

Capítulo 9

Anexos

9.1. Anexo I: Función *eficienciageoysombras*

En este anexo se presenta la función *eficienciageoysombras* que se encarga del cálculo del Factor de Sombra en el campo de espejos de Fresnel como consecuencia de las sombras propias entre espejos, pérdidas ópticas y tramo de tubo de absorción en sombra como consecuencia del azimut del Sol. Este factor interviene en el cálculo de la radiación efectiva que llega al tubo absorbedor.

```
USE MATH
USE MATH_2
-- Cálculo factor de sombra
-- Entradas a la función: ángulo diario [radianes] y hora solar verdadera Ts [decimal]
-- Salidas de la función: Factor de sombra [0-1] del campo (tanto por 1)
FUNCTION REAL eficienciageoysombras(REAL angulodiario, REAL Ts, REAL latitud)

DECLS
  CONST INTEGER N=11 -- 11 lazos
  CONST REAL oriplanta=(12+3/600+11/3600)*PI/180 -- Orientación del campo
  -- Definición de posición por filas de espejos, posición tubo, anchura espejos, área
  del campo y longitud del tubo
  REAL XE[N]={-3.5, -2.8, -2.1, -1.4, -0.7, 1e-8, 0.7, 1.4, 2.1, 2.8, 3.5}
  REAL Xtubo=0
  REAL Ytubo=4
  REAL Anchuraespejos=0.25
  REAL Areatotal=352
  REAL longitud=64
  -- Otras variables locales del componente
  REAL declinacion
  REAL anghorario
  REAL alturasolar
  REAL azimut
  REAL sinazimut
  REAL i2d[2]
  REAL i3d[3]
  REAL modi2d
  REAL gamma
  REAL r2d[2,N]
  REAL Beta[N]
  REAL Alpha[N]
  REAL Theta[N]
  REAL inclinacion[N]=1
  REAL factorsombra[N]=0
  REAL kappa[N]=1
  INTEGER i
```

```

REAL Pii
REAL Pj
REAL D[2]
REAL E[2]
REAL dx
REAL dy
REAL modulo
REAL w
REAL a
REAL b
REAL Pps
REAL eficienciageometrica[N]
REAL n[3,N]
REAL ZR
REAL C
REAL ff
REAL partesombra[N]
REAL fulit
REAL tubosombra
REAL Eficienciatubosombra
REAL Factorsombra
REAL Div[N]

```

BODY

```

-- Cálculo de la declinación por Spenser
declinacion=0.006918-0.399912*cos(angulodiario)+0.070257*sin(angulodiario)-
0.006758*cos(2*angulodiario)+0.000907*sin(2*angulodiario)-
0.002697*cos(3*angulodiario)+0.00148*sin(3*angulodiario)
-- Cálculo del ángulo horario en radianes
anghorario=((Ts-12)*15)*(PI/180)
-- Cálculo de la altura y el azimut solar
alturasolar=asin(sin(declinacion)*sin((latitud*PI)/180)+cos(declinacion)*cos((latitud
*PI)/180)*cos(anghorario))
azimut=acos((sin(alturasolar)*sin((latitud*PI)/180)-
sin(declinacion))/(cos(alturasolar)*cos((latitud*PI)/180)))
-- Cálculo del signo del azimut solar
sinazimut=(cos(declinacion)*sin(anghorario))/(cos(alturasolar))
-- Si la cantidad es positiva el ángulo es negativo
IF(sinazimut>0)THEN
    azimut=-azimut
END IF
-- Modelo 2D, cálculo del vector i2d solar
IF(azimut>0) THEN
    i2d[1]=cos(alturasolar)*cos(azimut+oriplanta)
    i2d[2]=sin(alturasolar)
    i3d[1]=i2d[1]
    i3d[2]=i2d[2]
    i3d[3]=cos(alturasolar)*sin(azimut+oriplanta)
ELSEIF (azimut<0 AND abs(azimut)<=oriplanta) THEN
    i2d[1]= cos(alturasolar)*cos(oriplanta-abs(azimut))
    i2d[2]=sin(alturasolar)

```

```

        i3d[1]=i2d[1]
        i3d[2]=i2d[2]
        i3d[3]=cos(alturasolar)*sin(oriplanta-abs(azimut))
    ELSEIF (azimut<0 AND abs(azimut)>oriplanta) THEN
        i2d[1]=cos(alturasolar)*cos(abs(azimut)-oriplanta)
        i2d[2]=sin(alturasolar)
        i3d[1]=i2d[1]
        i3d[2]=i2d[2]
        i3d[3]=cos(alturasolar)*sin(abs(azimut)-oriplanta)
    END IF
    modi2d=sqrt(i2d[1]**2+i2d[2]**2)
    FOR (i IN 1,3) i3d[i]=i3d[i]*(1/sqrt(i3d[1]**2+i3d[2]**2+i3d[3]**2))
    END FOR
    gamma=atan(i2d[2]/i2d[1])
    IF(alturasolar<=0)THEN gamma=0
    END IF
    -- Cálculo del vector r2d de posiciones de espejos
    FOR(j IN 1,N) r2d[1,j]=-XE[j]/sqrt(XE[j]**2+Ytubo**2)
    END FOR
    FOR(j IN 1,N) r2d[2,j]=Ytubo/sqrt(XE[j]**2+Ytubo**2)
    END FOR
    -- Cálculo del ángulo que forma cada espejo con la horizontal: Beta
    FOR (i IN 1,N) Beta[i]=atan(r2d[2,i]/abs(r2d[1,i]))
    END FOR
    -- Cálculo de los ángulos alpha y theta
    FOR (i IN 1,N) Alpha[i]=acos((r2d[1,i]*i2d[1]+r2d[2,i]*i2d[2])/modi2d)
    END FOR
    FOR (i IN 1,N) Theta[i]=Alpha[i]/2
    END FOR
    -- Distintos casos, para cada espejo, primero para las primeras 5 filas. Cálculo de
    inclinación
    FOR (i IN 1,5)
        IF (gamma>=Beta[i]) THEN inclinacion[i]=PI/2-Beta[i]-Theta[i]
        ELSEIF (gamma<Beta[i]) THEN inclinacion[i]=PI/2-Beta[i]+Theta[i]
        END IF
    END FOR
    inclinacion[6]=Theta[6]
    FOR (i IN 7,N) inclinacion[i]=Beta[i]+Theta[i]-PI/2
    END FOR
    -- Una vez calculado los ángulos necesarios, se calculan los factores de sombra y la
    eficiencia geométrica
    FOR (i IN 1,10)
        j=i+1
        Pii=inclinacion[i]
        Pj=inclinacion[j]
        -- analizamos los casos:
        -- 1º Si ambas inclinaciones son positivas
        -- 2º Si la primera positiva y la segunda negativa
        -- 3º Si las dos son negativas
        IF (Pii>0 AND Pj>0) THEN
            D[1]=XE[j]-Anchuraespejos*cos(Pj)

```

```

        D[2]=Anchuraespejos*sin(Pj)
        E[1]=XE[i]+Anchuraespejos*cos(Pii)
        E[2]=-Anchuraespejos*sin(Pii)
        -- Ángulo w
        dx=abs(D[1]-E[1])
        dy=abs(D[2]-E[2])
        modulo=sqrt(dx**2+dy**2)
        w=atan(dy/dx)
        a=w-gamma
        b=Pii+gamma
        IF(gamma<w)THEN factorsombra[i]=modulo*sin(a)/sin(b)
        END IF
    END IF
    IF (Pii>0 AND Pj<0) THEN
        D[1]=XE[j]+Anchuraespejos*cos(Pj)
        D[2]=Anchuraespejos*sin(Pj)
        E[1]=XE[i]+Anchuraespejos*cos(Pii)
        E[2]=-Anchuraespejos*sin(Pii)
        -- Ángulo w
        dx=abs(D[1]-E[1])
        dy=abs(D[2]-E[2])
        modulo=sqrt(dx**2+dy**2)
        w=atan(dy/dx)
        a=w-gamma
        b=Pii+gamma
        IF(gamma<w) THEN factorsombra[i]=modulo*sin(a)/sin(b)
        END IF
    END IF
    IF (Pii<0 AND Pj<0) THEN factorsombra[i]=0
    END IF
END FOR
FOR (i IN 1,N) factorsombra[i]=abs(factorsombra[i])*64
END FOR
Pps=1-sum1D(N,factorsombra)/352
-- Se halla ahora la parte del tubo que está en sombra. Ángulo kappa
FOR (i IN 1,N)
    IF(inclinacion[i]>0) THEN kappa[i]=PI/2-gamma-inclinacion[i]
    ELSEIF (inclinacion[i]<0)THEN kappa[i]=PI/2+inclinacion[i]-gamma
    END IF
END FOR
FOR (i IN 1,N) eficienciageometrica[i]=abs(cos(kappa[i]))
END FOR
-- Modelo 3D, parte del tubo que está en sombra
FOR (j IN 1,N) n[1,j]=sin(inclinacion[j])
        n[2,j]=cos(inclinacion[j])
        n[3,j]=0
END FOR
IF(azimut>0) THEN ZR=64
ELSE ZR=0
END IF

```

```

FOR (i IN 1,N)
  C= i3d[2]*n[3,i]-i3d[3]*n[2,i]  --1er elemento del producto vectorial
  ff=(C/cos(inclinacion[i]))**2
  partesombra[i]=ZR-sqrt((XE[i]**2+Ytubo**2)*ff/(1-ff))
END FOR
fulit=mean(N,partesombra)
-- Comprobamos si hay sombra efectivamente o no
IF(ZR==0)THEN tubosombra=abs(fulit/64)
ELSEIF(ZR==64) THEN tubosombra=(64-fulit)/64
END IF
Eficienciatubosombra=1-tubosombra
IF(Eficienciatubosombra>1) THEN Eficienciatubosombra=1
ELSEIF (Eficienciatubosombra<0)THEN Eficienciatubosombra=0
END IF
Factorsombra=Eficienciatubosombra*maxvec(N,eficienciageometrica)*Pps
IF (Factorsombra<0 OR alturasolar<0) THEN Factorsombra=0
ELSEIF (Factorsombra>1) THEN Factorsombra=1
END IF
RETURN Factorsombra

END FUNCTION

```

