN,Normalhel,Cosrefhel,cosinc,
    XV,YV);

    [SOMBRASTOTAL,Areautil]=sombrasplano(cosinc,XN,YN,ZN,parejasvecinos);
    %[BLOQUEOSTOTAL]=bloqueosplano(parejasvecinos,distanaim,vectoraim,XN,YN,ZN,
    CAMPO(jj).THT);
    [BLOQUEOSTOTAL]=bloqueosesfera(AIM(jj).THT,XN,YN,ZN,parejasvecinos);

    sizecampo=size(CAMPO(jj).THT);
    M=sizecampo(1);
    [SOMBRATERRENO]=sombraterreno(PENDIENTETERRENO,cosinc,M);

    for kk=1:3           %índice relativo a los ángulos de inclinación
        TILTANGU(kk).TILT=TILTANG(kk);
        VectorRec=[0 -cos(TILTANGU(kk).TILT*pi/180) -
                    sin(TILTANGU(kk).TILT*pi/180)];

        [SPILLGEO]=spillage(vectoraim,distanaim,VectorRec,
        Rrec,AIM(jj).THT,SIGMA);

        POTENCIA=Areautil.*(1-SOMBRATERRENO).*(1-SOMBRASTOTAL).*
        (1-BLOQUEOSTOTAL).*(1-SPILLGEO).*(1-PORCATENUACION)*RADIACION;
        POTENCIATOTAL=sum(POTENCIA);

        if POTENCIATOTAL<DESIGNPOW*1000
            disp('no hay potencia suficiente para este caso')
        else
            [potenciasort,indices]=sort(POTENCIA);
            potenciasortflip=flipud(potenciasort);

```

```

indicesflip=flipud(indices);
cumpotenciasort=cumsum(potenciasortflip);
iindices=find(cumpotenciasort<=DESIGNPOW*1000);
indicesok=indicesflip(iindices);
campocrop{jj,kk}=CAMPO(jj).THT(indicesok,:);
Campo=CAMPO(jj).THT(indicesok,:);

```

%HACEMOS LOS CALCULOS ANUALES

```

[MedhPOT, MedhCOSEFF, MedhSOMBterr, MedhSOMB, MedhBLOCK, MedhATEN, MedhSpill]=
performance3DMio(LATITUDE, Ndays, Nhours, RADIACION, AIM(jj).THT, Campo, THT(jj),
RTOWER, ANCHURAH, ALTURAH, PEDESTH, TILTANG(kk), Rrec, SIGMA, XV, YV,
PENDIENTETERRENO);

```

```

yearperfpow=sum(MedhPOT)    %potencia total anual

```

```

%INTENTONUM=INTENTONUM+1;

```

```

COSEFFECTLOSSESDESIGN=((ALTURAH.*ANCHURAH)-
Areautil(indicesok)).*RADIACION;
SOMBRATERRENOLOSSESDESIGN=Areautil(indicesok).
*SOMBRATERRENO(indicesok).*RADIACION;
SOMBRASLOSSESDESIGN=Areautil(indicesok).*
(1-SOMBRATERRENO(indicesok)).*SOMBRASTOTAL(indicesok).*RADIACION;
BLOQUEOSLOSSESDESIGN=Areautil(indicesok).
*(1-SOMBRATERRENO(indicesok)).
*(1-SOMBRASTOTAL(indicesok)).*(BLOQUEOSTOTAL(indicesok)).*RADIACION;
ATENLOSSESDESIGN=Areautil(indicesok).*
(1-SOMBRATERRENO(indicesok)).*(1-SOMBRASTOTAL(indicesok)).*
(1-BLOQUEOSTOTAL(indicesok)).*(ATENUACION(indicesok))*RADIACION;
SPILLLOSSESDESIGN=Areautil(indicesok).
*(1-SOMBRATERRENO(indicesok)).*(1-SOMBRASTOTAL(indicesok)).
*(1-BLOQUEOSTOTAL(indicesok)).*(SPILLGEO(indicesok)).
*(1-PORCATENUACION(indicesok))*RADIACION;

```

```

%XN,YN,ZN,Normalhel,Cosrefhel

```

```

XNDESIGN=XN(indicesok,:);
YNDESIGN=YN(indicesok,:);
ZNDESIGN=ZN(indicesok,:);
NormalhelDESIGN=Normalhel(indicesok,:);
CosrefhelDESIGN=Cosrefhel(indicesok,:);

```

```

if strcmp(CRITERIO, 'Maxpotanual')==1
    if yearperfpow>MAXPOW
        MAXPOW=yearperfpow
        THTOPTIM=THT(jj)
        TILTOPTIM=TILTANG(kk)
        %INDSOLUCION=INTENTONUM
        SOLUCION.CAMPO=Campo;
        SOLUCION.MedhPOT=MedhPOT;
        SOLUCION.MedhCOSEFF=MedhCOSEFF;
        SOLUCION.MedhSOMBterr=MedhSOMBterr;
        SOLUCION.MedhSOMB=MedhSOMB;
        SOLUCION.MedhBLOCK=MedhBLOCK;
        SOLUCION.MedhATEN=MedhATEN;
        SOLUCION.MedhSpill=MedhSpill;
    end
end

```

```

        SOLUCION.THT=THT(jj);
        SOLUCION.TILTANG=TILTANG(kk);
        SOLUCION.yearperfpow=yearperfpow;

        SOLUCIONDESIGN.POTDESIGN=POTENCIA(indicesok);
        SOLUCIONDESIGN.COSEFFDESIGN=COSEFFECTLOSSESDESIGN;
        SOLUCIONDESIGN.SOMBTERRENODESIGN=SOMBRATERRENOLOSSESDESIGN;
        SOLUCIONDESIGN.SOMBDESIGN=SOMBRASLOSSESDESIGN;
SOLUCIONDESIGN.BLOCKDESIGN=BLOQUEOSLOSSESDESIGN;
        SOLUCIONDESIGN.ATENDESIGN=ATENLOSSESDESIGN;
        SOLUCIONDESIGN.SpillDESIGN=SPILLLOSSESDESIGN;

SOLUCIONDESIGN.POTTOTALDESIGN=sum(SOLUCIONDESIGN.POTDESIGN);
        SOLUCIONDESIGN.XN=XNDESIGN;
        SOLUCIONDESIGN.YN=YNDESIGN;
        SOLUCIONDESIGN.ZN=ZNDESIGN;
        SOLUCIONDESIGN.Normalhel=NormalhelDESIGN;
        SOLUCIONDESIGN.Cosrefhel=CosrefhelDESIGN;

    end
else

    TORRECOST=costetorre(THT(jj));
    TERRENO COST=costeterreno(Area,PENDIENTETERRENO);
    RECCOST=costerec(pi*Rrec^2);
    sizecampo=size(Campo);
    NUMHEL=sizecampo(1);
    HELCOST=36371/166;
    HELCOST=NUMHEL*ALTURAH*ANCHURAH*HELCOST;
    TCOST=TORRECOST+RECCOST+HELCOST+TERRENO COST;

    COSTEPORPOTANUAL=TCOST/yearperfpow

%save SOLUCION
if COSTEPORPOTANUAL<MINCOST
    MINCOST=COSTEPORPOTANUAL;
    THTOPTIM=THT(jj)
    TILTOPTIM=TILTANG(kk)
    %INDSOLUCION=INTENTONUM;
    SOLUCION.CAMPO=Campo;
    SOLUCION.MedhPOT=MedhPOT;
    SOLUCION.MedhCOSEFF=MedhCOSEFF;
    SOLUCION.MedhSOMB TERR=MedhSOMB TERR;
    SOLUCION.MedhSOMB=MedhSOMB;
    SOLUCION.MedhBLOCK=MedhBLOCK;
    SOLUCION.MedhATEN=MedhATEN;
    SOLUCION.MedhSpill=MedhSpill;
    SOLUCION.THT=THT(jj);
    SOLUCION.TILTANG=TILTANG(kk);
    SOLUCION.yearperfpow=yearperfpow;
    SOLUCION.COSTEPORPOTANUAL=COSTEPORPOTANUAL;

    SOLUCIONDESIGN.POTDESIGN=POTENCIA(indicesok);
    SOLUCIONDESIGN.COSEFFDESIGN=COSEFFECTLOSSESDESIGN;
    SOLUCIONDESIGN.SOMBTERRENODESIGN=SOMBRATERRENOLOSSESDESIGN;

```

```
SOLUCIONDESIGN.SOMBDESIGN=SOMBRASLOSSESDESIGN;
SOLUCIONDESIGN.BLOCKDESIGN=BLOQUEOSLOSSESDESIGN;
SOLUCIONDESIGN.ATENDESIGN=ATENLOSSESDESIGN;
SOLUCIONDESIGN.SpillDESIGN=SPILLLOSSESDESIGN;

SOLUCIONDESIGN.POTTOTALDESIGN=sum(SOLUCIONDESIGN.POTDESIGN);

SOLUCIONDESIGN.COSTEPORPOTDESIGN=TCOST/SOLUCIONDESIGN.POTTOTALDESIGN;
SOLUCIONDESIGN.XN=XNDESIGN;
SOLUCIONDESIGN.YN=YNDESIGN;
SOLUCIONDESIGN.ZN=ZNDESIGN;
SOLUCIONDESIGN.Normalhel=NormalhelDESIGN;
SOLUCIONDESIGN.Cosrefhel=CosrefhelDESIGN;

end

end% if criterio
end %if
end% kk
end% jj

%evaluamos si merece la pena seguir optimizando, hacer otro grupo de 9

switch CRITERIO
case 'Maxpotannual'
if (SOLUCION.yearperfpow > MAXPOWBUCLE9 & (SOLUCION.yearperfpow-
MAXPOWBUCLE9)/MAXPOWBUCLE9 > 0.0005)
MAXPOWBUCLE9=SOLUCION.yearperfpow;
SOLUCIONBUCLE9.CAMPO=SOLUCION.CAMPO;
SOLUCIONBUCLE9.MedhPOT=SOLUCION.MedhPOT;
SOLUCIONBUCLE9.MedhCOSEFF=SOLUCION.MedhCOSEFF;
SOLUCIONBUCLE9.MedhSOMBterr=SOLUCION.MedhSOMBterr;
SOLUCIONBUCLE9.MedhSOMB=SOLUCION.MedhSOMB;
SOLUCIONBUCLE9.MedhBLOCK=SOLUCION.MedhBLOCK;
SOLUCIONBUCLE9.MedhATEN=SOLUCION.MedhATEN;
SOLUCIONBUCLE9.MedhSpill=SOLUCION.MedhSpill;
SOLUCIONBUCLE9.THT=SOLUCION.THT;
SOLUCIONBUCLE9.TILTANG=SOLUCION.TILTANG;
SOLUCIONBUCLE9.yearperfpow=SOLUCION.yearperfpow;

SOLUCIONDESIGNBUCLE9.POTDESIGN=SOLUCIONDESIGN.POTDESIGN;
SOLUCIONDESIGNBUCLE9.COSEFFDESIGN=SOLUCIONDESIGN.COSEFFDESIGN;

SOLUCIONDESIGNBUCLE9.SOMBTERRENODESIGN=SOLUCIONDESIGN.SOMBTERRENODESIGN;
SOLUCIONDESIGNBUCLE9.SOMBDESIGN=SOLUCIONDESIGN.SOMBDESIGN;
SOLUCIONDESIGNBUCLE9.BLOCKDESIGN=SOLUCIONDESIGN.BLOCKDESIGN;
SOLUCIONDESIGNBUCLE9.ATENDESIGN=SOLUCIONDESIGN.ATENDESIGN;
SOLUCIONDESIGNBUCLE9.SpillDESIGN=SOLUCIONDESIGN.SpillDESIGN;
SOLUCIONDESIGNBUCLE9.POTTOTALDESIGN=SOLUCIONDESIGN.POTTOTALDESIGN;
SOLUCIONDESIGNBUCLE9.XN=SOLUCIONDESIGN.XN;
SOLUCIONDESIGNBUCLE9.YN=SOLUCIONDESIGN.YN;
SOLUCIONDESIGNBUCLE9.ZN=SOLUCIONDESIGN.ZN;
SOLUCIONDESIGNBUCLE9.Normalhel=SOLUCIONDESIGN.Normalhel;
SOLUCIONDESIGNBUCLE9.Cosrefhel=SOLUCIONDESIGN.Cosrefhel;
```

%cambiamos las condiciones de contorno de altura e inclinación

```

    if THTOPTIM==THTMINTOTAL
        THTMAX=(THTMAX-THTMIN)/4+THTMIN;
    elseif THTOPTIM==THTMAXTOTAL
        THTMIN=THTMAX-(THTMAX-THTMIN)/4;
    else
        THTMIN=THTOPTIM-(THTMAX-THTMIN)/4;
        THTMAX=THTOPTIM+(THTMAX-THTMIN)/4;
    end
    if TILTOPTIM==TILTMINTOTAL
        TILTMAX=(TILTMAX-TILTMIN)/4+TILTMIN;
    elseif TILTOPTIM==TILTMAXTOTAL
        TILTMIN=TILTMAX-(TILTMAX-TILTMIN)/4;
    else
        TILTMIN=TILTOPTIM-(TILTMAX-TILTMIN)/4;
        TILTMAX=TILTOPTIM+(TILTMAX-TILTMIN)/4;
    end

    else %if MINCOST < MINCOSTBUCLE9 no hay mejora
        MEJORA=0
    end

case 'Mincosteporpotanual'
    if (SOLUCION.COSTEPORPOTANUAL < MINCOSTBUCLE9 & (MINCOSTBUCLE9-
        SOLUCION.COSTEPORPOTANUAL)/MINCOSTBUCLE9 > 0.0005)

        MINCOSTBUCLE9=SOLUCION.COSTEPORPOTANUAL;
        SOLUCIONBUCLE9.CAMPO=SOLUCION.CAMPO;
        SOLUCIONBUCLE9.MedhPOT=SOLUCION.MedhPOT;
        SOLUCIONBUCLE9.MedhCOSEFF=SOLUCION.MedhCOSEFF;
        SOLUCIONBUCLE9.MedhSOMBterr=SOLUCION.MedhSOMBterr;
        SOLUCIONBUCLE9.MedhSOMB=SOLUCION.MedhSOMB;
        SOLUCIONBUCLE9.MedhBLOCK=SOLUCION.MedhBLOCK;
        SOLUCIONBUCLE9.MedhATEN=SOLUCION.MedhATEN;
        SOLUCIONBUCLE9.MedhSpill=SOLUCION.MedhSpill;
        SOLUCIONBUCLE9.THT=SOLUCION.THT;
        SOLUCIONBUCLE9.TILTANG=SOLUCION.TILTANG;
        SOLUCIONBUCLE9.yearperfpow=SOLUCION.yearperfpow;
        SOLUCIONBUCLE9.COSTEPORPOTANUAL=SOLUCION.COSTEPORPOTANUAL;

        SOLUCIONDESIGNBUCLE9.POTDESIGN=SOLUCIONDESIGN.POTDESIGN;
        SOLUCIONDESIGNBUCLE9.COSEFFDESIGN=SOLUCIONDESIGN.COSEFFDESIGN;
        SOLUCIONDESIGNBUCLE9.SOMBTERRENODESIGN=SOLUCIONDESIGN.SOMBTERRENODESIGN;
        SOLUCIONDESIGNBUCLE9.SOMBDESIGN=SOLUCIONDESIGN.SOMBDESIGN;
        SOLUCIONDESIGNBUCLE9.BLOCKDESIGN=SOLUCIONDESIGN.BLOCKDESIGN;
        SOLUCIONDESIGNBUCLE9.ATENDESIGN=SOLUCIONDESIGN.ATENDESIGN;
        SOLUCIONDESIGNBUCLE9.SpillDESIGN=SOLUCIONDESIGN.SpillDESIGN;
        SOLUCIONDESIGNBUCLE9.POTTOTALDESIGN=SOLUCIONDESIGN.POTTOTALDESIGN;
        SOLUCIONDESIGNBUCLE9.COSTEPORPOTDESIGN=SOLUCIONDESIGN.COSTEPORPOTDESIGN;
        SOLUCIONDESIGNBUCLE9.XN=SOLUCIONDESIGN.XN;
        SOLUCIONDESIGNBUCLE9.YN=SOLUCIONDESIGN.YN;
        SOLUCIONDESIGNBUCLE9.ZN=SOLUCIONDESIGN.ZN;
        SOLUCIONDESIGNBUCLE9.Normalhel=SOLUCIONDESIGN.Normalhel;
        SOLUCIONDESIGNBUCLE9.Cosrefhel=SOLUCIONDESIGN.Cosrefhel;

```

```

%cambiamos las condiciones de contorno de altura e inclinación

if THTOPTIM==THTMINTOTAL
    THTMAX=(THTMAX-THTMIN)/4+THTMIN;
elseif THTOPTIM==THTMAXTOTAL
    THTMIN=THTMAX-(THTMAX-THTMIN)/4;
else
    THTMIN=THTOPTIM-(THTMAX-THTMIN)/4;
    THTMAX=THTOPTIM+(THTMAX-THTMIN)/4;
end
if TILTOPTIM==TILTMINTOTAL
    TILTMAX=(TILTMAX-TILTMIN)/4+TILTMIN;
elseif TILTOPTIM==TILTMAXTOTAL
    TILTMIN=TILTMAX-(TILTMAX-TILTMIN)/4;
else
    TILTMIN=TILTOPTIM-(TILTMAX-TILTMIN)/4;
    TILTMAX=TILTOPTIM+(TILTMAX-TILTMIN)/4;
end

else %if MINCOST < MINCOSTBUCLE9 no hay mejora
    MEJORA=0
end
end %switch
end % end while
%save SOLUCION

pintasolucion3D(SOLUCIONBUCLE9,SOLUCIONDESIGNBUCLE9,ALTURAH,ANCHURAH,
PENDIENTETERRENO, CRITERIO,XV,YV,cosinc,PEDESTH);

```

Función performance 3D

```

function[MedhPOT,MedhCOSEFF,MedhSOMBterr,MedhSOMB,MedhBLOCK,MedhATEN,
MedhSpill]=performance3D(Latitude,Ndays,Nhours,RADIACION,Aim,Campo,THT,RTOWER
,ANCHURAH,ALTURAH,PEDEST,TILTREC,Rrec,SIGMA,XCENTRO,YCENTRO,ZCENTRO,HT,
PendienteTerreno,XV,YV);

%performance.m -->calcula el número total de kW que van a parar al receptor
en los días Ndays a las horas Nhours

% Latitude latitud en grados
% Ndays vector con el número de los días donde se estudia el performance
% Ndays=[354 35 81 126 172];
% Número de horas para cada día Nhours=[9 10 11 12];

```

```

% RADIACION
% Aim es un vector 1X3 con las coordenadas X Y Z del aiming point
% Campo es una matriz de No helióstatos X 2 con las coordenadas X e Y de cada
% helióstato
% THT es la altura total de la torre, la necesito para calcular las sombras
% RTOWER es el radio de la torre o de su circulo envolvente equivalente lo
% usamos para calcular las sombras
% ANCHURAH es la anchura del helióstato
% ALTURAH es la altura del helióstato
% PEDEST Altura del pedestal
% TILTREC= inclinación del receptor respecto a la vertical en grados
% Rrec radio del receptor
% SIGMATOTAL en mrad (conv del del sol mas errores)
% MedhPOT MedhCOSEFF MedhSOMB MedhBLOCK MedhATEN MedhSpill son vectores de
dimensiones igual a número de helióstatos nos dan los valores anuales de

%MedhPOT potencia suministrada en kW)
%MedhCOSEFF efecto coseno (en kW)
%MedhSOMB MedhBLOCK MedhATEN MedhSpill (sombras, bloqueos,atenuacion y
spill)

sizecampo=size(Campo);
M=sizecampo(1);
%inicializo contadores

POTENCIAACUM=zeros(M,1);

COSEFFECTACUM=zeros(M,1);
SOMBRATERRACUM=zeros(M,1);
SOMBRASACUM=zeros(M,1);
BLOQUEOSACUM=zeros(M,1);
ATENACUM=zeros(M,1);
SPILLACUM=zeros(M,1);

CONTADORHORAS=0;

parejasvecinos=vecinos3D(Campo(:,1),Campo(:,2),Campo(:,3),10);

for i=1:length(Ndays)
    for j=1:length(Nhours)

        [AZIMUT,ELEVATION]=solpos(Latitude,Ndays(i),Nhours(j)+0.01,TILTPLANE);
        [cosinc]= cosdiincMMios(AZIMUT, ELEVATION);
        cosinc=squeeze(cosinc)';

        %Campo(:,3)=zeros(size(Campo(:,2)))+PEDEST;

        distanaim=sqrt((Campo(:,1)-Aim(1,1)).^2+(Campo(:,2)-
Aim(1,2)).^2+(Campo(:,3)-Aim(1,3)).^2);
        vectoraim=[(Aim(1,1)-Campo(:,1))./distanaim (Aim(1,2)-
Campo(:,2))./distanaim (Aim(1,3)-Campo(:,3))./distanaim];
        [ATENUACION]=atenuacion(distanaim);
        [XN,YN,ZN,Normalhel,Cosrefhel]=vertices(Campo,ALTURAH,ANCHURAH,cosinc,XCENTRO
,YCENTRO,ZCENTRO,HT,PendienteTerreno);

```

```

%dibujaHelióstatos(Campo,THT,XN,YN,ZN,Normalhel,Cosrefhel,cosinc,XV,YV);

[SOMBRASTOTAL,Areautil]=sombrasplano(cosinc,XN,YN,ZN,parejasvecinos);

%[BLOQUEOSTOTAL]=bloqueosplano(parejasvecinos,distanaim,vectoraim,XN,YN,ZN,
CAMPO);
[BLOQUEOSTOTAL]=bloqueosesfera(Aim,XN,YN,ZN,parejasvecinos);

[SOMBRASTERRENO]=sombraterreno(PendienteTerreno,cosinc,M);

VectorRec=[0 -cos(TILTREC*pi/180) -sin(TILTREC*pi/180)];
[SPILLGEO]=geospill(vectoraim,distanaim,VectorRec,Rrec,Aim,SIGMA);
POTENCIA=Areautil.*(1-SOMBRASTERRENO).*(1-SOMBRASTOTAL).*(1-BLOQUEOSTOTAL).*(1-SPILLGEO).*(1-PORCATENUACION)*RADIACION;

%hallamos las pérdidas en kW
COSEFFECTLOSSES=((ALTURAH.*ANCHURAH)-Areautil).*RADIACION;
SOMBRATERRENOLOSSES=Areautil.*SOMBRASTERRENO;
SOMBRASLOSSES=Areautil.*(1-SOMBRASTERRENO).*SOMBRASTOTAL.*RADIACION;
BLOQUEOSLOSSES=Areautil.*(1-SOMBRASTERRENO).*(1-SOMBRASTOTAL).*(BLOQUEOSTOTAL).*RADIACION;
ATENLOSSES=Areautil.*(1-SOMBRASTERRENO).*(1-SOMBRASTOTAL).*(1-BLOQUEOSTOTAL).*(PORCATENUACION)*RADIACION;
SPILLLOSSES=Areautil.*(1-SOMBRASTERRENO).*(1-SOMBRASTOTAL).*(1-BLOQUEOSTOTAL).*(SPILLGEO).*(1-PORCATENUACION)*RADIACION;
%% hallamos los valores acumulados por helióstato

POTENCIAACUM=POTENCIAACUM+POTENCIA;%datos por helióstato
COSEFFECTACUM=COSEFFECTACUM+ COSEFFECTLOSSES;
SOMBRATERACUM=SOMBRATERACUM+SOMBRATERRENOLOSSES;
SOMBRASACUM=SOMBRASACUM+SOMBRASLOSSES;
BLOQUEOSACUM=BLOQUEOSACUM+BLOQUEOSLOSSES;
ATENACUM=ATENACUM+ATENLOSSES;
SPILLACUM= SPILLACUM+SPILLLOSSES;
CONTADORHORAS=CONTADORHORAS+1;

end %for
end %for
%hacemos las medias horarias (por helióstato) y multiplicamos por el
%número de horas anuales

% suponemos 2800h al año
HORASAN=2800;
MedhPOT=POTENCIAACUM./CONTADORHORAS;%.*HORASAN;
MedhCOEFF=COSEFFECTACUM./CONTADORHORAS;%.*HORASAN;
MedhSOMBterr=SOMBRATERACUM./CONTADORHORAS;%.*HORASAN;
MedhSOMB=SOMBRASACUM./CONTADORHORAS;%.*HORASAN;
MedhBLOCK=BLOQUEOSACUM./CONTADORHORAS;%.*HORASAN;
MedhATEN=ATENACUM./CONTADORHORAS;%.*HORASAN;
MedhSpill=SPILLACUM./CONTADORHORAS;%.*HORASAN;

TOTALPOT=sum(MedhPOT);

```

Función malla 3D

```

function[CAMPO3d,Area]=mallado3DMio(HT,LM,WM,XV,YV,Z0,LR,BL)
%
%
%XV=[0 -350 -350 0];% vertices X
%YV=[0 0 -1000 -1000];% vertices Y

%LM=8;           %ALTURA;
%WM=6;           %ANCHURA;
DIAGONAL=sqrt(LM.^2+WM.^2);
FA=1;           %RELACION (AREA DE REFLEXION/AREA TOTAL) , PARA UN HELIÓSTATO

%HT=75;          %THT
%Z0=5;           %PEDESTH;
%LR=2;           %ALTURA DEL RECEPTOR

%BL=0;           %PENDIENTE DEL TERRENO EN RADIANTES

r=size(XV); % r número de vértices que tenga el polígono

XVP=[XV,XV(:,1)];
YVP=[YV,YV(:,1)];

%hallamos datos geometricos del poligono XVN,YVN
Area=0;
Pendientes(1:r(2),1)=0;
OrdenadasOrigen(1:r(2),1)=0; for i=1:r(2) %para cada segmento de cada
helióstatto
    maxX(i)=max(XVP(i),XVP(i+1));
    minX(i)=min(XVP(i),XVP(i+1));
    maxY(i)=max(YVP(i),YVP(i+1));
    minY(i)=min(YVP(i),YVP(i+1));

    % para segmentos no verticales y no verticales
    Pendientes(i)=(YVP(i+1)-YVP(i))/(XVP(i+1)-XVP(i));
    OrdenadasOrigen(i)=YVP(i)-Pendientes(i)*XVP(i);
    a=XVP(i+1)-XVP(i);
    b=YVP(i+1)+YVP(i);
    Ar=a*b/2;
    Area=Area+Ar;

end %for i
Area=abs(Area);

%intentamos hallar el radio minimo para empezar a mallar
if(inpolygon(0,0,XV,YV)~=1 & inpolygon(0,0,XV,YV)~=0.5) %el centro
coordenadas esta estrictamente fuera del poligono normalizado
    t=size(XV);
    for i=1:t(2)

```

```

vectorRef1=[XVP(i) YVP(i)];
vectorRef2=[XVP(i+1) YVP(i+1)];
vectorLinea=[(XVP(i+1)-XVP(i)) (YVP(i+1)-YVP(i))];
distLinea=norm(vectorLinea);
vectorLineaNORM=vectorLinea./norm(vectorLinea);

RadioPuntosLinea=0;
RadioPuntosLinea(1)=norm(vectorRef1);
RadioPuntosLinea(floor(distLinea)+2)=norm(vectorRef2);
for j=2:(floor(distLinea)+1)
    PuntosLinea=vectorRef1+vectorLineaNORM.*(j-1);
    RadioPuntosLinea(j)=norm(PuntosLinea);
end

RadioMin(i)=min(RadioPuntosLinea);
end %for i
RadioMin=min(RadioMin);
if RadioMin<HT*0.75
    RadioMin=HT*0.75;%se recomienda un mínimo de 0.75 veces la altura de la
torre
end
else
    RadioMin=HT*0.75;
end

%hallamos el radio maximo
[th,R]=cart2pol(XV,YV);
RadioMax=max(R);

% STEP 1
F=WM/LM;
DSMin=2*F-sqrt(1+F.^2); %RELACION (DISTANCIA DE SEPARACION ENTRE
HELIOSRTATOS/ALTURA DE HELIÓSTATO)
%si ponemos esta formula sale 2*WM
DS=DSMin; % dejamos una distancia de separación entre helióstatos de
(DS/2)*LM

AM=F*FA*LM.^2;
%Z1=HT-0.5*LR;
Z1=HT;
RM=LM/2;

% STEP 2
DM=max(LM*(sqrt(1+F.^2)+DS), 2*WM);
ARmin=DM*cos(30*pi/180)*cos(BL);

% STEP 3
i=0;
j=0;

```

```

R=zeros(100);
Y=zeros(1,100);

%NPos=zeros(100);
%NNeg=zeros(100);
thetaMax=zeros(100);
thetaMin=zeros(100);
NRG=zeros(1,100);

R(i+1,j+1)=RadioMin;
Y(j+1)=asin(DM/(2*R(i+1,j+1)));
[thetaMax(i+1,j+1),thetaMin(i+1,j+1)]=rangoangular(Pendientes,OrdenadasOrigen
,maxX,maxY,minX,minY,R(i+1,j+1),XV,YV);
%NPos(i+1,j+1)=floor(thetaMax(i+1,j+1)/(2*Y(j+1)))+1; %metemos el del centro
en la rama positiva
%NNeg(i+1,j+1)=floor(-thetaMin(i+1,j+1)/(2*Y(j+1)));

termina=0;
% STEP 4
while termina==0;

    % STEP 5
    i=i+1;
    R(i+1,j+1)=R(i,j+1)*cos(Y(j+1))+sqrt((DM*cos(BL)).^2-
(R(i,j+1)*sin(Y(j+1))).^2);
    if(R(i+1,j+1)>RadioMax)
        termina=1;
        R(i+1,j+1)=0;
        i=NaN;
        break;
    end

[thetaMax(i+1,j+1),thetaMin(i+1,j+1)]=rangoangular(Pendientes,OrdenadasOrigen
,maxX,maxY,minX,minY,R(i+1,j+1),XV,YV);
    %NPos(i+1,j+1)=floor((thetaMax(i+1,j+1)-Y(j+1))/(2*Y(j+1)))+1;
    %NNeg(i+1,j+1)=floor((-thetaMin(i+1,j+1)-Y(j+1))/(2*Y(j+1)))+1;

% STEP 6
while (i~=0 & termina==0)

    % STEP 7
    ZM=Z0+R(i,j+1)*tan(BL);
    a=Z1.^2*(RM.^2-R(i,j+1).^2);
    b=2*R(i,j+1)*Z1*(Z1-ZM);
    c=RM.^2-(ZM-Z1).^2;
    YR=(-b+sqrt(b.^2-4*a*c))/(2*a);

    A=-((2*Z1*YR+tan(BL))*tan(BL)+Z1.^2*YR.^2);
    B=2*(Z1-Z0)*(Z1*YR+tan(BL));

```

```

C=RM.^2*(1+Z1.^2*YR.^2)-(Z1-Z0).^2;

YM2=max((-B-sqrt(B.^2-4*A*C))/(2*A),R(i+1,j+1)+ARmin);

[Wmax,Wmin]=rangoangular(Pendientes,OrdenadasOrigen,maxX,maxY,minX,minY,YM2,X
V,YV);

% STEP 8
if(ispar(i+1)==1)
    if(Wmax==0)
        NmPos=0;
        NmNeg=floor(-Wmin/(2*Y(j+1)))+1;
    elseif(Wmin==0)
        NmPos=floor(Wmax/(2*Y(j+1)))+1;
        NmNeg=0;
    else
        NmPos=floor(Wmax/(2*Y(j+1)))+1;%metemos el del centro en la rama
positiva
        NmNeg=floor(-Wmin/(2*Y(j+1)));
    end
else
    if(Wmax==0)
        NmPos=0;
        NmNeg=floor((-Wmin-Y(j+1))/(2*Y(j+1)))+1;
    elseif(Wmin==0)
        NmPos=floor((Wmax-Y(j+1))/(2*Y(j+1)))+1;
        NmNeg=0;
    else
        NmPos=floor((Wmax-Y(j+1))/(2*Y(j+1)))+1;
        NmNeg=floor((-Wmin-Y(j+1))/(2*Y(j+1)))+1;
    end
end

% STEP 9
AF=Wmax/2*((YM2+0.5*DM).^2-(R(i+1,j+1)+0.5*DM).^2)+(-
Wmin/2)*((YM2+0.5*DM).^2-(R(i+1,j+1)+0.5*DM).^2);
DELTA=(NmPos+NmNeg)*AM/AF;

% STEP 10
ZM=Z0+R(i+1,j+1)*tan(BL);
a=Z1.^2*(RM.^2-R(i+1,j+1).^2);
b=2*R(i+1,j+1)*Z1*(Z1-ZM);
c=RM.^2-(ZM-Z1).^2;
YR=(-b+sqrt(b.^2-4*a*c))/(2*a);

A=-((2*Z1*YR+tan(BL))*tan(BL)+Z1.^2*YR.^2);
B=2*(Z1-Z0)*(Z1*YR+tan(BL));
C=RM.^2*(1+Z1.^2*YR.^2)-(Z1-Z0).^2;

YM2grupo=max((-B-sqrt(B.^2-4*A*C))/(2*A),R(i+1,j+1)+ARmin);

```

```
[Wmaxgrupo,Wmingrupo]=rangoangular(Pendientes,OrdenadasOrigen,maxX,maxY,minX,
minY,YM2grupo,XV,YV);
```

```
% STEP 11
Y(j+2)=DM/(2*YM2grupo);

if(Wmaxgrupo==0)
    NmPosgrupo=0;
    NmNeggrupo=floor(-Wmingrupo/(2*Y(j+2)))+1;
elseif(Wmingrupo==0)
    NmPosgrupo=floor(Wmaxgrupo/(2*Y(j+2)))+1;
    NmNeggrupo=0;
else
    NmPosgrupo=floor(Wmaxgrupo/(2*Y(j+2)))+1;%metemos el del centro en
la rama positiva
    NmNeggrupo=floor(-Wmingrupo/(2*Y(j+2)));
end

AFgrupo=Wmaxgrupo/2*((YM2grupo+0.5*DM).^2-(R(i+1,j+1)+0.5*DM).^2)+(-
Wmingrupo/2)*((YM2grupo+0.5*DM).^2-(R(i+1,j+1)+0.5*DM).^2);
DELTAgrupo=(NmPosgrupo+NmNeggrupo)*AM/AFgrupo;

% STEP 12

if (YM2<=RadioMax & YM2grupo<=RadioMax)
    if DELTA >= DELTAgrupo
        i=i+1;
        R(i+1,j+1)=YM2;
        %NPos(i+1,j+1)=NmPos;
        %NNeg(i+1,j+1)=NmNeg;
        thetaMax(i+1,j+1)=Wmax;
        thetaMin(i+1,j+1)=Wmin;
    else
        NRG(j+1)=i+1;
        j=j+1;
        i=0;

        R(i+1,j+1)=YM2grupo;
        %NPos(i+1,j+1)=NmPosgrupo;
        %NNeg(i+1,j+1)=NmNeggrupo;
        thetaMax(i+1,j+1)=Wmaxgrupo;
        thetaMin(i+1,j+1)=Wmingrupo;
    end

elseif (YM2<=RadioMax & YM2grupo>RadioMax)
    i=i+1;
    R(i+1,j+1)=YM2;
    %NPos(i+1,j+1)=NmPos;
    %NNeg(i+1,j+1)=NmNeg;
    thetaMax(i+1,j+1)=Wmax;
    thetaMin(i+1,j+1)=Wmin;
else (YM2>RadioMax & YM2grupo>RadioMax) %se acabo lo que se daba, fin
del algoritmo
```

```
        termina=1;
    end

    end % while step 6

end % while step 4

% STEP 13

if isnan(i)==1
    NG=j+1;
    NRG(j+1)=1;
elseif i~=0
    NG=j+1;
    NRG(j+1)=i+1;
end

m=1;

x=zeros(1000,1);
y=zeros(1000,1);
z=zeros(1000,1);

for j=1:NG
    for i=1:NRG(j)
        if ispar(i-1)==1
            k=1;
            x(m)=0;
            y(m)=-R(i,j);
            z(m)=Z0+R(i,j)*tan(BL);
            m=m+1;

            while thetaMax(i,j)>=Y(j)*k*2
                y(m)=sin(Y(j)*k*2-pi/2)*R(i,j);
                x(m)=cos(Y(j)*k*2-pi/2)*R(i,j);
                z(m)=Z0+R(i,j)*tan(BL);
                m=m+1;
                k=k+1;
            end

            k=1;

            while thetaMin(i,j)<=-Y(j)*k*2
                y(m)=sin(-Y(j)*k*2-pi/2)*R(i,j);
                x(m)=cos(-Y(j)*k*2-pi/2)*R(i,j);
                z(m)=Z0+R(i,j)*tan(BL);
                m=m+1;
                k=k+1;
            end

        else
            k=0;
        end
    end
end
```

```
while thetaMax(i,j)>=Y(j)+Y(j)*k*2
    y(m)=sin(Y(j)+Y(j)*k*2-pi/2)*R(i,j);
    x(m)=cos(Y(j)+Y(j)*k*2-pi/2)*R(i,j);
    z(m)=Z0+R(i,j)*tan(BL);
    m=m+1;
    k=k+1;
end

k=0;

while thetaMin(i,j)<=-(Y(j)+Y(j)*k*2)
    y(m)=sin(-Y(j)-Y(j)*k*2-pi/2)*R(i,j);
    x(m)=cos(-Y(j)-Y(j)*k*2-pi/2)*R(i,j);
    z(m)=Z0+R(i,j)*tan(BL);
    m=m+1;
    k=k+1;
end

end %else

end %for i

end %for j

%plot(x,y,'o',XVP,YVP);
%axis equal;

IN1=inpolygon(x,y,XV,YV);
X=x.*ceil(IN1);
Y=y.*ceil(IN1);
Z=z.*ceil(IN1);

%plot(X,Y,'o',XVP,YVP);
%axis equal;

[CAMP03d,POSICIONX12]=ordenaX1(X,Y,Z);

%plot(CAMP03d(:,1),CAMP03d(:,2),'o',XVP,YVP);
%axis equal;
```

