

ANEXOS

1. Función Matlab - Cálculo de Ángulos Solares

```
% FUNCION DE CALCULO DE ANGULOS SOLARES CARACTERISTICOS- PRECISION 2
function
[Solar_Time,omega,theta,theta_z,gamma_s,alpha_s,Hora_sale_sol,Hora_pone_
sol] = Ang_solar_f7_3( mm,dd )

% Declaración de datos básicos
Lst=75;           % Longitud estándar (grados)
Lloc=75.083;     % Longitud local (grados)
Md=1;           % Oeste=1, Este=2
phi=13.347;     % Latitud (grados)
Ltns=1;         % Sur=1, Norte=2
beta=0;         % Angulo inclinación plano S-N (grados)
gamma=90;       % Angulo azimuth plano (grados)
St=0;          % Vector para almacenar hora local

% Cálculo de n
m_dd=[dd 31+dd 59+dd 90+dd 120+dd 151+dd 181+dd 212+dd 243+dd 273+dd
304+dd 334+dd];
n=m_dd(mm);

% Radiación extraterrestre
Gsc=1367; %W/m2
B=(n-1)*(360/365)*pi/180;

Gon=Gsc*(1.000110+0.034221*cos(B)+0.001280*sin(B)+0.000719*cos(2*B)+
0.000077*sin(2*B));

% Declinación solar
if Ltns==1
    delta=(180/pi)*(0.006918-0.399912*cos(B)+0.070257*sin(B)-
    0.006758*cos(2*B)+0.000907*sin(2*B)-
    0.002697*cos(3*B)+0.00148*sin(3*B));
end
if Ltns==2
    delta=-(180/pi)*(0.006918-0.399912*cos(B)+0.070257*sin(B)-
    0.006758*cos(2*B)+0.000907*sin(2*B)-
    0.002697*cos(3*B)+0.00148*sin(3*B));
end

% Hora solar
for i=1:49
    St(i)=0.25*i+5.75; % Rango de hora local

    E=229.2*(0.000075+0.001868*cos(B)-0.032077*sin(B)-
    0.014615*cos(2*B)-0.04089*sin(2*B)); % minutos

    if Md==1;
        Solar_Time= St+(4*(Lst-Lloc)+E)/60;
    end
    if Md==2;
```

```

        Solar_Time= St+(4*(Lloc-Lst)+E)/60;
    end

    % Cálculo de ángulos principales
    omega=(12-Solar_Time)*15;
    omega_s=acos(-tan(phi*pi/180)*tan(delta*pi/180))*(180/pi);
    Hora_sale_sol=12-(omega_s/15);
    Hora_pone_sol=12+(omega_s/15);
    cos_theta=sin(delta*pi/180)*sin(phi*pi/180)*cos(beta*pi/180)-
    sin(delta*pi/180)*cos(phi*pi/180)*sin(beta*pi/180)*cos(gamma*pi/
    180)+cos(delta*pi/180)*cos(phi*pi/180)*cos(beta*pi/180)*cos(omeg
    a*pi/180)+cos(delta*pi/180)*sin(phi*pi/180)*sin(beta*pi/180)*cos
    (gamma*pi/180)*cos(omega*pi/180)+cos(delta*pi/180)*sin(beta*pi/1
    80)*sin(gamma*pi/180)*sin(omega*pi/180);
    theta=acos(cos_theta)*180/pi;

    cos_theta_z=cos(phi*pi/180)*cos(delta*pi/180)*cos(omega*pi/180)+
    sin(phi*pi/180)*sin(delta*pi/180);
    theta_z=acos(cos_theta_z)*180/pi;
    alpha_s=90-theta_z;
    k=(cos(theta_z*pi/180)*sin(phi*pi/180)-
    sin(delta*pi/180))./(sin(theta_z*pi/180)*cos(phi*pi/180));
    gamma_s=sign(omega).*abs(acos(k)).*180/pi;
end
end

```

2. Script Matlab – Cálculo de Radiaciones Modelo Anisotrópico

```

% SCRIPT DE CALCULO DE RADIACIONES - MODELO ANISOTROPICO HDKR
% Precisión 2 - Modelo final
clc
clear all

% Declaración de datos
mm=12;           % Mes
dd=21;           % Día
Lst=75;          % Longitud estándar (grados)
Lloc=75.083;     % Longitud local (grados)
Md=1;           % Oeste=1, Este=2
phi=13.347;      % Latitud (grados)
Ltns=1;          % Sur=1, Norte=2
betan=0;         % Angulo giro (grados) (+=antihorario, -=horario)
gamma=90;        % Angulo azimuth plano (grados)
St=0;           % Hora local

%=====
% CALCULO DE ANGULOS CARACTERISTICOS
%=====

% Cálculo de n (nº día del año)
m_dd=[dd 31+dd 59+dd 90+dd 120+dd 151+dd 181+dd 212+dd 243+dd 273+dd
304+dd 334+dd];
n=m_dd(mm);

% Radiación extraterrestre
Gsc=1367; %W/m2
B=(n-1)*(360/365)*pi/180;

```

```

Gon=Gsc*(1.000110+0.034221*cos(B)+0.001280*sin(B)+0.000719*cos(2*B)+0.00
0077*sin(2*B));

% Declinación solar
if Ltns==1
    delta=(180/pi)*(0.006918-0.399912*cos(B)+0.070257*sin(B)-
0.006758*cos(2*B)+0.000907*sin(2*B)-0.002697*cos(3*B)+0.00148*sin(3*B));
end
if Ltns==2
    delta=-(180/pi)*(0.006918-0.399912*cos(B)+0.070257*sin(B)-
0.006758*cos(2*B)+0.000907*sin(2*B)-0.002697*cos(3*B)+0.00148*sin(3*B));
end

% Hora solar
for i=1:49
    St(i)=0.25*i+5.75; % Rango de hora local

    E=229.2*(0.000075+0.001868*cos(B)-0.032077*sin(B)-0.014615*cos(2*B)-
0.04089*sin(2*B)); % minutos

    if Md==1;
        Solar_Time= St+(4*(Lst-Lloc)+E)/60;
    end
    if Md==2;
        Solar_Time= St+(4*(Lloc-Lst)+E)/60;
    end

% Cálculo de ángulos principales
    omega=-(12-Solar_Time)*15;
    omega_s=acos(-tan(phi*pi/180)*tan(delta*pi/180))*(180/pi);
    Hora_sale_sol=12-(omega_s/15);
    Hora_pone_sol=12+(omega_s/15);

    cos_theta=sin(delta*pi/180)*sin(phi*pi/180)*cos(betan*pi/180)-
sin(delta*pi/180)*cos(phi*pi/180)*sin(betan*pi/180)*cos(gamma*p
i/180)+cos(delta*pi/180)*cos(phi*pi/180)*cos(betan*pi/180)*cos(
omega*pi/180)+cos(delta*pi/180)*sin(phi*pi/180)*sin(betan*pi/18
0)*cos(gamma*pi/180)*cos(omega*pi/180)+cos(delta*pi/180)*sin(be
tan*pi/180)*sin(gamma*pi/180)*sin(omega*pi/180);

    cos_theta_z=cos(phi*pi/180)*cos(delta*pi/180)*cos(omega*pi/180)
+sin(phi*pi/180)*sin(delta*pi/180);
    theta_z=acos(cos_theta_z)*180/pi;
    alpha_s=90-theta_z;
    k=(cos(theta_z*pi/180)*sin(phi*pi/180)-
sin(delta*pi/180))./(sin(theta_z*pi/180)*cos(phi*pi/180));
    gamma_s=sign(omega).*abs(acos(k)).*180/pi;
    theta=acos(cos_theta)*180/pi;

end

%=====
% CALCULO DE LA RADIACIONES
%=====
Solar_Time1=Solar_Time-0.5;
Solar_Time2=Solar_Time+0.5;

```

```

% Radiación Extraterrestre en un periodo horario sobre una superficie
horizontal
omega1=-(12-Solar_Time1)*15;
omega2=-(12-Solar_Time2)*15;
Io=(12*3600*Gsc/pi).*(1+0.033*cos(360*(n/365)*pi/180)).*(cos(phi*pi/180)
*cos(delta*pi/180)*(sin(omega2*pi/180)-sin(omega1*pi/180))+pi*(omega2-
omega1)/180)*sin(phi*pi/180)*sin(delta*pi/180))*1e-6;

% Distribución de Radiación diaria a horaria
m_rad_dd=[18 19.8 21.6 19.8 19.8 18 18 18 19.8 21.6 21.6 18]; % Dato de
registro Atlas Solar del Perú (SENAMHI) (MJ/m2)
H=1.35*m_rad_dd(mm);

% Cálculo de rt
ah=0.409+0.5016*sin(omega_s*pi/180-60);
bh=0.6609-0.4767*sin(omega_s*pi/180-60);
rt=pi/24*(ah+bh*cos(omega*pi/180)).*(cos(omega*pi/180)-
cos(omega_s*pi/180))./(sin(omega_s*pi/180)-
(pi*omega_s/180)*cos(omega_s*pi/180));

I=rt.*H;

Kt=I./Io;

for i=1:49
    if Kt(i)<=0.22
        Id(i)=I(i).*(1-0.09*Kt(i));
    end
    if Kt(i)>0.22 && Kt(i)<=0.8
        Id(i)=I(i).*(0.9511-0.1604*Kt(i)+4.388*Kt(i).^2-
16.638*Kt(i).^3+12.336*Kt(i).^4);
    end
    if Kt(i)>0.8
        Id(i)=I(i)*0.165;
    end
end

% Cálculo de Rb
Rb=cos(theta*pi/180)./cos(theta_z*pi/180);

% Cálculo de It
rho_g=0.25; % Reflectancia del suelo (introducir dato experimental)

Ib=I-Id;
Ai=Ib./Io;
f=Ib./I;

It=(Ib+Id.*Ai).*Rb+Id.*(1-
Ai).*((1+cos(betan*pi/180))/2).*(1+f*sin(0.5*betan*pi/180)^3+I.*rho_g*((
1-cos(betan*pi/180))/2));

% Cálculo de radiaciones
Rad_direc=Ib.*Rb;
Rad_dif_cirsolar=Id.*Ai.*Rb;
Rad_dif_Iscielo=Id.*(1-
Ai).*((1+cos(betan*pi/180))/2).*(1+f*sin(0.5*betan*pi/180)^3);
Rad_dif_Issuelo=I*rho_g*((1-cos(betan*pi/180))/2);

```

```

Solar_Time
alpha_s
gamma_s
theta_z
theta
Rb
Io
I
rt
Kt
Ib
Id
It
Rad_direc
Rad_dif_cirsolar
Rad_dif_Iscielo
Rad_dif_Issuelo

% Gráficos
plot(Solar_Time, I, 'bo:')
hold on
plot (Solar_Time, Rad_direc, 'ro-')
plot (Solar_Time, Rad_dif_cirsolar, 'co-')
plot (Solar_Time, Rad_dif_Iscielo, 'go-')
plot (Solar_Time, Rad_dif_Issuelo, 'mo-')
plot (Solar_Time, It, 'b*-')
grid on
xlabel('Hora Solar (hr)', 'FontSize',14);
ylabel('Radiación solar (Mj/m2)', 'FontSize',14);
title('Radiación horaria - Modelo Anisotrópico- Pilpichaca
Huancavelica', 'FontSize',16);
legend('Irradiancia horaria', 'Radiación directa', 'Radiación difusa
circunsolar', 'Radiación difusa cielo', 'Radiación difusa
suelo', 'Radiación total',2);

```

3. Script Matlab - Perfilamiento óptimo de CPC1T

```

% =====
% DIMENSIONAMIENTO DE CONCENTRADOR PARABOLICO COMPUESTO (CPC)
% MODELO FINAL
% PERFIL PARA TUBERIA DE ABSORCION UNICA, ESTANDAR
% =====

clc
clear all
% Declaración de variables
d_v=0.125;           % Diámetro exterior tubo de vidrio al vacío
d_r=0.075;           % Diámetro exterior tubo receptor
deltag=0.03;         % Tolerancia generatriz
d_g=d_v+deltag;     % Diámetro tubo de vidrio + tolerancia (generatriz)
L=50;                % Longitud útil del sistema colector

% Concentración geométrica CLF
% Cg=1/sen(theta_a);
% De resultados de investigaciones (Tiba y Fraidenraich, 2011)
% theta_a óptimo cumple: 1.6<=Cg<=2 entonces:
Cg=[1.6 1.8 2.0];
theta_am=asin(1./Cg);

```

```

theta_a=theta_am(1);

r_g=d_g/2;      % Radio generatriz de curvas

% Límites de phi
phimax=1.5*pi-theta_a;
phimin=0.5*pi+theta_a;

% Ecuaciones para involuta en el plano X-Y
phil=0:0.01:(0.5*pi+theta_a);
rho1=r_g.*phil;
x1=r_g*sin(phil)-rho1.*cos(phil);
y1=-r_g*cos(phil)-rho1.*sin(phil);

% Ecuación para la sección parabólica plano X-Y
phi2=0.5*pi+theta_a:0.01:(1.5*pi-theta_a);

% Distancia entre la tangente del tubo colector y el inicio de la
parábola
a2=(0.5*pi+theta_a+phi2-cos(phi2-theta_a))./(1+sin(phi2-theta_a));
rho2=r_g.*a2;

x2=r_g*sin(phi2)-rho2.*cos(phi2);
y2=-r_g*cos(phi2)-rho2.*sin(phi2);

% Ecuaciones para graficar tubo de vidrio y tubo colector
theta=0:0.125*pi:2*pi;
x3=d_v/2*cos(theta);
y3=d_v/2*sin(theta);
x4=d_r/2*cos(theta);
y4=d_r/2*sin(theta);

% Ecuaciones para graficar radio generatriz
theta=0:0.125*pi:2*pi;
x5=r_g*cos(theta);
y5=r_g*sin(theta);

% Area de la superficie reflectante CPC
n1=(0.5*pi+theta_a-0)/(0.1);
sum1=0;
for i=1:n1
    A_r1(i)=sqrt(x1(i)^2+y1(i)^2);
    sum_1=sum1+A_r1(i);
end

n2=((1.5*pi-theta_a)-(0.5*pi+theta_a-0))/(0.1);
sum2=0;
for i=1:n2
    A_r2(i)=sqrt(x2(i)^2+y2(i)^2);
    sum_2=sum2+A_r2(i);
end

A_r=2*(sum_1+sum_2)*L; % m2

% Area de tubo colector
A_a=pi*d_r*L; % m2

```

```

x6=-0.3869:0.001:0.1695;
n6=floor((0.3869-0.1695)/0.01+1);
for i=1:n6
    y6=-1.249*x6+0.1224;
end
x7=-0.1695:0.001:0.3869;
n7=floor((0.3869-0.1695)/0.01+1);
for i=1:n7
    y7=1.249*x7+0.1224;
end

% Cálculo de dimensiones básicas
H_icpc=abs(min(y1))+abs(max(y2)); % m
Wc=max(x2)*2; % m
disp('El ángulo de aceptación es: ')
theta_a
disp('La altura interna del CPC es: ')
H_icpc
disp('La abertura del CPC es: ')
Wc
disp('El área del tubo colector es: ')
A_a
disp('El área de la superficie reflectante CPC es: ')
A_r

% Gráfico de Geometría de concentrador secundario CPC
plot(x1,y1,'b-')
hold on
plot(x2,y2,'r-')
% plot(x3,y3,'k-')
% plot(x4,y4,'c-')
plot(x6,y6,'m-')
plot(-x1,y1,'b-')
plot(-x2,y2,'r-')
plot(x7,y7,'m-')
plot(x5,y5,'k-')
grid on
xlabel('X (m)', 'FontSize',14);
ylabel('Y (m)', 'FontSize',14);
title('Sección Geométrica CPC', 'FontSize',16);
legend('Involuta', 'Parábola', 'Tangente límite',0);

```

4. Script Matlab - Perfilamiento óptimo de CPC2T

```

% =====
% DIMENSIONAMIENTO DE CONCENTRADOR PARABOLICO COMPUESTO (CPC)
% MODELO FINAL
% PERFIL PARA DOBLE TUBERIA DE ABSORCION
% =====

clc
clear all
% Declaración de variables
d_v=0.125*2; % Diámetro exterior tubo de vidrio al vacío
límite
d_r=0.075*2; % Diámetro exterior tubo receptor límite
deltag=0.00; % Tolerancia generatriz límite
d_g=d_v+deltag; % Diámetro tubo de vidrio + tolerancia
(generatriz)

```

```

L=50; % Longitud útil del sistema colector
d_v_1=0.125; % Diámetro exterior tubo de vidrio al vacío
d_r_1=0.075; % Diámetro exterior tubo receptor

% Concentración geométrica CLF
% Cg=1/sen(theta_a);
% De resultados de investigaciones (Tiba y Fraidenraich, 2011) theta_a
óptimo cumple: 1.6<=Cg<=2 entonces:
Cg=[1.6 1.8 2.0];
theta_am=asin(1./Cg);
theta_a=theta_am(3);

r_g=d_g/2; % Radio generatriz de curvas

% Límites de phi
phimax=1.5*pi-theta_a;
phimin=0.5*pi+theta_a;

% Ecuaciones para involuta en el plano X-Y
phi1=0:0.1:(0.5*pi+theta_a);
rho1=r_g.*phi1;
x1=r_g*sin(phi1)-rho1.*cos(phi1);
y1=-r_g*cos(phi1)-rho1.*sin(phi1);

% Ecuación para la sección parabólica plano X-Y
phi2=0.5*pi+theta_a:0.1:(1.5*pi-theta_a);

% Distancia entre la tangente del tubo colector y el inicio de la
parábola
a2=(0.5*pi+theta_a+phi2-cos(phi2-theta_a))./(1+sin(phi2-theta_a));
rho2=r_g.*a2;

x2=r_g*sin(phi2)-rho2.*cos(phi2);
y2=-r_g*cos(phi2)-rho2.*sin(phi2);

% Ecuaciones para graficar tubo colector aparente
theta=0:0.125*pi:2*pi;
x3=d_v/2*cos(theta);
y3=d_v/2*sin(theta);
x4=d_r/2*cos(theta);
y4=d_r/2*sin(theta);

% Ecuaciones para graficar tubo de vidrio y tubo colector 1
theta_1=0:0.0625*pi:2*pi;
x7=0.5*d_v_1+d_v_1/2*cos(theta_1);
y7=d_v_1/2*sin(theta_1);
x6=0.5*d_v_1+d_r_1/2*cos(theta_1);
y6=d_r_1/2*sin(theta_1);

% Ecuaciones para graficar tubo de vidrio y tubo colector 2
theta_1=0:0.0625*pi:2*pi;
x8=-(0.5*d_v_1+d_v_1/2*cos(theta_1));
y8=d_v_1/2*sin(theta_1);
x9=-(0.5*d_v_1+d_r_1/2*cos(theta_1));
y9=d_r_1/2*sin(theta_1);

% Ecuaciones para graficar radio generatriz
theta=0:0.125*pi:2*pi;

```



```

x5=r_g*cos(theta);
y5=r_g*sin(theta);

% Area de la superficie reflectante CPC
n1=(0.5*pi+theta_a-0)/(0.1);
sum1=0;
for i=1:n1
    A_r1(i)=sqrt(x1(i)^2+y1(i)^2);
    sum_1=sum1+A_r1(i);
end

n2=((1.5*pi-theta_a)-(0.5*pi+theta_a-0))/(0.1);
sum2=0;
for i=1:n2
    A_r2(i)=sqrt(x2(i)^2+y2(i)^2);
    sum_2=sum2+A_r2(i);
end

A_r=2*(sum_1+sum_2)*L; % m2

% Area de tubo colector
A_a=pi*d_r*L; % m2

% Cálculo de dimensiones básicas
H_icpc=abs(min(y1))+abs(max(y2)); % m
Wc=max(x2)*2; % m
disp('El ángulo de aceptación es: ')
theta_a
disp('La altura interna del CPC es: ')
H_icpc
disp('La abertura del CPC es: ')
Wc
disp('El área del tubo colector es: ')
A_a
disp('El área de la superficie reflectante CPC es: ')
A_r

% Gráfico de Geometría de concentrador secundario CPC
plot(x1,y1,'b*-')
hold on
plot(x2,y2,'r*-')
% plot(x3,y3,'c-')
% plot(x4,y4,'c-')
plot(x7,y7,'b-')
plot(x6,y6,'k-')
plot(x5,y5,'m-')
plot(x8,y8,'b-')
plot(x9,y9,'k-')
plot(-x1,y1,'b*-')
plot(-x2,y2,'r*-')
grid on
xlabel('X (m)', 'FontSize',14);
ylabel('Y (m)', 'FontSize',14);
title('Medio Perfil Geométrico CPC', 'FontSize',16);
legend('Involuta', 'Parábola', 'Sección tubo vidrio al vacío', 'Sección
tubo receptor', 'Diámetro de generatriz',2);

```

5. Script Matlab - Seguimiento Solar

```
% =====  
% SEGUIMIENTO SOLAR, PASO Y SEPARACIÓN ENTRE ESPEJOS LFC FRESNEL  
% MODELO FINAL  
% Dimensiones 24 x 14 m.  
% =====  
% Cálculo de ángulos solares  
clc  
clear all  
  
% Declaración de datos básicos  
mm=3;           % Mes  
dd=21;          % Día  
Lst=75;         % Longitud  
Lloc=75.083;    % Longitud Estándar  
Md=1;          % Oeste=1, Este=2  
phi=13.347;     % Latitud  
Ltns=1;         % Sur=1, Norte=2  
beta=0;         % Angulo inclinación plano  
gamma=90;       % Angulo azimuth plano  
St=0;          % Hora local  
Horacontrol=12; % Hora para angulación de espejos  
  
% Cálculo de n (n° día del año)  
m_dd=[dd 31+dd 59+dd 90+dd 120+dd 151+dd 181+dd 212+dd 243+dd 273+dd  
304+dd 334+dd];  
n=m_dd(mm);  
  
% Radiación extraterrestre  
Gsc=1367; %W/m2  
B=(n-1)*(360/365)*pi/180;  
Gon=Gsc*(1.000110+0.034221*cos(B)+0.001280*sin(B)+0.000719*cos(2*B)+0.00  
0077*sin(2*B));  
  
% Declinación solar  
if Ltns==1  
    delta=(180/pi)*(0.006918-0.399912*cos(B)+0.070257*sin(B)-  
0.006758*cos(2*B)+0.000907*sin(2*B)-0.002697*cos(3*B)+0.00148*sin(3*B));  
end  
if Ltns==2  
    delta=-(180/pi)*(0.006918-0.399912*cos(B)+0.070257*sin(B)-  
0.006758*cos(2*B)+0.000907*sin(2*B)-0.002697*cos(3*B)+0.00148*sin(3*B));  
end  
  
% Hora solar  
for i=1:13  
    St(i)=5+i;           % Rango de hora local  
  
    E=229.2*(0.000075+0.001868*cos(B)-0.032077*sin(B)-0.014615*cos(2*B)-  
0.04089*sin(2*B)); % minutos  
  
    if Md==1;  
        Solar_Time= St+(4*(Lst-Lloc)+E)/60;  
    end  
    if Md==2;
```

```

        Solar_Time= St+(4*(Lloc-Lst)+E)/60;
    end

% Cálculo de ángulos principales
    omega=- (12-Solar_Time)*15;
    omega_s=acos(-tan(phi*pi/180)*tan(delta*pi/180))*(180/pi);
    Hora_sale_sol=12-(omega_s/15);
    Hora_pone_sol=12+(omega_s/15);
    cos_theta=sin(delta*pi/180)*sin(phi*pi/180)*cos(beta*pi/180)-
    sin(delta*pi/180)*cos(phi*pi/180)*sin(beta*pi/180)*cos(gamma*pi/180)
    +cos(delta*pi/180)*cos(phi*pi/180)*cos(beta*pi/180)*cos(omega*pi/180)
    +cos(delta*pi/180)*sin(phi*pi/180)*sin(beta*pi/180)*cos(gamma*pi/180)
    *cos(omega*pi/180)+cos(delta*pi/180)*sin(beta*pi/180)*sin(gamma*pi
    /180)*sin(omega*pi/180);
    theta=acos(cos_theta)*180/pi;

    cos_theta_z=cos(phi*pi/180)*cos(delta*pi/180)*cos(omega*pi/180)+sin(
    phi*pi/180)*sin(delta*pi/180);
    theta_z=acos(cos_theta_z)*180/pi;
    alpha_s=90-theta_z;

    k=(cos(theta_z*pi/180)*sin(phi*pi/180)*sin(delta*pi/180))./(sin(theta
    _z*pi/180)*cos(phi*pi/180));
    gamma_s=sign(omega).*abs(acos(k)).*180/pi;

end
Solar_Time
gamma_s
alpha_s

% Cálculo de ángulos de formación

theta_p=atan(tan(alpha_s*pi/180)./cos((gamma_s-90)*pi/180))*180/pi; %
(Nixon y Davis 2012).
theta_t=90-theta_p;

% Datos de Configuración de Heliostatos
L=10.0;           % Amplitud de campo de heliostatos
s=0.8;           % Distancia entre heliostatos
n_h=fix((L-1)/s+1); % Número de heliostatos por flanco Este/Oeste
d_o=0.8;         % Distancia del primer heliostato al CPC
Hc=14;           % Altura de heliostatos al CPC
W=0.5;           % ancho de heliostato típico
theta_ne=zeros(n_h,13);
theta_no=zeros(n_h,13);

% Cálculo de ángulos heliostatos lado Este

Q_ne=d_o:s:L;     % Distancia horizontal de heliostatos al CPC
beta_n=(180/pi)*atan(Q_ne/Hc); % ángulo de proyección del CPC

for j=1:n_h
    for i=1:13
        if theta_p(i)<0
            theta_ne(j,i)=- (theta_p(i)+(90-beta_n(j)))/2;
        end
        if theta_p(i)>=0
            theta_ne(j,i)=((180-theta_p(i))-(90-beta_n(j)))/2;
        end
    end
end

```

```

        end
    end
end

theta_ne

% Cálculo de ángulos heliostatos lado Oeste

Q_no=-Q_ne; % Distancia horizontal de heliostatos oeste al CPC

beta_n=(180/pi)*atan(Q_no/Hc); % Angulo de proyección del CPC

for j=1:n_h
    for i=1:13
        if theta_p(i)<0
            theta_no(j,i)=-(theta_p(i)+(90-beta_n(j)))/2;
        end
        if theta_p(i)>=0
            theta_no(j,i)=((180-theta_p(i))-(90-beta_n(j)))/2;
        end
    end
end

theta_no

theta_p

% Revisión de sombras entre espejos consecutivos
% Cálculo de la distancia adecuada para evitar sombreado:

% Verificación lado Este
for b=1:13;
    for a=1:n_h-1

S_ne(a,b)=(0.5*W*(abs(sin(theta_ne(a,b)*pi/180))+abs(sin(theta_ne(a+1,b)
*pi/180))))./abs(tan((theta_p(1,b)-0.27)*pi/180));

P_ne(a,b)=(0.5*W*(abs(sin(theta_ne(a,b)*pi/180))+abs(sin(theta_ne(a+1,b)
*pi/180))))./abs(tan((theta_p(1,b)-
0.27)*pi/180))+0.5*W*(abs(cos(theta_ne(a,b)*pi/180))+abs(cos(theta_ne(a+
1,b)*pi/180))));
        end
    end
P_ne
S_ne
disp('La mínima separación sin sombra entre heliostatos lado Este es: ')
MinimaSep_E=max(S_ne)

% Verificación lado Oeste
for b=1:13;
    for a=1:n_h-1

S_no(a,b)=(0.5*W*(abs(sin(theta_no(a,b)*pi/180))+abs(sin(theta_no(a+1,b)
*pi/180))))./abs(tan((theta_p(1,b)-0.27)*pi/180));

P_no(a,b)=(0.5*W*(abs(sin(theta_no(a,b)*pi/180))+abs(sin(theta_no(a+1,b)
*pi/180))))./abs(tan((theta_p(1,b)-

```

```

0.27)*pi/180))+0.5*W*(abs(cos(theta_no(a,b)*pi/180))+abs(cos(theta_no(a+
1,b)*pi/180)));
    end
end
P_no
S_no
disp('La mínima separación sin sombra entre heliostatos lado Oeste es:
')
MinimaSep_O=max(S_no)

% Cálculo de la abertura de CPC
R=0.5*W; % radio de giro de heliostato típico
theta_k=atan(Hc/(d_o+s*(n_h-1)))*180/pi;
theta_pr=180-max(max(abs(theta_no)))-theta_k;
H_pr=R*sin(theta_pr*pi/180);
Wc=2*H_pr/sin(theta_k*pi/180);
disp('La mínima abertura del CPC debe ser:')
Wc

% Preparación de gráficos
% Rayos reflejados ideales hacia el colector CPC

Yh=[Hc 0];

X1e=[0 Q_ne(1)];
X2e=[0 Q_ne(2)];
X3e=[0 Q_ne(3)];
X4e=[0 Q_ne(4)];
X5e=[0 Q_ne(5)];
X6e=[0 Q_ne(6)];
X7e=[0 Q_ne(7)];
X8e=[0 Q_ne(8)];
X9e=[0 Q_ne(9)];
X10e=[0 Q_ne(10)];
X11e=[0 Q_ne(11)];
X12e=[0 Q_ne(12)];

X1o=[0 Q_no(1)];
X2o=[0 Q_no(2)];
X3o=[0 Q_no(3)];
X4o=[0 Q_no(4)];
X5o=[0 Q_no(5)];
X6o=[0 Q_no(6)];
X7o=[0 Q_no(7)];
X8o=[0 Q_no(8)];
X9o=[0 Q_no(9)];
X10o=[0 Q_no(10)];
X11o=[0 Q_no(11)];
X12o=[0 Q_no(12)];

% Orientación de heliostatos
% Heliostatos Lado Este

ff=Horacontrol-5;

Xe11=zeros(n_h,13);
Ye11=zeros(n_h,13);
Xe12=zeros(n_h,13);

```

```

Ye12=zeros(n_h,13);

for u=1:13
    for m=1:n_h
        Xe11(m,u)=Q_ne(m)+R*cos(theta_ne(m,u)*pi/180);
    end
end
Ye11=R*sin(theta_ne*pi/180);

Xe11;
Ye11;

for u=1:13
    for m=1:n_h
        Xe12(m,u)=Q_ne(m)-R*cos(theta_ne(m,u)*pi/180);
    end
end
Ye12=-R*sin(theta_ne*pi/180);

Xe12;
Ye12;

XE1=[Xe11(1,ff) Xe12(1,ff)];
YE1=[Ye11(1,ff) Ye12(1,ff)];
XE2=[Xe11(2,ff) Xe12(2,ff)];
YE2=[Ye11(2,ff) Ye12(2,ff)];
XE3=[Xe11(3,ff) Xe12(3,ff)];
YE3=[Ye11(3,ff) Ye12(3,ff)];
XE4=[Xe11(4,ff) Xe12(4,ff)];
YE4=[Ye11(4,ff) Ye12(4,ff)];
XE5=[Xe11(5,ff) Xe12(5,ff)];
YE5=[Ye11(5,ff) Ye12(5,ff)];
XE6=[Xe11(6,ff) Xe12(6,ff)];
YE6=[Ye11(6,ff) Ye12(6,ff)];
XE7=[Xe11(7,ff) Xe12(7,ff)];
YE7=[Ye11(7,ff) Ye12(7,ff)];
XE8=[Xe11(8,ff) Xe12(8,ff)];
YE8=[Ye11(8,ff) Ye12(8,ff)];
XE9=[Xe11(9,ff) Xe12(9,ff)];
YE9=[Ye11(9,ff) Ye12(9,ff)];
XE10=[Xe11(10,ff) Xe12(10,ff)];
YE10=[Ye11(10,ff) Ye12(10,ff)];
XE11=[Xe11(11,ff) Xe12(11,ff)];
YE11=[Ye11(11,ff) Ye12(11,ff)];
XE12=[Xe11(12,ff) Xe12(12,ff)];
YE12=[Ye11(12,ff) Ye12(12,ff)];

% Heliostatos Lado Oeste

Xo11=zeros(n_h,13);
Yo11=zeros(n_h,13);
Xo12=zeros(n_h,13);
Yo12=zeros(n_h,13);

for u=1:13
    for m=1:n_h
        Xo11(m,u)=Q_no(m)-R*cos(theta_no(m,u)*pi/180);
    end
end

```

```

end
Yo11=-R*sin(theta_no*pi/180);

Xo11;
Yo11;

for u=1:13
    for m=1:n_h
        Xo12(m,u)=Q_no(m)+R*cos(theta_no(m,u)*pi/180);
    end
end
Yo12=R*sin(theta_no*pi/180);

Xo12;
Yo12;

XO1=[Xo11(1,ff) Xo12(1,ff)];
YO1=[Yo11(1,ff) Yo12(1,ff)];
XO2=[Xo11(2,ff) Xo12(2,ff)];
YO2=[Yo11(2,ff) Yo12(2,ff)];
XO3=[Xo11(3,ff) Xo12(3,ff)];
YO3=[Yo11(3,ff) Yo12(3,ff)];
XO4=[Xo11(4,ff) Xo12(4,ff)];
YO4=[Yo11(4,ff) Yo12(4,ff)];
XO5=[Xo11(5,ff) Xo12(5,ff)];
YO5=[Yo11(5,ff) Yo12(5,ff)];
XO6=[Xo11(6,ff) Xo12(6,ff)];
YO6=[Yo11(6,ff) Yo12(6,ff)];
XO7=[Xo11(7,ff) Xo12(7,ff)];
YO7=[Yo11(7,ff) Yo12(7,ff)];
XO8=[Xo11(8,ff) Xo12(8,ff)];
YO8=[Yo11(8,ff) Yo12(8,ff)];
XO9=[Xo11(9,ff) Xo12(9,ff)];
YO9=[Yo11(9,ff) Yo12(9,ff)];
XO10=[Xo11(10,ff) Xo12(10,ff)];
YO10=[Yo11(10,ff) Yo12(10,ff)];
XO11=[Xo11(11,ff) Xo12(11,ff)];
YO11=[Yo11(11,ff) Yo12(11,ff)];
XO12=[Xo11(12,ff) Xo12(12,ff)];
YO12=[Yo11(12,ff) Yo12(12,ff)];

figure
% Trazado de Rayos Este
plot(X1e,Yh,'bo-')
hold on
plot(X2e,Yh,'bo-')
plot(X3e,Yh,'bo-')
plot(X4e,Yh,'bo-')
plot(X5e,Yh,'bo-')
plot(X6e,Yh,'bo-')
plot(X7e,Yh,'bo-')
plot(X8e,Yh,'bo-')
plot(X9e,Yh,'bo-')
plot(X10e,Yh,'bo-')
plot(X11e,Yh,'bo-')
plot(X12e,Yh,'bo-')

% Trazado de Rayos Oeste
plot(X1o,Yh,'bo-')

```

```

plot(X2o,Yh,'bo-')
plot(X3o,Yh,'bo-')
plot(X4o,Yh,'bo-')
plot(X5o,Yh,'bo-')
plot(X6o,Yh,'bo-')
plot(X7o,Yh,'bo-')
plot(X8o,Yh,'bo-')
plot(X9o,Yh,'bo-')
plot(X10o,Yh,'bo-')
plot(X11o,Yh,'bo-')
plot(X12o,Yh,'bo-')

grid on

% Formación de Heliostatos
hold on
plot(XE1,YE1,'r-')
plot(XE2,YE2,'r-')
plot(XE3,YE3,'r-')
plot(XE4,YE4,'r-')
plot(XE5,YE5,'r-')
plot(XE6,YE6,'r-')
plot(XE7,YE7,'r-')
plot(XE8,YE8,'r-')
plot(XE9,YE9,'r-')
plot(XE10,YE10,'r-')
plot(XE11,YE11,'r-')
plot(XE12,YE12,'r-')

plot(XO1,YO1,'r-')
plot(XO2,YO2,'r-')
plot(XO3,YO3,'r-')
plot(XO4,YO4,'r-')
plot(XO5,YO5,'r-')
plot(XO6,YO6,'r-')
plot(XO7,YO7,'r-')
plot(XO8,YO8,'r-')
plot(XO9,YO9,'r-')
plot(XO10,YO10,'r-')
plot(XO11,YO11,'r-')
plot(XO12,YO12,'r-')

xlabel('X (m)', 'FontSize',14);
ylabel('Y (m)', 'FontSize',14);
axis([-11 11 -1 14])
title('Angulación de Espejos LFC Seguimiento Solar - Mes: Marzo, Día:
21, Hora: 12:00 m', 'FontSize',16);

figure
plot(Solar_Time,MinimaSep_O,'bo-')
hold on
plot(Solar_Time,MinimaSep_E,'ro-')
xlabel('Hora Solar (hr)', 'FontSize',14);
ylabel('Distancia (m)', 'FontSize',14);
axis([6 19 0 2])
title('Separación mín. espejos contiguos, Mes: Marzo', 'FontSize',16);
legend('Espejos Oeste','Espejos Este',1);
grid on
figure

```



```

plot(Solar_Time,theta,'bo-')
xlabel('Hora Solar (hr)', 'FontSize',14);
ylabel('Angulo Incidencia (°)', 'FontSize',14);
grid on

```

6. Script Matlab - Balance Térmico

```

%=====
% COLECTOR PARABOLICO COMPUESTO - DOBLE ABSORBEDOR
% SCRIPT DE PERFORMANCE TERMICO
% Area individual espejos: 0.50 x 50 m; Cantidad: 24 Espejos
% Campo de espejos: 1000 m2; Amplitud: 20 m; Altura: 14 m.
% Precisión 2
%=====

clc
clear all

mm=3;
dd=21;

[Solar_Time,omega,theta,theta_z,gamma_s,alpha_s,Hora_sale_sol,Hora_pone_sol] = Ang_solar_f7_3( mm,dd );

% Horario de control 8:00 am a 16:00 pm
Solar_Time_1=Solar_Time(9:41);

FlujoQ_total=xlsread('Flujo Energetico CPCS','DATA2TCPC','C3:AI14');

FlujoQ_local=FlujoQ_total(mm,:);

% Evaluación de un colector CPC, con agua de fluido colector térmico
% tubería de cobre como receptor y cubierta de vidrio al vacío.
% Cálculo de temperaturas del fluido, tubo colector y tubo de vidrio

L=50; % Longitud útil del sistema receptor (m)
d_ci=0.119; % Diámetro interior de la cubierta de vidrio (m)
et_cub=0.003; % Espesor de cubierta de vidrio (m)
d_ce=d_ci+2*et_cub; % Diámetro exterior de la cubierta de vidrio (m)
d_ri=0.076; % Diámetro interior del tubo receptor (m)
et_r=0.0025; % Espesor de cubierta de tubo receptor (m)
d_re=d_ri+2*et_r; % Diámetro exterior del tubo receptor (m)
A_r=pi*d_re*L; % Area exterior del tubo receptor (m2)
A_c=pi*d_ce*L; % Area exterior de la cubierta (m2)
k_g=0.81; % Conductividad termica del vidrio (W/m-K)
k_cu=375; % Conductividad termica del cobre (W/m-K)
sigma=5.67e-8; % Constante Stefan-Boltzman (W/m2 k^4)
epsilon_r=0.75; % Emitancia de la tubería de cobre
epsilon_v=0.09; % Emitancia de la cubierta de vidrio
thau_vidrio=0.81; % Transmitancia de la cubierta de vidrio
alpha_vidrio=0.088; % Absortancia de la cubierta de vidrio
phsi_vidrio=0.102; % Reflectancia de la cubierta de vidrio
thau_tcu=0; % Tramitancia tubo receptor
alpha_tcu=0.89; % Absortancia tubo receptor (superf. selectiva)
phsi_tcu=0.11; % Reflectancia tubo receptor (superf. selectiva)
Fgpm=4; % Flujo volumétrico propuesto fluido (gpm)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
BALANCE ENERGETICO Y TRANSFERENCIA DE CALOR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Flujo de Radiación directa en (J/s=Watt) obtenida de Ray_trazing
% Corrección por nubosidad
Mnube_local=xlsread('Climatic_Huancavelica_Model','ModelNube','C11:N43')
;
RNube=Mnube_local(:,mm);
Q_input_i=A_r*FlujoQ_local.*(1-RNube)';
Q_input_i

% Cálculo de coeficiente por convección del viento h_v
MViento_local=xlsread('Climatic_Huancavelica_Model','ModelViento','C11:N
43');
V=MViento_local(:,mm); % Velocidad de viento (m/s) (Tm=10°C)
P=59.28; % Presión atmosférica de la zona (kPa)
nu_0=1.426e-5; % Viscosidad cinemática aire a 10°C (m2/s)
nu_1=nu_0/(P/101.33); % Viscosidad cinemática aire correc. altitud
(m2/s)
rho_aire=1.247; % Densidad del aire a 10°C (kg/m3)
k_v=0.02439; % Conductividad térmica del aire a 10°C (W/m-K)
Pr_v=0.7336; % Número de Prandlt aire a 10°C
Tsol=5778; % Temperatura promedio del sol (K)

% Cálculo del Número de Reynolds del flujo de aire
Re_v=V.*d_ce/nu_1;
Re_vc=5e+5; % Número de Reynolds crítico

% Cálculo del Número de Nusselt (Correlaciones Zukauskas)
for i=1:25
    if 0.4<Re_v(i) && Re_v(i)<4
        Nu=0.989*(Re_v(i)^0.330)*(Pr_v^0.33);
    end
    if 4<Re_v(i) && Re_v(i)<40
        Nu=0.911*(Re_v(i)^0.385)*(Pr_v^0.33);
    end
    if 40<Re_v(i) && Re_v(i)<4000
        Nu=0.683*(Re_v(i)^0.466)*(Pr_v^0.33);
    end
    if 4000<Re_v(i) && Re_v(i)<40000
        Nu=0.193*(Re_v(i)^0.618)*(Pr_v^0.33);
    end
    if 40000<Re_v(i) && Re_v(i)<400000
        Nu=0.027*(Re_v(i)^0.805)*(Pr_v^0.33);
    end
end
h_v=Nu*k_v/d_ce; % Coeficiente de convección por viento

% Datos para estimación de pérdidas en el fluido
% Estimación de propiedades correlación de datos experimentales 0-100°C
Mtemp_local=xlsread('Climatic_Huancavelica_Model','ModelTemp','C11:N43')
;
T_eC=Mtemp_local(:,mm);
T_e=T_eC'+273; % Temperatura exterior ambiente (K)
T_s=T_e; % Temperatura del suelo (K)
Tf_in=T_e; % Temperatura de ingreso del fluido (K)
tau_w=0.90; % Transmitancia del medio
T_a=3+273; % Temperatura ambiente del medio (K)

```

```

k_air=0.03095;          % Conductividad del aire (m2/s) a 100°C
Pr_air=0.7111;         % Número de Prandlt del aire a 100°C
rho_wprom=971.5;       % Densidad promedio agua rango 50-100°C
Cp_wprom=4.1962e+3;    % Capacidad calorífica del agua rango 50-100°C

%-----
% Balance Energía en Cubierta Concéntrica de Vidrio
%-----
% Q_input_i=Q_input_i_1+Q_abs2+Q_reflej2
% Q_abs2=Q_C2+Q_R2
% Q_C2=h_v*A_c*(Tsv1-T_e);
% Q_R2=sigma*epsilon_v*A_c*(Tsv1.^4-T_s.^4);

Q_input_i_1=Q_input_i*thau_vidrio;
Q_reflej2=Q_input_i*phsi_vidrio;
Q_abs2=Q_input_i*alpha_vidrio;

AA=h_v*A_c;
BB=sigma*epsilon_v*A_c;
Tsv1=zeros(1,33);

for i=1:33
    Func=[BB 0 0 AA -(AA*T_e(i)+BB*T_s(i)^4)];
    Txsv1=roots(Func);
    Tsv1(i)=min(Txsv1);
end

Q_C2=h_v*A_c*(Tsv1-T_e);
Q_R2=sigma*epsilon_v*A_c*(Tsv1.^4-T_s.^4);
Tsv2=Q_input_i_1*log(d_ce/d_ci)/(2*pi*k_g*L)+Tsv1;

Q_input_i_1
Q_reflej2
Q_C2
Q_R2
Tsv1
Tsv2

%-----
% Balance Energía en el Receptor
%-----
% Q_input_i_1=Q_abs+Q_reflej1
% Q_abs=Q_C1+Q_R1

Q_abs=alpha_tcu*Q_input_i_1;
Q_reflej1=phsi_tcu*Q_input_i_1;
Q_abs

% Flujo volumétrico definido para el fluido colector (gpm)
Fvol=Fgpm*0.00006309; % (m3/s)

% Flujo másico definido para el fluido colector (kg/s)
fmasico=Fvol*rho_wprom;
Tf_out=Q_abs./(fmasico*Cp_wprom)+Tf_in;
Tf_out

T_x=0.5*(Tf_out+Tf_in)-273 % Temperatura para hallar propiedades de
evaluación (K)

```

```

for i=1:33
    if T_x(i)<=100
        % Capacidad calorífica del agua (J/kg-K)
        Cp_w=1000*(2e-9*T_x.^4-5e-7*T_x.^3+6e-5*T_x.^2-
            0.0027*T_x+4.2128);
        % Viscosidad dinámica del agua (Ns/m2)
        mu_w=2e-7*T_x.^2-3e-5*T_x+0.0016;
        % Conductividad térmica del agua (W/m-K)
        k_w=-8e-6*T_x.^2+0.002*T_x+0.5648;
        % Densidad del agua (kg/m3)
        rho_w=-0.0025*T_x.^2-0.2151*T_x+1005;
        % Número de Prandlt del agua
        Pr_w=2e7*T_x.^4-6e-5*T_x.^3+0.0061*T_x.^2-0.3426*T_x+10.951;
        % Pr_w=0.0013*T_x.^2-0.2057*T_x+10.16;
    end
    if T_x(i)>100
        % Capacidad calorífica del agua (J/kg-K)
        Cp_w=1000*(2e-11*T_x.^4-3e-8*T_x.^3+2e-5*T_x.^2-
            0.0041*T_x+2.2869);
        % Viscosidad dinámica del agua (Ns/m2)
        mu_w=4e-8*T_x+8e-6;
        % Conductividad térmica del agua (W/m-K)
        k_w=6E-8*T_x.^2+7e-5*T_x+0.0169;
        % Densidad del agua (kg/m3)
        rho_w=4e-12*T_x.^4-7e-9*T_x.^3+6e-6*T_x.^2-0.0026*T_x+0.8034;
        % Número de Prandlt del agua
        Pr_w=4e-12*T_x.^4-6e-9*T_x.^3+3e-6*T_x.^2-0.001*T_x+1.0671;
    end
end

T_m=(Tf_out+Tf_in)/2;

% Número de Reynolds del fluido colector térmico
Re_w=4*fmasico./(pi*d_ri*mu_w);
Re_w

% Número de Nusselt del fluido colector térmico
Nu_f=zeros(1,33);

for j=1:33
    if Re_w(j)<=2300 % Régimen laminar
        disp('Régimen de flujo laminar')
        % Solución asintótica transferencia T=const.
        friccion=64./Re_w;
        Nu_f(j)=3.66;
    end
    if Re_w(j)>2300 && Re_w(j)<5e+6 % Régimen transitorio-turbulento
        disp('Régimen de flujo transitorio-turbulento')
        % Correlación de Petukhov
        friccion=(0.790*log(Re_w)-1.64).^(-2);
        % Correlación de Gnielinski
        Nu_f(j)=((friccion(j)/8).*(Re_w(j)-
            1000)*Pr_w(j))/(1+12.7*(friccion(j)/8).^0.5*((Pr_w(j).^0.667)-1));
    end
end
Nu_f
friccion

```

```

% Cálculo de la temperatura del tubo receptor (K)
h_w=Nu_f.*k_w/d_ri;
h_w

% Ensamble de ecuaciones vinculantes
% Q_C1=h_w*A_r*(Tst2-T_m);
% Q_R1=sigma*A_r*(Tst2.^4-Tsv2.^4)/(1/epsilon_r+(1-
epsilon_v)/epsilon_v)*(d_re/d_ce);
% Q_abs=sigma*A_r*(Tst2.^4-Tsv2.^4)/(1/epsilon_r+(1-
epsilon_v)/epsilon_v)*(d_re/d_ce)+h_w*A_r*(Tst2-T_m);
% sigma*A_r*(Tst2.^4-Tsv2.^4)/(1/epsilon_r+(1-
epsilon_v)/epsilon_v)*(d_re/d_ce)+h_w*A_r*(Tst2-T_m)-Q_abs=0;
CC1=sigma*A_r;
CC2=1/(1/epsilon_r+(1-epsilon_v)*(d_re/d_ce)/epsilon_v);
CC3=h_w*A_r;
% CC1/CC2*Tst2^4-CC1/CC2*Tsv2^4+CC3*Tst2-CC3*T_m-Q_abs=0;
% CC1/CC2*Tst2^4+CC3*Tst2-CC1/CC2*Tsv2^4-CC3*T_m-Q_abs=0;

Tst2=zeros(1,33);

% Tst1=Q_input_i_1*log(d_re/d_ri)/(2*pi*k_air*L)+Tsv2;

for i=1:33
    Func=[CC1/CC2 0 0 CC3(i) (-CC1/CC2*Tsv2(i)^4-CC3(i)*T_m(i)-
Q_abs(i))];
    Txst2=roots(Func);
    Tst2(i)=min(Txst2);
end

Tst2

Q_R1=sigma*A_r*(Tst2.^4-Tsv2.^4)/(1/epsilon_r+(1-
epsilon_v)*(d_re/d_ce)/epsilon_v);

% Energía ganada para el Proceso
Q_C1=h_w*A_r*(Tst2-T_m);
Q_C1
Tst1=Tst2+Q_C1*log(d_re/d_ri)/(2*pi*k_cu*L);
Tst1

%-----
% PERFORMANCE DEL SISTEMA
%-----
% Cálculo de la eficiencia del colector por 1° Ley Termodinámica
Q_win=Q_C1;
n_ef=Q_win./Q_input_i;
n_ef

% Cálculo de la eficiencia del colector por 2° Ley Termodinámica
% Potencia de bombeo del fluido
v_w=fmasico./(rho_w*0.25*pi*d_ri^2); % Velocidad del flujo
deltaP=friccion*L.*rho_w.*v_w.^2/(2*d_ri); % Caída de presión
W_p=(fmasico./rho_w).*deltaP;
v_w
deltaP
W_p
Q_win

```

```

% Cálculo de la Energía pérdida
Q_loss=Q_R1+Q_R2+Q_reflej1+Q_reflej2+Q_C2;
Q_loss

% Cálculo de la Energía neta y la Exergía
Q_net=Q_input_i-Q_loss;
Exergy=Q_win.*(1-T_e./Tf_out);
Exergy

% Cálculo de la Eficiencia exergética
n_ex=(Q_win.*(1-T_e./Tf_out)-W_p)./(Q_input_i.*(1-T_e./Tsol));
n_ex

figure
plot(Solar_Time_1,Q_input_i,'b*-')
grid on
hold on
plot(Solar_Time_1,Q_loss,'r*-')
plot(Solar_Time_1,Q_net,'ko-')
plot(Solar_Time_1,Exergy,'mo-')
xlabel('Hora Solar (hr)','FontSize',14);
ylabel('Flujo de calor (Watt)','FontSize',14);
title('Energía Radiante CPC-2T - Pilpichaca
Huancavelica','FontSize',16);
legend('Q_input_i','Qloss','Qnet','Exergy',1);
hold off

figure
plot(Solar_Time_1,Tst2-273,'b*-')
hold on
plot(Solar_Time_1,Tsv1-273,'r*-')
plot(Solar_Time_1,Tf_out-273,'ko-')
xlabel('Hora Solar (hr)','FontSize',14);
ylabel('Temperatura (°C)','FontSize',14);
title('Variación de Temperatura CPC-2T - Pilpichaca
Huancavelica','FontSize',16);
legend('Superf. Tubo colector','Superf. Cubierta Vidrio','Salida
fluido',1);
grid on

figure
plot(Solar_Time_1,n_ef,'bo-')
xlabel('Hora Solar (hr)','FontSize',14);
ylabel('nef - 1° Ley Termod.','FontSize',14);
title('Eficiencia energética CPC-2T - Pilpichaca
Huancavelica','FontSize',16);
grid on

figure
plot(Solar_Time_1,n_ex,'ro-')
xlabel('Hora Solar (hr)','FontSize',14);
ylabel('nex - 2° Ley Termod.','FontSize',14);
title('Eficiencia exergética CPC-2T - Pilpichaca
Huancavelica','FontSize',16);
grid on

```