



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA  
ENERGÍA - SOLAR FOTOTÉRMICA

ANÁLISIS DE UN SISTEMA DE SEGUIMIENTO SOLAR PARA  
ARREGLOS DE HELIÓSTATOS ACOPLADOS

T E S I S  
QUE PARA OPTAR POR EL GRADO DE:  
DOCTOR EN INGENIERÍA

PRESENTA:  
ISAIAS MORENO CRUZ

TUTOR PRINCIPAL:  
DR. CAMILO ALBERTO ARANCIBIA BULNES  
INSTITUTO DE ENERGÍAS RENOVABLES, UNAM

CIUDAD DE MÉXICO, MAYO 2019



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**

**Tesis Digitales**

**Restricciones de uso**

**DERECHOS RESERVADOS ©**

**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## **JURADO ASIGNADO:**

Presidente: Dr. Raúl Mauricio Rechtman Schrenzel

Secretario: Dra. Heidi Isabel Villafán Vidales

1<sup>er.</sup> Vocal: Dr. Camilo Alberto Arancibia Bulnes

2<sup>do.</sup> Vocal: Dr. David Riveros Rosas

3<sup>er.</sup> Vocal: Dr. Iván Salgado Tránsito

Lugar donde se realizó la tesis:

Instituto de Energías Renovables, UNAM  
Temixco Morelos

## **TUTOR DE TESIS**

DR. CAMILO ALBERTO ARANCIBIA BULNES  
INSTITUTO DE ENERGÍAS RENOVABLES, UNAM

---

**FIRMA**

# Agradecimientos

---

Agradezco a mi familia. A mi madre por su infinito amor, a mi padre por haberme enseñado su dedicación al trabajo. A mi hermano Gilberto que siempre me ha acompañado y apoyado. A mi hermana Lourdes por el apoyo que me ha brindado a lo largo de mi vida, sin ella mi educación no habría sido posible. A mi hermana Margarita por el amor que me ha brindado. A mi sobrina Fátima que alegra mi vida. A mis hermanos Rafael, Guadalupe y Julio.

Agradezco a los amigos que encontré en el instituto y que espero sean para toda la vida. A Camilo Jiménez, Aldo Guadarrama, Alejandro Ayala, Ernesto Suárez, Pedro Soto, gracias por su amistad y capacidad de reír de prácticamente todo. A mis amigos no menos risueños con quienes intercambie ideas y frustraciones, gracias a Hilda Mota, Carlos López, Rodolfo Pérez.

Agradezco a mi asesor Camilo Arancibia, que me ha guiado y apoyado. Agradezco también al mis asesores David Riveros, Raúl Rechtman, Heidi Villafán, Marcelino Sánchez e Iván Salgado por brindarme parte de su tiempo y conocimiento.

Agradezco a Juan Manuel Valero y Cristopher Escamilla por su amistad e impartición del “Taller de radio y literatura”, espacio que fomenta los cuentos cortos.

Agradezco al IER y a la UNAM por brindarme las instalaciones necesarias para desarrollar mi doctorado.

Agradezco al CONACYT y a todos los mexicanos, que gracias a su trabajo me han permitido terminar mis estudios de doctorado.



# Resumen

---

La motivación para realizar el presente trabajo es repercutir en los costos de los helióstatos, mediante una alternativa del diseño óptico. La razón es que en la tecnología de torre central cerca del 50 % del costo total de la planta se debe al campo de helióstatos, el cual está compuesto por cientos o miles de helióstatos, por lo que un diseño más económico se reflejará en los cientos o miles de helióstatos que componen el campo. Se plantea un diseño de arreglos de helióstatos, pasando de, e.g., 8 mini-helióstatos individuales a un único arreglo que incluyan estos ocho mini-helióstatos. La idea de dicho arreglo es disminuir el costo del sistema de seguimiento solar y de la estructura, el primero debido al mecanismo en donde pasamos de utilizar 16 actuadores para 8 helióstatos individuales a únicamente dos actuadores para realizar el seguimiento en el arreglo, el segundo es debido al diseño de una estructura más rígida y sometida a menores cargas de viento.

Los diseños de los helióstatos convencionales consisten en un seguimiento solar basado en dos rotaciones, una acimutal y otra cenital. En un arreglo de helióstatos al agrupar varios mini-helióstatos, o facetas del arreglo, no es posible mantener estas rotaciones por lo que se proponen las rotaciones que se han denominado *elevación-Fresnel*. Estas dos rotaciones del arreglo permiten realizar el seguimiento basándose en la faceta central del arreglo, provocando errores de seguimiento en las facetas laterales. Por lo que el objetivo de la presente tesis es analizar el desempeño de los arreglos mediante simulación numérica para conocer si es viable utilizar este tipo de arreglos para realizar concentración solar en la tecnología de torre central.

El presente documento consta de siete secciones, la primera de ellas la **Sección 1** lleva por nombre **Introducción**, la cual aborda brevemente la tecnología de *Concentración Solar de Potencia* (CSP) y la prospectiva que se tiene. Se indican algunos de los beneficios de la tecnología de concentración solar de Torre Central, que es la tecnología de nuestro interés. También se introduce brevemente en el tema central de la presente tesis mostrando cuales son los objetivos generales y específicos. En la **Sección 2** se presenta el **Marco Teórico** necesario para entender el análisis óptico del arreglo de helióstatos. Se resume brevemente el movimiento aparente del sol para entender el concepto de *vector solar*. La reflexión es otro concepto importante en la concentración solar, necesario para determinar la orientación de un plano mediante su normal, además de los errores ópticos que se presentan. La tecnología de concentración solar es posible gracias a que se puede redireccionar de manera aceptable la radiación solar, específicamente la radiación normal directa, tema que también es tratado en este apartado. Por último se menciona lo que es el trazado de rayos, útil para realizar simulaciones de concentración solar.

En la **Sección 3** se describe la evolución de los **Helióstatos** a partir de la planta piloto *Solar One*. Se hace mención a un análisis de *Sandia* que llevó a tener heliostatos grandes y baratos, y como consecuencia áreas de reflexión de hasta  $150 \text{ m}^2$ . Posteriormente un cambio de paradigma permitió dar cabida a heliostatos más pequeños conocidos como mini-heliostatos, de hasta  $1.14 \text{ m}^2$ . Se muestra que no existe una correlación entre la potencia instalada de una planta y el área de reflexión de los heliostatos que componen el campo, es decir, un planta de gran capacidad instalada no necesariamente tienen unidades de heliostatos con grandes áreas de reflexión, e.g., el campo de *Ivanpah* está formado por mini-heliostatos ( $15.2 \text{ m}^2$ ) y su potencia es de  $390 \text{ MW}_e$ . Al final de esta sección se hace mención del comportamiento óptico de un heliostato, lo que significa tener un concentrador fuera de eje, además de indicar lo que es el canteo o la curvatura que puede llegar a tener.

Una vez que se tiene un panorama general de la tecnología de concentración solar y más específicamente en torre central, la **Sección 4** titulada **Seguimiento solar elevación-Fresnel** explica la configuración del seguimiento y muestra las ecuaciones necesarias para realizarlo, partiendo de las consideraciones que se hicieron sobre las rotaciones del seguimiento. Se describe la metodología para calcular los ángulos necesarios y orientar las normales de las facetas así como el cálculo del impacto del vector de reflexión en el receptor, ya sea que se trate de un receptor vertical, horizontal o inclinado. La radiación al mediodía solar es de aproximadamente  $1 \text{ kW/m}^2$ , valor que varía con la posición del sol por lo que se utiliza un modelo de cielo claro, mismo que en esta sección se describe. En la **Sección 5** titulada **Programación** se describe la estructura del programa utilizado para realizar el seguimiento solar indicada en la sección anterior, así como los datos de entrada y salida, programa que está escrito en el lenguaje de programación C. Se indica de qué manera los datos de salida del programa en C son utilizados para alimentar las entradas de simulación en SolTrace, gracias a la programación de *scripts* en el lenguaje LK.

En la **Sección 6** de **Resultados** se describen los cinco arreglos simulados y las consideraciones para tales simulaciones. Las simulaciones consistieron en conocer el impacto de cada faceta del arreglo en el receptor, impacto que debido al mecanismo de seguimiento no incide exactamente en el valor deseado pero si muy próximo a él. La desviación que se obtiene en el plano del receptor respecto al valor ideal de reflexión, un valor en  $x$  y otro en  $z$ , permite calcular la desviación estándar del comportamiento diario del heliostato. Esta desviación estándar es utilizada para hacer comparaciones anuales de los distintos arreglos, y poder visualizar el efecto del número de facetas en el arreglo, la orientación del receptor, el efecto de la latitud y del canteo. En esta sección se realiza un análisis de la distribución de flujo de arreglo de mini-heliostatos, así como el área de la imagen en el receptor.

Para finalizar en la **Sección 7** encontramos las conclusiones, así como tres apéndices. En el Apéndice A profundizamos más en los distintos algoritmos para el cálculo de la posición solar, en el Apéndice B se muestra el desarrollo de los cambios de coordenadas utilizados en el código y por último se pueden leer los códigos utilizados en las simulaciones del presente trabajo en el Apéndice C.

---

# Contenido

---

<b>Lista de Figuras</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Objetivo General . . . . .	3
1.1.1. Objetivos específicos . . . . .	3
<b>2. Marco Teórico</b>	<b>5</b>
2.1. Conceptos básicos de concentración solar . . . . .	5
2.2. Calidad óptica de la reflexión de un concentrador . . . . .	7
2.2.1. Errores de superficies . . . . .	8
2.2.2. Combinación de los errores ópticos . . . . .	9
2.2.3. La forma solar . . . . .	10
2.3. Concentración de radiación solar . . . . .	10
2.4. Trazado de rayos . . . . .	11
<b>3. Helióstatos</b>	<b>15</b>
3.1. Historia . . . . .	15
3.2. Funcionamiento . . . . .	17
3.2.1. Canteo . . . . .	20
3.3. Diseño . . . . .	23
<b>4. Seguimiento solar elevación-Fresnel</b>	<b>27</b>
4.1. Configuración de seguimiento solar . . . . .	27
4.2. Metodología . . . . .	28
<b>5. Programación</b>	<b>33</b>
5.1. Programación en C . . . . .	33
5.2. SolTrace . . . . .	35
<b>6. Resultados</b>	<b>39</b>
6.0.1. Efecto del número de facetas . . . . .	43
6.0.2. Efecto de la orientación del receptor . . . . .	45
6.0.3. Efecto de Latitud . . . . .	47



6.0.4. Efecto del día de canteo . . . . .	49
6.0.5. Arreglos de Helióstatos Fuera de Eje . . . . .	49
6.0.6. Análisis de distribución de flujo . . . . .	52
<b>7. Conclusiones</b>	<b>59</b>
<b>A. Cálculo de la posición solar</b>	<b>61</b>
<b>B. Cambio de Coordenadas</b>	<b>63</b>
B.1. Sistemas de referencia . . . . .	63
<b>C. Código</b>	<b>67</b>
C.1. Programación en C . . . . .	67
C.1.1. Archivo principal <code>erun.c</code> . . . . .	67
C.1.2. Archivos de cabecera . . . . .	72
C.1.3. Archivos de funciones . . . . .	74
C.2. Script LK . . . . .	85
<b>Referencias</b>	<b>91</b>

---

# Lista de Figuras

---

2.1.	Sistema de receptor central. Adaptada de (Biggs y Vittitoe, 1979) . . . . .	5
2.2.	(a) Vector solar $\hat{s}$ descrito por sus ángulos cenital $\theta_z$ y acimutal $\gamma_s$ , este último medido desde el sur. (b) Trayectoria del Sol en el plano de su trayectoria aparente. . . . .	6
2.3.	Reflexión especular. . . . .	8
2.4.	Características de una superficie de concentración a gran, mediana y pequeña escala. . . . .	9
2.5.	(a) error de pendiente por canteo y (b) error óptico global. . . . .	10
3.1.	Evolución de reducción de costos. Tomada de Kolb y cols. (2007). . . . .	15
3.2.	Se muestran las plantas de Torre Central en el tiempo comparadas así como la potencia eléctrica de las mismas, el círculo indica el área de reflexión de un helióstato que compone el campo. El helióstato con menor área de reflexión es el de Sierra SunTower (eSolar) con 1.14 m <sup>2</sup> y el mayor es de 121 m <sup>2</sup> que le corresponden a los helióstatos de la PS-10 y PS-20. . . . .	18
3.3.	Helióstato de Campo Experimental de Torre Central (CEToC). Sonora, México. . . . .	19
3.4.	Rayos tangenciales y sagitales para un helióstato esférico a la distancia $D$ del receptor, en la imagen de la izquierda. Rayos tangenciales y sagitales para una concentración fuera de eje de un helióstato esférico. Tomada Rabl (1985). . . . .	20
3.5.	Esquema de un helióstato (a) plano, (b) enfoque y (c) canteo. Adaptada de Kistler (1986). . . . .	20
3.6.	Canteo esférico. . . . .	21
3.7.	Canteo de un helióstato para un día en específico al medio día solar, del lado izquierdo. Sistema de referencia de la normal $\hat{n}$ central del helióstato, del lado derecho. . . . .	22
3.8.	Diseño del campo de helióstatos dependiendo de la localización. Adaptada de Alexopoulos y Hoffschmidt (2013). . . . .	24
3.9.	Bloqueo. Adaptada de Alexopoulos y Hoffschmidt (2013). . . . .	25
3.10.	Sombreo. Adaptada de Alexopoulos y Hoffschmidt (2013). . . . .	25
4.1.	Sistema de seguimiento general. . . . .	28
4.2.	Vector de incidencia y reflexión en el plano de rotación de las facetas. . . . .	29

4.3. (a) Sistema de referencia global en donde se observa el vector solar $\hat{\mathbf{s}}$ , el vector normal $\hat{\mathbf{n}}$ , y el vector de reflexión $\hat{\mathbf{t}}$ . (b) Rotaciones $\xi_0$ y $\psi_0$ de la faceta central para el seguimiento de un arreglo de helióstatos. . . . .	29
5.1. Diagrama del programa <b>erun.c</b> . . . . .	34
6.1. Diferentes ubicaciones de los arreglos de helióstatos analizados. . . . .	39
6.2. Punto de impacto del rayo reflejado de cada una de las facetas en el receptor, para las cuatro facetas $f_1$ , $f_2$ , $f_3$ y $f_4$ del arreglo H4. Simulación para el 21 de marzo. Para las posiciones de $0.75h$ y $4h$ , de izquierda a derecha. . . . .	40
6.3. Puntos de impactos de los rayos centrales en el objetivo para el arreglo H4 localizado a $0.75h$ , para el día 21 de cada mes de junio a diciembre. . . . .	41
6.4. Desviación estándar diaria de los puntos de impacto en $x$ (a) y $z$ (b), y el error angular (c), para el arreglo H4. Para las posiciones de arreglo ubicado a $0.75h$ ( $A$ ), $2h$ ( $B$ ) y $4h$ ( $C$ ). . . . .	42
6.5. Desviación estándar diaria para los arreglos de helióstatos H2, H4, H8, H16, H24, y H32 a lo largo del año, a lo largo de los ejes $x$ (a) y $z$ (b), y el error angular (c). Los arreglos están localizados en $2h$ . . . . .	43
6.6. Cuartiles de la desviación estándar diaria para los arreglos de helióstatos H2, H4, H8, H16, H24, y H32 en los ejes $x$ (a) y $z$ (b), y el error angular (c). Para arreglos colocados en las posiciones de $A$ ( $0.75h$ ), $B$ ( $2h$ ) y $C$ ( $4h$ ). . . . .	44
6.7. Comportamiento de los valores de desviación máxima anual como una función del tamaño del arreglo (puntos), para las posiciones de $A$ , $B$ y $C$ con sus correspondientes ajustes polinomiales (líneas). . . . .	45
6.8. La tendencia polinomial de los valores de la desviación estándar máxima para los arreglos colocados a las distancias de $A$ , $B$ y $C$ para el receptor vertical (línea punteada) y horizontal (línea continua). Los resultados para el eje $x$ (a) son presentados para ambos receptores, mientras que los ejes $y$ y $z$ (b) son presentados para el receptor horizontal y vertical, respectivamente. . . . .	46
6.9. Tendencias de la desviación estándar en un receptor plano $\Pi$ para un arreglo de helióstatos colocado en las posiciones de $A$ , $B$ y $C$ y sus intervalos limitados por un receptor vertical y horizontal. . . . .	47
6.10. Comportamiento del valor máximo de la desviación estándar diaria de cada arreglo de helióstato a las latitudes de 2, 6, 10, 30, y $45^\circ + N$ , para los ejes $x$ (a), $z$ (b) y para el error angular $\varepsilon$ . Los arreglos son colocados en una posición de $2h$ . . . . .	48
6.11. Diferencia entre la desviación estándar entre un receptor vertical y un horizontal a diferentes latitudes. (a) Muestra la diferencia de $\Delta\sigma_x$ (b) la diferencia de $\Delta\sigma_{z-y}$ para la posición del arreglo a $0.74h$ , $2h$ y $4h$ . . . . .	50
6.12. Cuartiles de la desviación estándar diaria para el arreglo H16, en el eje $x$ (a) y $z$ (b) de un receptor vertical, en función del día de canteo. Para el arreglo a la posición de $0.75h$ ( $A$ ), $2h$ ( $B$ ) y $4h$ ( $C$ ). . . . .	51

6.13. Se muestra el comportamiento de los impactos del vector de reflexión central de cada una de las facetas ( $f_1$ , $f_2$ , $f_3$ y $f_4$ ) del arreglo H4 en un receptor vertical para el 21 de marzo. Para un arreglo colocado a la distancia $2\mathbf{h}$ cuando el arreglo esta fuera de eje. . . . .	52
6.14. Muestra la desviación estándar diaria en un receptor vertical para los arreglos H2, H4, H8, H16, H24 y H32 colocados fuera de eje, en una posición de $2\mathbf{h}$ . .	52
6.15. Tendencia de crecimiento de la desviación estándar diaria máxima respecto al tipo de arreglo de helióstatos, para las posiciones de $0.75\mathbf{h}$ , $2\mathbf{h}$ y $4\mathbf{h}$ . . . . .	53
6.16. La distribución de flujo solar en el receptor vertical ( $6 \times 6$ m) para un arreglo H8 localizado a $0.75h$ . Al mediodía solar el 21 de junio (a) y diciembre (b). .	54
6.17. El comportamiento diario de la desviación estándar $S_r$ del <i>spot</i> , se muestran en el eje $x$ y $z$ a lo largo del año, para diferentes arreglos colocados a una distancia de $2h$ . . . . .	55
6.18. Comportamiento del arreglo H32 para el 21 de junio a las 6:30 (a), 8:30 (b) y 12:30 (c). Para un arreglo ubicado a una distancia de $2h$ . . . . .	56
6.19. Contribución de energía por una faceta del arreglo correspondiente a lo largo del año. Para una distancia fija de $0.75h$ . . . . .	57
B.1. Cambios de sistemas de referencia . . . . .	64



---

# Capítulo 1

## Introducción

---

La tecnología de *Concentración Solar de Potencia* (CSP, por sus siglas en inglés), se basa en la concentración de los rayos del Sol en una zona focal mediante superficies reflectoras, para lograr grandes temperaturas en un objetivo, lo que permite posteriormente calentar un fluido de trabajo, que mediante un intercambiador de calor, permite generar vapor y hacer funcionar una turbina de vapor similar a las plantas convencionales de generación de potencia basadas en combustibles fósiles. La tecnología de CSP tiene la ventaja de poder ser equipada con un sistema de almacenamiento de energía térmica de bajo costo, usualmente tanques de sales fundidas, lo que permite tener energía renovable despachable. Aunque la tecnología de CSP está en desarrollo, se observa una tendencia de costos a la baja iniciada en 2012, durante el periodo de 2013–2014 el costo nivelado de electricidad (LCOE, por sus siglas en inglés) fue menor en una quinta parte en comparación con el periodo de 2009–2012. En promedio el LCOE fue de 0.27 USD/kWh la reducción se debió a los mejores niveles de radiación, pero también al incremento de los factores de planta debidos a las configuraciones que incluyen almacenamiento térmico. Se espera una nueva tendencia de costos a la baja debida al incremento de la competitividad y a mejoras tecnológicas que le permitiría competir directamente con los combustibles fósiles a partir del 2020, con las políticas y regulaciones correctas (IRENA, 2018).

La tecnología de CSP puede ser clasificada en concentración lineal y puntual, según el tipo de receptor utilizado. Los sistemas de concentración lineal son aquellos que concentran la energía solar en una línea focal como la tecnología de *Canal Parabólico* y los *Concentradores de Fresnel* de foco lineal, mientras que la tecnología de concentración puntual permite concentrar la energía en un punto, la tecnología de *Torre Central* es la tecnología de foco puntual más desarrollada actualmente.

La tecnología de Torre Central (TC), también conocida como tecnología de Receptor Central, tiene la capacidad de concentrar grandes cantidades de energía solar a altas temperaturas en una región fija, i.e., el receptor. Tener un receptor fijo evita la necesidad de transportar la energía a través de grandes recorridos evitando pérdidas de calor, el trabajar con temperaturas altas (hasta de 1,000°C) permiten altas eficiencias en el sistema de conversión de potencia (eficiencia de Carnot) (Vant-Hull, 2012). Esta tecnología tiene un bajo impacto ambiental, sin emisiones de contaminantes asociadas a las tecnologías, considerando el ciclo de vida de las emisiones de dióxido de carbono en la atmósfera se encontró que emite solo 12 kg de dióxido de carbono por MWh, emisiones menores a por ejemplo plantas

hidroeléctricas (14 kg/MWh), turbinas eólicas (17 kg/MWh) y los sistemas fotovoltaicos (110 kg/MWh) (Alexopoulos y Hoffschmidt, 2013). Por lo que la tecnología de Torre Central es una de las opciones más atractivas de la Concentración Solar de Potencia para la producción de electricidad a gran escala.

Las Torre Central consiste de tres principales subsistemas: el campo de helióstatos, el receptor y el bloque de potencia. Adicionalmente puede incluir un sistema de almacenamiento térmico que le permite generar energía eléctrica o vapor de proceso después de que el sol se a ocultado, así como mantener una producción continua por cambios súbitos en el clima, i.e., nubosidades. El campo de helióstatos está formado por una gran cantidad de espejos, pueden ser planos o con una curvatura, que reflejan la radiación a un receptor localizado en lo alto de la torre. Estos helióstatos tienen un seguimiento automatizado que permiten reflejar la energía del sol a un punto fijo en cualquier instante de tiempo, para realizar el seguimiento del sol son necesarias dos rotaciones independientes. Los helióstatos son uno de los aspectos más críticos de la tecnología de TC. Desde un punto de vista económico, el campo de helióstatos es el componente más caro de la planta de potencia (Ortega y cols., 2008; Li y cols., 2016; Götttsche y cols., 2010), cerca del 50 % de costo total de la planta, lo que crea un gran incentivo para reducir el costo de estos dispositivos.

Se han realizados algunos análisis de costos (Kolb y cols., 2007) en donde se muestra que los grandes helióstatos permiten reducir costos para plantas de gran escala. Tradicionalmente los helióstatos son de gran tamaño, y su evolución eventualmente a llevado a incrementar su tamaño por arriba de los 120 m<sup>2</sup>. La lógica nos indica que al incrementar el área de cada helióstato individual el número de motores (un componente costoso) se reduce, junto con el número de controladores electrónicos para el seguimiento. Sin embargo, algunos diseños van en un sentido opuesto que se habían estado desarrollando, debido al uso de mini-helióstatos. Los mini-helióstatos son helióstatos con un área de reflexión pequeña en comparación con los primeros diseños de helióstatos. La idea es que estos mini-helióstatos puedan soportar menores cargas de viento que sus predecesores más grandes, debido a que la carga de viento tiene una dependencia cuadrática respecto de la velocidad del viento, al reducir el área de contacto se reduce drásticamente las carga de viento, lo cual permite reducir la estructura del helióstato. Eso hace que se reduzca la carga de gravedad y por consiguiente el torque o potencia de los motores. Por lo tanto, el conjunto de los componentes mecánicos, incluyendo motores, pueden ser mucho más ligeros (en proporción al tamaño). La reducción de las piezas electrónicas en la última década ha sido un argumento en favor de esta opción (Domínguez-Bravo y cols., 2016).

Para alcanzar los costos nivelados de electricidad (LCOE) de 0.10 USD/kWh, que permitan tener una tecnología de torre central competitiva, los helióstatos deben de costar no más de 120 USD/m<sup>2</sup>, aunque el objetivo es de 75 USD/m<sup>2</sup> (LCOE de 0.06 USD/kWh). El porcentaje del costo de los componentes que constituyen un helióstato (Pfahl, 2013) son: 25 % en ensamble de los espejos, 15 % en la estructura de soporte, 15 % en el pedestal y cimentación, 5 % en sensores y control, 30 % en motores y 10 % en ensamble e instalación. Para reducir costos se requieren diseños innovadores (Pfahl y cols., 2017) respecto al concepto de helióstato, una modificación en la estructura y el seguimiento representaría un impacto sobre

---

el 60 % del costo del heliostato.

Se han realizado distintos esfuerzos por disminuir el costo de los heliostatos, como el diseño de un heliostato confinado dentro de una caja transparentes (Göttsche y cols., 2010) que minimiza las cargas de viento. En la planta *Sierra SunTower* construida por eSolar (Schell, 2011) la estrategia adoptada para reducir el costo del campo de heliostatos fue utilizar mini-heliostatos para simplificar la instalación de la planta y aprovechar la disminución de los costos por la producción en masa. La estabilidad de los mini-heliostatos es lograda por una estructura que une heliostatos individuales y debido a su baja altura, éstos están sujetos a bajas cargas de viento. Otra alternativa ha sido usar arreglos de heliostatos con la intención de disminuir el número de motores usados para el seguimiento (Dunham y cols., 2013). Todos los conceptos mencionados anteriormente utilizan un sistema de seguimiento solar basado en la rotación de dos ángulos, cenit/acimut. Amsbeck y cols. (2007) propone cambiar el sistema de seguimiento con la intención de usar grandes arreglos de heliostatos, su seguimiento consiste en un mecanismo de elevación común para toda una serie de facetas y un movimiento acimut donde todas las facetas están montadas en un tubo con torque que las une, sin embargo es un arreglo que tiene pérdidas por astigmatismo, en comparación con los heliostatos con seguimiento independientes.

En la presente tesis se analiza la precisión del seguimiento en los arreglos de heliostatos, los cuales son arreglos de facetas unidas mediante un seguimiento denominado *elevación-Fresnel*. Este enfoque permite reducir el número de motores y transmisiones utilizadas para el seguimiento, al realizar arreglos de tres o más facetas. Sin embargo, como el apuntamiento está basado en la faceta central del arreglo, existe un error en el seguimiento para las facetas laterales. Para evaluar el impacto de este error en el seguimiento que se propone, se ha analizado la reflexión del vector solar de una serie de arreglos de heliostatos, durante todo el año, así como los efectos del número de facetas del arreglo, la latitud, el día de canteo, y la orientación del receptor. El comportamiento del seguimiento de este tipo de arreglos es estudiado a detalle, así como sus efectos en la distribución de flujo concentrada. Para llevar a cabo este trabajo se desarrolló un código en el programa C el cual simula el seguimiento del sol para cada arreglo de heliostatos, para obtener la deriva de los puntos de impacto en el receptor. La simulación de flujo se realizó en *SolTrace* (Wendelin y cols., 2013) y el análisis por medio del programa R (R Core Team, 2015).

## 1.1. Objetivo General

El objetivo de la presente tesis es analizar el desempeño óptico de arreglos de heliostatos, mediante la simulación numérica, utilizando una nueva configuración a la cual le hemos llamado *elevación-Fresnel*.

### 1.1.1. Objetivos específicos

Los objetivos específicos son simular y analizar:

- El comportamiento diario y anual de la deriva en el receptor



- Distintos arreglos de helióstatos a distintas posiciones de la torre
- El efecto de la latitud en los arreglos
- Distintos tipos de receptores en la torre
- El efecto del día de canteo en los arreglos
- El trazado de rayos de los arreglos de helióstatos.

---

## Capítulo 2

# Marco Teórico

---

La energía de concentración solar es una tecnología capaz de generar energía eléctrica a gran escala, así como de suministrar energía térmica a la industria. Al incorporar un sistema de almacenamiento térmico, se añade flexibilidad al sistema permitiendo diferir la generación de energía eléctrica acorde a la demanda. Las torres centrales pueden llegar a operar a temperaturas de 1000 °C o mayores (Rabl, 1985), mientras los fluidos de trabajo se calientan típicamente de 300–700 °C (Pfahl y cols., 2017). La Figura 2.1 muestra un esquema de sistema de receptor central, o torre central, en donde los helióstatos están formados por nueve faceta cuadradas cada uno. Cada helióstato es orientado de tal forma que el rayo central del Sol sea reflejado por el centro de su faceta de referencia para interceptar el punto de destino deseado.

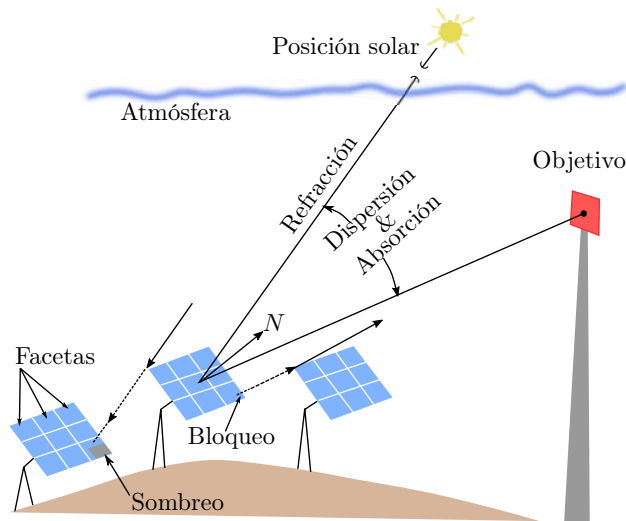


Figura 2.1: Sistema de receptor central. Adaptada de (Biggs y Vittitoe, 1979)

### 2.1. Conceptos básicos de concentración solar

La distancia entre el Sol y la Tierra es de  $1.495 \times 10^{11}$  m, la cual es muy grande comparada con sus dimensiones, de 107.5 y 11,771.6 veces el tamaño del Sol y la Tierra, respectivamente.

Al vector que llega del centro del Sol a un punto de la Tierra se le conoce como vector solar ( $\hat{s}$ ). La trayectoria del Sol puede ser descrita, desde un sistema de referencia local posicionado en la Tierra, a través de dos ángulos; el ángulo cenital  $\theta_z$  y el ángulo acimutal  $\gamma_s$ , como lo muestra la Figura 2.2(a), a esta trayectoria se le conoce como movimiento aparente del Sol.

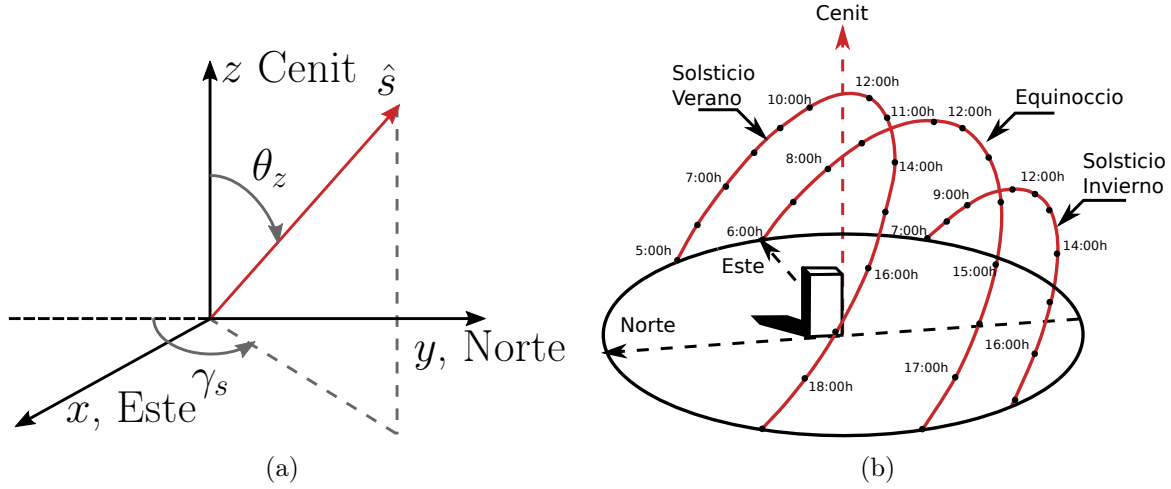


Figura 2.2: (a) Vector solar  $\hat{s}$  descrito por sus ángulos cenital  $\theta_z$  y acimutal  $\gamma_s$ , este último medido desde el sur. (b) Trayectoria del Sol en el plano de su trayectoria aparente.

El ángulo cenital  $\theta_z$ <sup>1</sup> indica la desviación del vector solar respecto a la vertical, mientras que el ángulo acimutal  $\gamma_z$  mide la desviación de su proyección en el piso, respecto al sur. Estos dos ángulos pueden ser calculados (Duffie y Beckman, 2013) mediante las ecuaciones (2.1) y (2.2).

$$\cos \theta_z = \cos \phi \cos \delta \cos \omega + \sin \delta \sin \phi \quad (2.1)$$

$$\tan \gamma_s = \frac{\sin \omega \cos \delta}{\sin \phi \cos \omega \cos \delta - \cos \phi \sin \delta} \quad (2.2)$$

El ángulo  $\phi$  se refiere a la latitud del lugar y tiene valor positivo cuando se mide hacia el norte del ecuador,  $\delta$  es el ángulo de declinación que se forma entre el plano de la eclíptica y el plano del ecuador de la Tierra y varía a lo largo del año desde 23.45 hasta -23.45 pasando por cero, para los solsticios de verano e invierno y los equinoccios, respectivamente. El ángulo  $\omega$  es el ángulo horario y describe la trayectoria del Sol en el plano de su trayectoria aparente, es medido a partir del mediodía solar, y es positivo hacia el oeste y negativo hacia el este, ver Figura 2.2(b).

Existen aplicaciones donde la posición del Sol debe determinarse con gran precisión, i.e., un heliostato de una torre central debe ser posicionado con una precisión de alrededor de  $0.05^\circ$  (Walraven, 1978) y la precisión del heliostato dependerá de la posición en el campo. En la

<sup>1</sup>Su ángulo complementario se conoce como ángulo de elevación y se representa por  $\alpha$ .

literatura se encuentra reportados diversos algoritmos para el cálculo de la posición solar, los cuales han sido contruidos bajo aproximaciones y suposiciones sobre la geometría terrestre. Dichos algoritmos pueden clasificarse en dos grupos. El primero es un grupo de algoritmos relativamente simple que dado el día del año, estiman los parámetros básicos de la posición solar, como la declinación solar o la ecuación del tiempo (Cooper, 1969; Spencer, 1971). El segundo esta formado por algoritmos más complejos (Walraven, 1978; Michalsky, 1988) que dada una posición calculan la posición del sol en la eclíptica, en la esfera celeste, y/o en las coordenadas locales. Típicamente el primer grupo de algoritmos son simples y adecuado para la mayoría de las aplicaciones, como por ejemplo, los paneles fotovoltaicos con seguimiento, los cuales no requieren una precisión extrema, incluso si el panel esta inclinado  $5^\circ$  sobre la normal respecto al vector solar, la salida se reduce un 0.4 % de la salida máxima. Sin embargo los sistemas de concentración solar requieren seguimiento preciso. Algunos autores utiliza 1 mrad ( $0.06^\circ$ ) en sus cálculos, limitados por las precisiones mecánicas (Michalsky, 1988).

En el presente estudio los cálculos se realizaron en tiempo solar, por lo que no es necesario utilizar alguno de estos algoritmos, sin embargo si se desea construir el arreglo de helióstatos propuesto es necesario tomarlos en cuenta, para mayor detalle ver Apéndice A.

## 2.2. Calidad óptica de la reflexión de un concentrador

Cuando la radiación incide en un concentrador, idealmente es reflejada de manera especular y lo hace de acuerdo a la ley de reflexión, es decir que el ángulo con el que incide  $\hat{s}$  respecto a la normal  $\hat{n}$  de una superficie, es el mismo que con el que se refleja  $\hat{t}$ ; y los vectores  $\hat{s}$ ,  $\hat{n}$  y  $\hat{t}$  siempre se encuentran en el mismo plano, como se muestra en la Figura 2.3.

Por lo que conociendo el vector solar  $\hat{s}$  y la posición del concentrador y del receptor, podemos calcular la dirección del rayo reflejado  $\hat{t}$ , y mediante la ecuación (2.3)<sup>2</sup> conocer la orientación de nuestra superficie, mediante la normal  $\hat{n}$ .

$$\hat{n} = \frac{\hat{t} - \hat{s}}{|\hat{t} - \hat{s}|} \quad (2.3)$$

Sabemos que  $-\hat{s} \cdot \hat{n} = \hat{t} \cdot \hat{n}$ , es decir que

$$|\hat{t} - \hat{s}| = 2 \cos \theta = -2\hat{n} \cdot \hat{s} = 2\hat{n} \cdot \hat{t} \quad (2.4)$$

Por lo que para calcular el vector reflejado a partir de la normal  $\hat{n}$  y el vector incidente  $\hat{s}$ , sustituimos la ecuación (2.4) en la ecuación (2.3) y despejamos el vector  $\hat{t}$ .

$$\hat{n} = \frac{\hat{t} - \hat{s}}{-2\hat{n} \cdot \hat{s}} \quad (2.5)$$

$$\hat{t} = -2\hat{n}(\hat{n} \cdot \hat{s}) + \hat{s} \quad (2.6)$$

---

<sup>2</sup>Hay que hacer notar que si el vector solar  $\hat{s}$  tiene la dirección opuesta a la indicada, entonces  $\vec{n} = \hat{t} + \hat{s}$ .

---

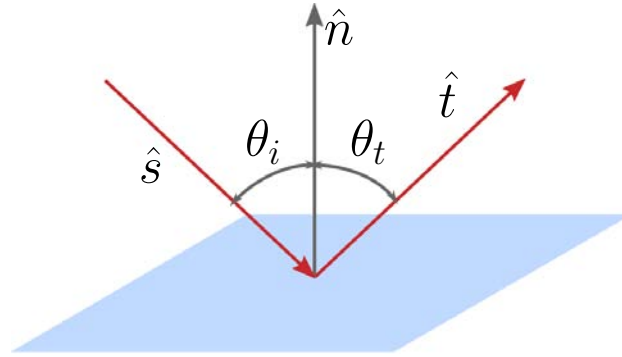


Figura 2.3: Reflexión especular.

### 2.2.1. Errores de superficies

La reflexión es una idealización basada en una superficie lisa perfecta. Las superficies reales tienen toda clase de irregularidades, desde rugosidades microscópicas hasta ondulaciones macroscópicas. Además, las superficies de los reflectores reales no tienen la forma exacta de diseño, y el seguimiento de colectores pueden tener errores de alineación. Finalmente el Sol en sí no es una fuente puntual si no que es una fuente extendida de radiación. Todos estos efectos contribuyen a ensanchar la zona focal del concentrador solar. Para considerar estos errores ópticos en los cálculos de interés lo adecuado es un análisis estadístico de estos errores ya sea general o promedio.

Al referirse a las superficies de reflexión es conceptualmente útil distinguir tres escalas como se ilustran en la Figura 2.4. La línea punteada muestra la forma de diseño. La curva sólida, el promedio sobre las ondulaciones. La diferencia entre la forma de diseño y la forma promedio representan errores ópticos de “gran escala”, causadas por gravedad, viento, esfuerzos de los materiales, o errores de manufactura. Las ondulaciones, con típicas longitudes de onda en el orden de centímetros a decímetros, representan errores de “escala media”, son cuantificados por “errores de pendiente”  $\beta$ . No es necesario especificar el error de pendiente en cada punto en la superficie de referencia, sino una distribución estadística del error de pendiente aplica para algunas (o quizás todas) las posiciones de la superficie del concentrador. Finalmente, existe una “escala pequeña”: rugosidades superficiales microscópicas, que causan dispersión (*scattering*) lejos de la dirección especular. No se muestra la superficie de pequeña escala, en su lugar se muestran sus efectos en la reflexión de los rayos inicialmente alineados que dan como resultado una reflexión de un cono designado como el “cono reflejado”.

Los errores de escala pequeña son propios de los materiales de la superficie de reflexión, mientras los errores de mediana y gran escala son debidos al sustrato y soporte estructural. Mientras los errores de pequeña escala son determinados al medir la propagación de un rayo colimado de luz después de que ha sido reflejado por una pequeña muestra del material. Los errores medianos y grandes pueden ser determinados por el mapeo del contorno del reflector usando métodos mecánicos o trazado de rayos láser.

Cada uno de estos errores ópticos puede ser caracterizado por una función de distribución

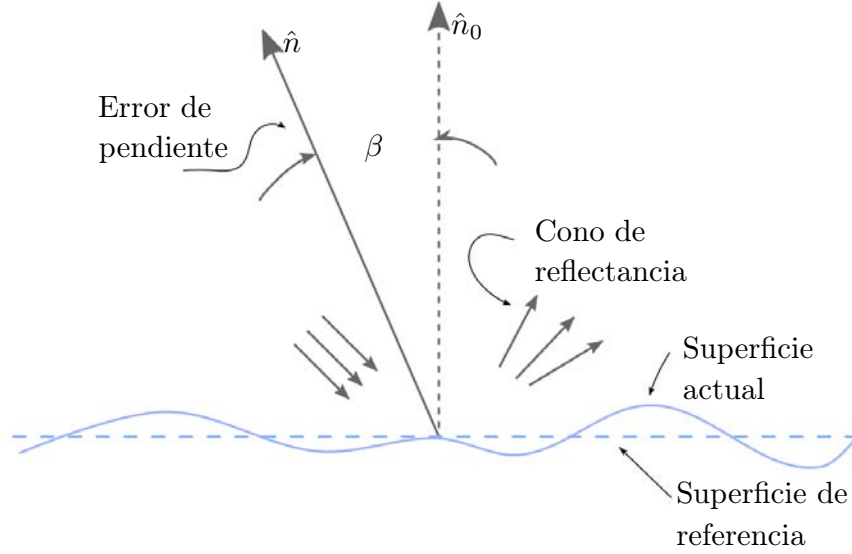


Figura 2.4: Características de una superficie de concentración a gran, mediana y pequeña escala.

estadística, donde solo importan las variaciones angulares, dado que la preservación de la imagen es irrelevante. Para datos disponibles parece que las distribuciones de error pueden ser bien aproximadas por una distribución normal o Gaussiana (Rabl, 1985).

### 2.2.2. Combinación de los errores ópticos

En un concentrador solar, varios factores estadísticos independientes contribuyen al error óptico: errores de pendiente<sup>3</sup>, ausencia de la especularidad ideal, errores de seguimiento, deformaciones y desplazamientos del receptor.

Conceptualmente, los errores de alineación y el seguimiento deben de ser manejados de forma diferente de los errores de pendiente. Es solamente cuando un campo entero de concentradores o en un tiempo promedio grande es que se considera una distribución y se obtiene un error de seguimiento, descrita por la desviación estándar  $\sigma_{\text{seguimiento}}$ .

Se considera generalmente que los errores pueden ser representados adecuadamente por distribuciones de probabilidad Gaussiana (Pettit, 1977). El error global es una combinación de los diferentes errores y su desviación estándar es una combinación en cuadratura de los errores individuales, ecuación (2.7).

$$\sigma_{\text{óptica}}^2 = \sigma_{\text{especular}}^2 + 4\sigma_{\text{pendiente}}^2 + \sigma_{\text{seguimiento}}^2 \quad (2.7)$$

$\sigma_{\text{pendiente}}$  es multiplicado por 2 debido a la ley de Snell; en reflectores Fresnel  $\sigma_{\text{seguimiento}}$  debe ser también multiplicada por 2.

El ancho total del rayo  $\sigma_{\text{total}}$  es obtenida al agregar el semi-ángulo del disco solar.

<sup>3</sup>El error de pendiente engloba los errores macroscópicos.

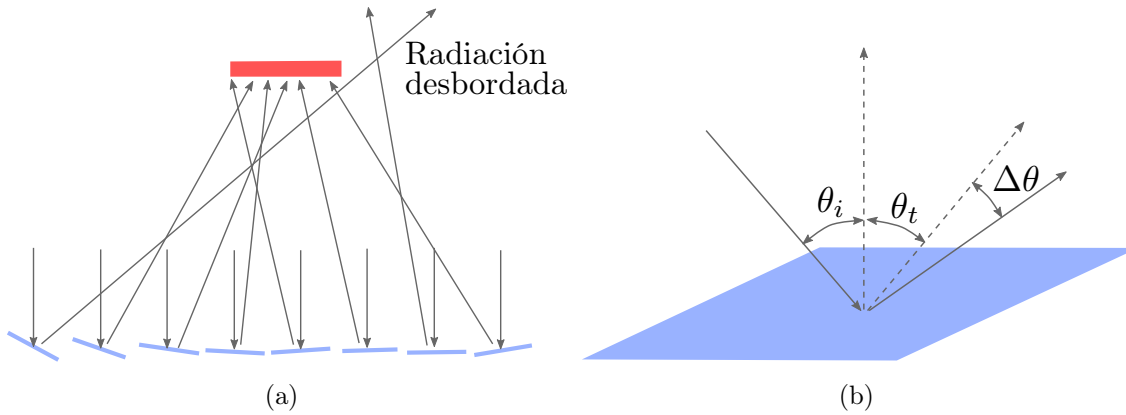


Figura 2.5: (a) error de pendiente por canteo y (b) error óptico global.

$$\sigma_{\text{total}}^2 = \sigma_{\text{óptica}}^2 + \sigma_{\text{sol}}^2 \quad (2.8)$$

### 2.2.3. La forma solar

Se denomina *forma solar* a la *distribución de intensidad* de los rayos solares que llegan a la superficie de la Tierra, como función de su ángulo con respecto al centro del disco solar. Esta forma no es constante en el tiempo y tampoco es independiente de la localización geográfica, ya que se ve afectada por las condiciones climáticas.

Las formas más utilizadas son el sol uniforme o “Pillbox”, el sol “Gaussiano” y los soles “reales”. Una manera de clasificar diferentes perfiles solares reales es a través de la *razón circunsolar* (CSR), que es el porcentaje de irradiancia directa que cae en la aureola solar, la llamada región circunsolar.

## 2.3. Concentración de radiación solar

La radiación puede ser transmitida, absorbida o dispersada al interactuar con un medio. Las interacciones de la radiación solar con la atmósfera terrestre da como resultado tres componentes de interés para la tecnología termosolar:

- Radiación normal directa ( $G_b$ ), es la radiación disponible del disco solar
- Radiación horizontal difusa ( $G_d$ ), radiación solar dispersada por la atmósfera terrestre
- Radiación horizontal global ( $G_h$ ), es la suma geométrica de la radiación directa y la difusa.

En la tecnología de concentración solar la componente que nos interesa es la radiación directa, debido a que es la componente que podemos direccionar al objetivo deseado. La

radiación solar tiene una densidad de energía baja respecto a los recursos convencionales; por lo que es necesario la concentración óptica de la radiación para incrementar la densidad y que sea útil para las aplicaciones técnicas. El diseño de la concentración solar está basado en el principio de la óptica de no imagen, es decir, que nos interesa transferir la radiación entre una fuente y un objetivo sin atender la forma de la imagen de la fuente. De tal forma que la concentración solar se define como la radiación incidente en una área de apertura  $A$  entregada a un receptor con área  $A'$ . La relación geométrica de concentración puede ser definida como

$$C = \frac{A'}{A} \quad (2.9)$$

Dado que no es posible tener temperaturas mayores a las de la fuente, existe un límite termodinámico de concentración debido a la segunda ley de la termodinámica, el valor máximo posible de concentración esta definido por la ecuación (2.10).

$$C \leq C_{ideal} = \frac{1}{\sin^2 \theta_a} \quad (2.10)$$

Donde  $\theta_a$  es el ángulo de aceptación, y la concentración máxima se alcanza cuando la fuente es el disco solar, es decir  $\theta_a = \Delta_s = 4.653$  mrad. En este caso se obtiene que para el caso de concentración puntual la concentración no puede ser mayor a  $C_{max} = 46,200$  soles (Leonardi y D' Aguanno, 2014).

## 2.4. Trazado de rayos

Una forma de calcular la irradiancia solar concentrada es la simulación de *trazado de rayos*, la cual se puede llevar a cabo mediante dos métodos:

- El método de *Monte Carlo*, basado en rayos al azar (método estadístico)
- El método de *Convolución*, basado en la convolución de distribuciones.

El método de Monte Carlo usa rayos al azar que vienen de una superficie, y luego determina cuáles de ellos llegan a una segunda superficie. La irradiancia sobre una superficie elemental es proporcional al número de impactos de rayos. En el caso de un concentrador con una reflexión, este algoritmo es usado dos veces, primero entre el Sol y la superficie reflectiva con una distribución energética correspondiente de la forma del sol, después entre la faceta del heliostato y el receptor con las leyes estadísticas del error de distribución relacionado con los defectos de las facetas. La precisión y el tiempo de cálculo aumentan con el número de rayos y la complejidad de la geometría. En el método de convolución (o cono óptico), los rayos reflejados de los espejos elementales son considerados como un cono de error calculado por la convolución de la distribución normal Gaussiana correspondiente a cada error (forma del Sol y los errores ópticos del heliostato). Una comparación sistemática muestra que se obtienen resultados muy similares con ambos métodos de trazo de rayos, aunque el Método de Monte Carlo es más versátil. La ventaja del método de convolución es una mayor velocidad de cómputo (García y cols., 2008).



Los primeros códigos diseñados para calcular el flujo solar concentrados fueron llevados a cabo a finales de los 70s, para el diseño de la planta de torre central *Solar One*. En Francia numerosos códigos fueron desarrollados antes y durante los experimentos de *Themis* pero sin una interfaz amigable Garcia y cols. (2008).

Los códigos para el análisis de concentración solar de torre central se pueden dividir en cinco grandes rubros (Ho, 2008): diseño óptico y funcionamiento del campo de helióstatos; funcionamiento del receptor central; transporte, intercambio, y almacenamiento del fluido de trabajo térmico; ciclo de potencia; y funcionamiento total del sistema. En particular el *diseño óptico y funcionamiento del campo de helióstatos* puede ser abordado por los siguiente códigos:

- UHC-IH/NS
- DELSOL
- HELIOS
- HFLCAL
- MIRVAL
- FIAT LUX
- SOLTRACE
- TONATIUH

Estas herramientas son de dominio publico, excepto FIAT LUX y HFLCAL. Y los códigos fuentes de SOLTRACE y programas adicionales (la versión de WINDELSOL desarrollada por el DLR alemán, extensiones para MIRVAL) no están disponibles actualmente. MIRVAL, FIAT LUX, SOLTRACE y TONATIUH dan una descripción detallada de la potencia reflejada del campo de helióstatos pero no incluye una conversión termodinámica u optimización del campo de helióstatos como lo hace UHC, DELSOL y HFLCAL. Aunque estos últimos contienen más aproximaciones en los métodos de solución, lo que les permite hacer evaluaciones más rápidas sobre el funcionamiento anual de grandes campos de helióstatos (Garcia y cols., 2008).

Aunque se menciona que varios de los códigos de trazados de rayos están disponibles, únicamente dos de ellos estan al alcance de cualquier lector: *Tonatiuh* y *SolTrace*. El primero de ellos, es un proyecto de código abierto que utiliza el trazado de rayos con el método de Monte Carlo para la simulación óptica del sistema de concentración solar. El desarrollo de Tonatiuh se llevo a cabo con la intención de disminuir el trabajo de diseño que implican los demás códigos, ya que regularmente se requiere explorar un gran numero de combinación factibles. En general Tonatiuh busca ser un software extremadamente accesible, amigable, fácil de adaptar, expandir y mantener. Para modelar un sistema de concentración es necesario una apertura de entrada, un volumen de control, y una serie de superficies; reales y virtuales (Blanco y cols., 2005).

**SolTrace** es un software que utiliza el trazado de rayos por el método de Monte Carlo, desarrollado por el NREL, puede usarse para analizar y diseñar sistemas ópticos de potencia de energía solar. En este software se pueden simular las principales tecnologías de conversión de energía térmica solar a electricidad y también los hornos solares. El sistema óptico está

---

organizado en etapas en un sistema de coordenadas globales: las etapas son secciones de la geometría óptica que exitosamente es golpeada por los rayos en su trayectoria del sol a el receptor final. Las etapas pueden ser ópticas (interacción física con los rayos) o virtuales (usadas para determinar la posición de los rayos o el mapa de flujo en cualquier parte del sistema sin interacción con él). Estas etapas están compuestas de elementos con su propia apertura, forma, y propiedades ópticas (error de especularidad, reflectividad, transmitividad y errores de pendiente). Cada elemento está descrito en un sistema coordinado asociado con su etapa. A diferencia de otras herramientas, la atenuación atmosférica y errores de seguimiento no se incluyen directamente pero pueden ser tomadas en cuenta en la reflectividad del espejo o en su error de pendiente, respectivamente (Wendelin y cols., 2013). Los resultados pueden ser mostrados y guardados como gráficas, mapas de flujo y gráficas de desempeño, además de tener una interface amigable con el usuario.

Ambos softwares, Tonatiuh y SolTrace, pueden ser programables. El primero mediante *javascript* el segundo en lenguaje *LK*. Entre ambos se eligió *SolTrace* porque su programación es muy similar a la programación en *C*.



### 3.1. Historia

El desarrollo de heliostatos inicio en 1975 a partir de un estudio financiado para el diseño de la primera generación de heliostatos. De este estudio y de una segunda generación se eligió el diseño de “heliostato montado en pedestal”, nombrado *McDonnell Douglas*, que mostraba tener una ventaja en precio sobre los otros y el cual fue seleccionado para la planta piloto *Solar One* en 1982, posteriormente el desarrollo del modelo *McDonnell Douglas* continuo su optimización y se encontró que la mejor forma de optimizarlo era incrementando el tamaño del heliostato a  $100 \text{ m}^2$ . Para reducir el costo aun más, *Sandia National Laboratories* propuso un diseño más grande de  $150 \text{ m}^2$ . La Fig. 3.1 muestra esta evolución, donde la tendencia son los heliostatos grandes para mejora el costo-beneficio debido a los componentes (Kolb y cols., 2007).

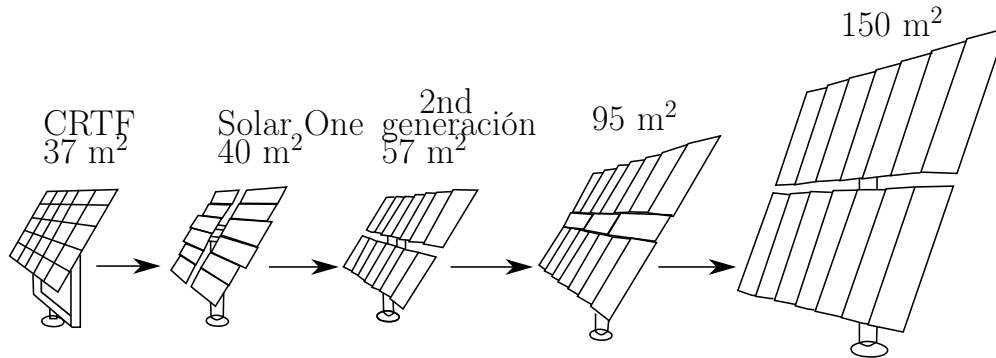


Figura 3.1: Evolución de reducción de costos. Tomada de Kolb y cols. (2007).

En 1980 la visión de construir heliostatos grandes para lograr menores precio por metro cuadrado fue reafirmado por un análisis realizado por Sandia (Kolb y cols., 2007).

Recientemente los heliostatos pequeños han estado proponiéndose para su uso en proyectos de torre en Europa y Australia. Los heliostatos pequeños tienen una óptica excelente (Kolb y cols., 2007). La idea es que estos mini-heliostatos puedan soportar menores cargas de viento que sus predecesores más grandes. Por lo tanto, el conjunto de los componentes mecánicos, incluyendo motores, pueden ser mucho más ligeros (en proporción al tamaño). La

reducción de las piezas electrónicas en la última década ha sido un argumento en favor de esta opción (Domínguez-Bravo y cols., 2016).

La primera planta experimental en Estados Unidos de América fue la *Solar One* (1982-1988) con 1,818 helióstatos montados en pedestal, cada helióstato era de  $39.1 \text{ m}^2$ , con un área total de reflexión de  $71,095 \text{ m}^2$  (Kolb y cols., 2007). Las pérdidas en el almacenamiento térmico debido a las pérdidas de calor en el sistema termocline de aceite/roca, el acoplamiento del receptor a la turbina, y las nubosidades dieron pie a el proyecto *Solar Two*, el campo de helióstatos fue usado nuevamente de 1996–1999. La planta *Solar Two* empleo sales fundidas, validando la tecnología de nitratos de sal, reduciendo los problemas técnicos y económicos y estimulando la comercialización de la tecnología de torre central. Con el almacenamiento térmico en sales fundidas la operación de la planta creció cerca del 25 % (Alexopoulos y Hoffschmidt, 2013).

En 2006 se reactivan los proyectos de torre central en Europa, proyectos impulsados desde el gobierno como lo fue la *PS10* (10 MW), la *PS20* (20 MW) y *Solar-Tres* (15 MW) en España, *Jülich* (1.5 MW) en Alemania y *Adrano* (200 kW) y *Thermis* (2.5 MW).

Las plantas piloto utilizan helióstatos con un área de reflexión que no supera los  $60 \text{ m}^2$  y la menor área de reflexión utilizada en una planta piloto fue de  $5 \text{ m}^2$ , en la planta de *CSIRO* de 0.2 MWe, ver Tabla 3.1. La planta de torre central de Jülich de 1.5 MW permitió asegurar la calidad de la tecnología existente, desarrollando procedimiento de pruebas y normas, así como la mejora de componentes y el desarrollo de nuevas técnicas (Alexopoulos y Hoffschmidt, 2013). La Tabla 3.1 muestra algunas plantas de torre central con su correspondiente potencia eléctrica así como el área de reflexión de por helióstato y el número de helióstatos que componen la planta.

Actualmente las plantas comerciales son la *PS10* (10 MWe), la *PS20* (20 MWe), *Sierra SunTower* (5 MWe), *Gemasolar* (19.9 MWe), *Ivanpah* ( $130 \times 3$  MWe), *Crescent Dunes* (110 MWe) y *Noor III* (150 MWe). Dentro de las plantas comerciales existen aquellas que utilizan áreas de reflexión grandes como la *PS10* o pequeñas como *Sierra SunTower*, ver Tabla 3.1. El almacenamiento térmico es de gran importancia para esta tecnología, *Gemasolar* fue la primera planta comercial en el mundo en utilizar la tecnología de almacenamiento de sales fundidas (Alexopoulos y Hoffschmidt, 2013) con hasta 15 horas de almacenamiento térmico.

El área de reflexión de los helióstatos de estas plantas comerciales son de 121, 115.7, 15.2 y  $1.14 \text{ m}^2$ , las plantas españolas *PS10*, *PS20*, *Gemasolar* y *Noor III*<sup>1</sup> mantienen el diseño de helióstatos grandes de  $115.7 \text{ m}^2$  y  $121 \text{ m}^2$ . Las plantas de Estados Unidos de América son más variadas en el uso del área de reflexión del helióstato utilizando áreas de  $1.14 \text{ m}^2$ ,  $15.2 \text{ m}^2$  y  $115.7 \text{ m}^2$  en las plantas de *Sierra SunTower*, *Ivanpah* y *Crescent Dunes*, respectivamente.

Lo que podemos observar en las plantas comerciales es que no existe un diseño estándar de helióstato, ejemplo de ello son las plantas más recientes *Noor III* (2017) e *Ivanpah* (2013). La primera de ella de una potencia de 150 MWe utilizando un área de reflexión por helióstato de  $115.7 \text{ m}^2$ , y por el otro lado *Ivanpah* una planta que se compone de tres torres cada una de ellas con una potencia de 130 MWe y un área de reflexión por helióstato de  $15.2 \text{ m}^2$ . La

---

<sup>1</sup>Es una planta ubicada en Marruecos pero desarrollada por SENER

Año	Nombre de la planta	Tipo	País	Área de reflexión del helióstato [m <sup>2</sup> ]	Número de helióstatos	Potencia [MW <sub>e</sub> ]
1981	Solar One	Piloto	EUA	39.1	1, 818	10
1983	CESA-1	Piloto	España	39.6	300	1.2
1983	Themis	Piloto	Francia	53.7	201	2.5
1996	Solar Two	Piloto	EUA	39.1	1,818	10
2001	Weizmann	Piloto	Israel	56	64	0.5
2006	CSIRO	Piloto	Australia	5	450	0.2
2006	PS-10	Comercial	España	121	624	10
2008	Jülich	Piloto	Alemania	8	2,150	1.5
2009	PS-10	Comercial	España	121	1,255	20
2010	Sierra SunTower	Comercial	EUA	1.14	24,000	5
2011	Gemasolar	Comercial	España	115.7	2,650	19.9
2013	Ivanpah	Comercial	EUA	15.2	173,500	390
2013	Crescent Dunes	Comercial	EUA	115.7	10,347	110
2017	Noor III	Comercial	Marruecos	115	7,400	150

Tabla 3.1: Tamaño del área de reflexión de los helióstatos que forman el campo de algunas plantas de concentración solar de Torre Central.

Figura 3.2 muestra las plantas de torre de central de manera histórica, así como la potencia en megawatts y el área de reflexión del helióstato de la planta asociado al tamaño de los círculos.

El diseño de helióstatos clásico está dominado por espejos, como superficie reflectora, posicionados en estructuras de acero y sistemas de seguimiento que garanticen alta precisión bajo cargas de viento y situaciones de esfuerzos térmicos. La Figura 3.3 muestra las partes típicas de un helióstato.

Las líneas actuales de investigación en el desarrollo de helióstatos esta basado principalmente en dos tendencias:

- La primera, explora el desarrollo de grandes helióstatos que tengas las mayores ventajas en componentes caros de alta-tecnología y beneficios para reducir el número de elementos y por lo tanto menor costo en el mantenimiento y operación
- La segunda tendencia, considera el desarrollo de helióstatos pequeños, con el beneficio de usar la manufactura de grandes volúmenes, configuraciones simples (sin necesidad de canteo, fácil instalación y mínima maniobra en sitio), garantizando el mismo nivel de calidad con componentes más baratos y producción en masa.

## 3.2. Funcionamiento

Los helióstatos grandes son frecuentemente divididos en secciones de espejos llamadas *facetas*. Cada faceta está desplazada relativamente respecto a las otras, de tal forma que

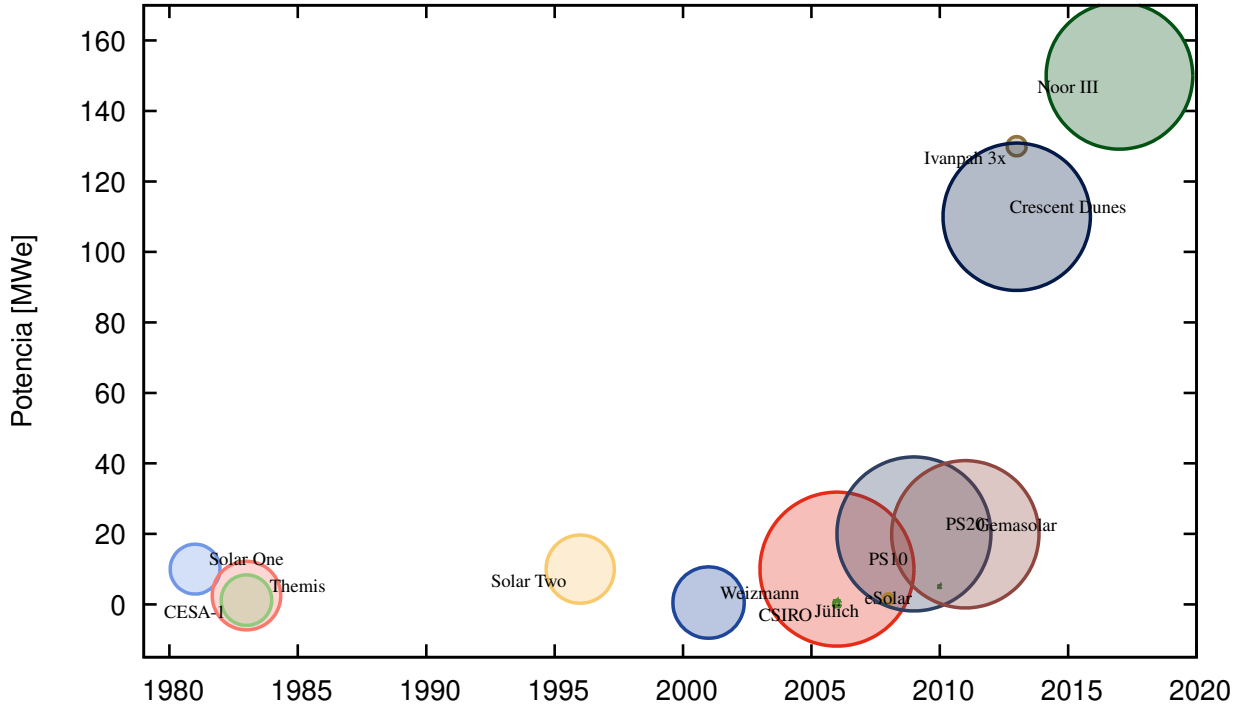


Figura 3.2: Se muestran las plantas de Torre Central en el tiempo comparadas así como la potencia eléctrica de las mismas, el círculo indica el área de reflexión de un helióstato que compone el campo. El helióstato con menor área de reflexión es el de Sierra SunTower (eSolar) con  $1.14 \text{ m}^2$  y el mayor es de  $121 \text{ m}^2$  que le corresponden a los helióstatos de la PS-10 y PS-20.

los rayos del Sol reflejados por estas se direccionen al receptor, es a lo que se conoce con el nombre de *canteo*, un canteo perfecto da como resultado el tamaño de una imagen mínima.

Un helióstato es típicamente un *concentrador fuera de eje*, es decir, que los rayos reflejados del concentrador inciden de manera no paralela al eje óptico del sistema y donde su distancia focal se ajusta a diferentes situaciones. En el caso de una *parábola* deja de existir un punto focal ideal y se transforma en una región focal. Existen dos distancias focales importantes que caracterizan este comportamiento, la *distancia focal tangencial* ( $f_t$ ) y la *distancia focal sagital* ( $f_s$ ). Estas dos distancias definen el tamaño de la región focal, que varía a lo largo del día y del año de acuerdo a las ecuaciones 3.1 y 3.2, respectivamente, en donde  $f_0$  es la distancia focal y el ángulo  $\mu$  esta formado por el vector solar  $\hat{s}$  y la normal de la superficie.

$$f_t = f_0 \cos(\mu) \quad (3.1)$$

$$f_s = \frac{f_0}{\cos(\mu)} \quad (3.2)$$

es decir, que mientras mayor sea el ángulo de incidencia, menor será  $f_t$  y más se agrandará

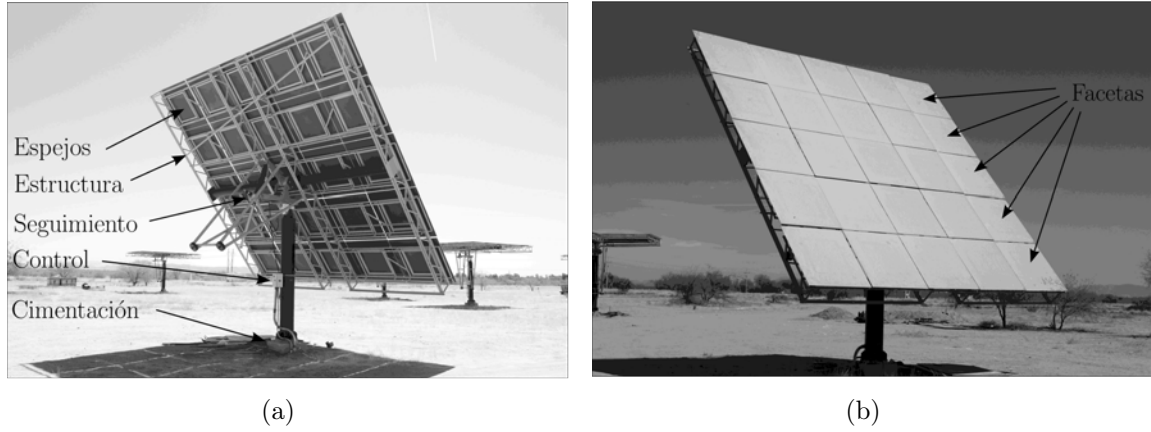


Figura 3.3: Helióstatos de Campo Experimental de Torre Central (CEToC). Sonora, México.

$f_s$ . Todos los demás puntos se enfocan en algún punto intermedio entre ambos.

La Figura 3.4(a) muestra lo que ocurre en la curvatura de un heliόstatos, debido al canteo. La distancia del heliόstatos al receptor se designa con la letra  $D$ . El heliόstatos se asume con un diámetro  $w$ . Los rayos tangenciales convergen a una distancia  $F_t$  y divergen con un ancho de  $h_t$  al momento de incidir en el receptor. Los rayos sagitales convergen a la distancia  $F_s$  y divergen al ancho  $h_s$ , cuando se ha cruzado el receptor. El punto de impacto para todos los demás rayos en el receptor serán en una elipse de mayor o menor diámetro de  $h_s$  y  $h_t$ .

El área de la imagen es minimizada (Rabl, 1985) mediante la ecuación 3.3:

$$F_{0,\text{optima}} = D \frac{2 \cos \mu}{1 + \cos^2 \mu} \quad (3.3)$$

Aunque cada heliόstatos pueda ser pensado para tener un espejo, los heliόstatos grandes son frecuentemente divididos en secciones de espejos llamadas *facetas*. Alinear una faceta consiste en orientarla apropiadamente, conocido como *canteo*, si la faceta puede ser alterada la alineación también consiste en introducir la curvatura apropiada en el espejo, lo que permite enfocarlo, para obtener un foco optimizado en el receptor. Se sabe que el canteo de heliόstatos tienen una influencia significativa en el funcionamiento anual del campo de heliόstatos (Buck y Teufel, 2009).

El canteo es dependiente del tiempo, ya que depende del ángulo entre el heliόstatos, el Sol y el receptor. Para la mayoría de los diseños de los heliόstatos la curvatura no puede variar, y los heliόstatos tendrán un enfoque o canteo perfecto solo para una o dos veces al año cuando el Sol está en la posición correcta, para todos los demás el heliόstatos producirá “aberraciones fuera de eje” de la imagen: distorsiones de la imagen real debido a la operación cuando el Sol no está en la posición de diseño para que el heliόstatos sea perfectamente enfocado (Alexopoulos y Hoffschmidt, 2013).

Las facetas de un heliόstatos se cantean para producir concentración, y algunas además se deforman para aumentar la concentración. La figura 3.5 muestra las aberraciones fuera de eje en un faceta, con curvatura o canteada, debidas a las incidencia del vector solar y a



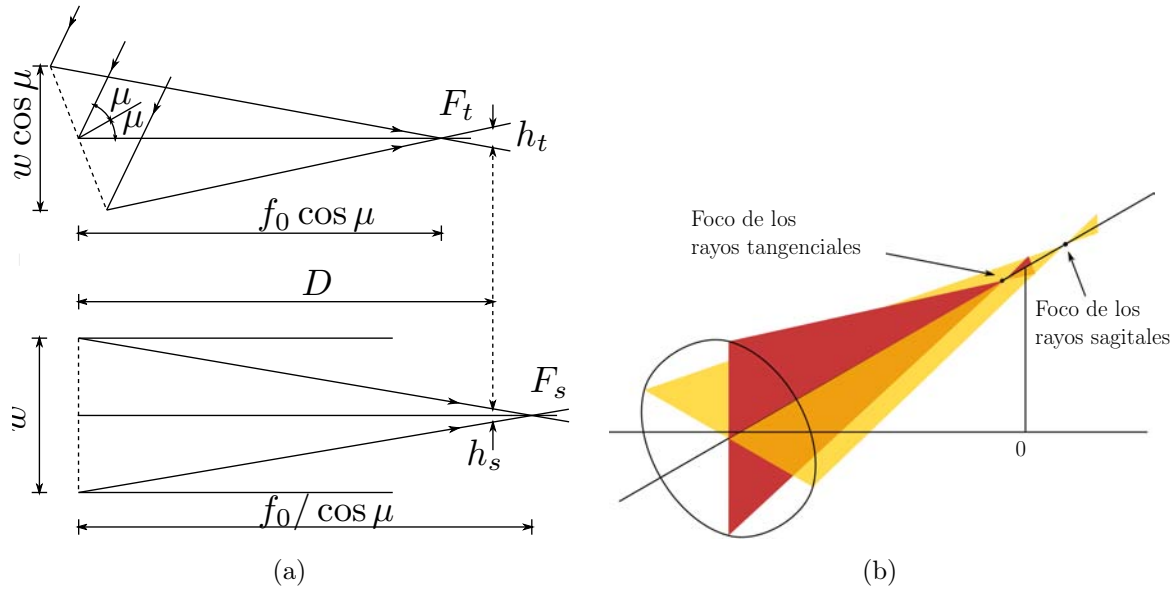


Figura 3.4: Rayos tangenciales y sagitales para un heliostato esférico a la distancia  $D$  del receptor, en la imagen de la izquierda. Rayos tangenciales y sagitales para una concentración fuera de eje de un heliostato esférico. Tomada Rabl (1985).

los errores ópticos. Cuando el número de espejos es pequeño una forma de incrementar la concentración significativamente es mediante una curvatura en los espejos, pero si el número de espejos es grande, 10,000 puntos focales, lo ganado en concentración debido a la curvatura del reflector es pequeña (Rabl, 1985).

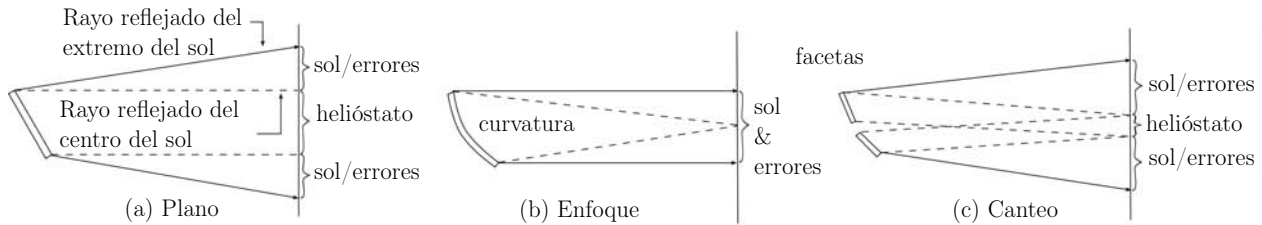


Figura 3.5: Esquema de un heliostato (a) plano, (b) enfoque y (c) canteo. Adaptada de Kistler (1986).

### 3.2.1. Canteo

Existen diferentes tipos de canteo, el más simple es el *canteo esférico en eje* que consiste en orientar la normal de cada faceta al radio  $R$  del doble de la distancia focal  $f_0$  deseada. La Figura 3.6 muestra este tipo de canteo esférico el cual funciona mejor para las horas y meses en que el sol tiene una declinación menor y peor cuando la declinación es máxima, al considerar un campo norte.

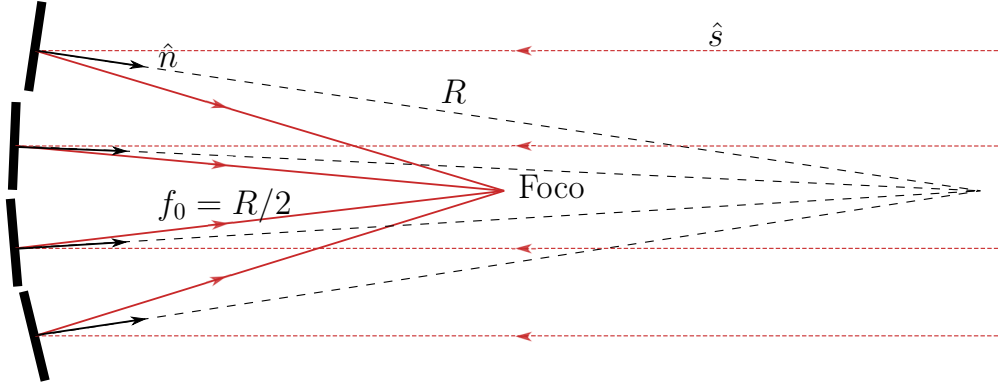


Figura 3.6: Canteo esférico.

Otra forma de canteo es seleccionar un día y una hora a la cual se desea que las facetas del heliostato reflejen el vector solar a la posición de interés. Para lo cual es necesario orientar las normales de las facetas de tal manera que el vector de reflexión de cada una de ellas incida en el receptor. Para poder calcular este canteo, primero es necesario determinar las rotaciones del heliostato, la cual está dada por la faceta central, por lo que es necesario calcular el valor de las rotaciones  $\beta$  y  $\gamma$  de la faceta central, Figura 3.7(b).

Para el calculo del ángulo cenital  $\theta_z$  del vector solar al medio día solar el valor de  $\cos \omega = 1$ , lo que nos lleva a simplificar la ecuación (2.1) mediante una identidad trigonométrica y obtener que  $\theta_z = |\phi - \delta|$ , por otro lado sabemos que al medio día solar  $\gamma_s = 0$ . A partir de la Figura 2.2(a) y el valor de  $\theta_z$  obtenido sabemos que el vector solar es  $\hat{s} = (\sin \theta_z, 0, \cos \theta_z)$ , es decir que para el canteo al medio día solar el vector solar esta dado por

$$\hat{s} = [\sin(\phi - \delta), 0, \cos(\phi - \delta)] \quad (3.4)$$

Dado que se conoce la posición de la faceta central es posible calcular la normal de la faceta central mediante la ecuación 2.3 y a partir de la normal los ángulos de rotación  $\beta$  y  $\gamma$ . Considerando la figura 2.2(a) encontramos que  $\beta = \arccos(n_3)$  y  $\gamma = \arctan n_2/n_1$ . Para determinar las rotaciones de cada una de las facetas a partir de un sistema de referencia local se ubica la posición de cada una de ellas, y dado que conocemos las rotaciones  $\beta$  y  $\gamma$  sabemos la posición de cada faceta en el sistema de referencias global a partir del cual podemos calcular la rotaciones  $\beta_i$  y  $\gamma_i$  de cada faceta mismas que nos indican el canteo del heliostato.

Desde un punto de vista práctico existen tres métodos de cantar heliostatos: alineación con el sol, alineación mecánico y alineación óptica. El **método de alineación con el Sol** ha sido ampliamente usado en la tecnología de Torre Central en los primeros años de desarrollo. En este método se asume que la posición del Sol prácticamente no cambia durante el proceso de alineación. Consisten en reflejar la radiación solar con el heliostato hacia una pantalla de observación, el blanco Lambertiano, y se van ajustando manualmente cada una de las facetas hasta conseguir formar una imagen lo más pequeña posible mediante la superposición de las imágenes de cada faceta. El proceso se debe hacer un tiempo corto, o en varios días

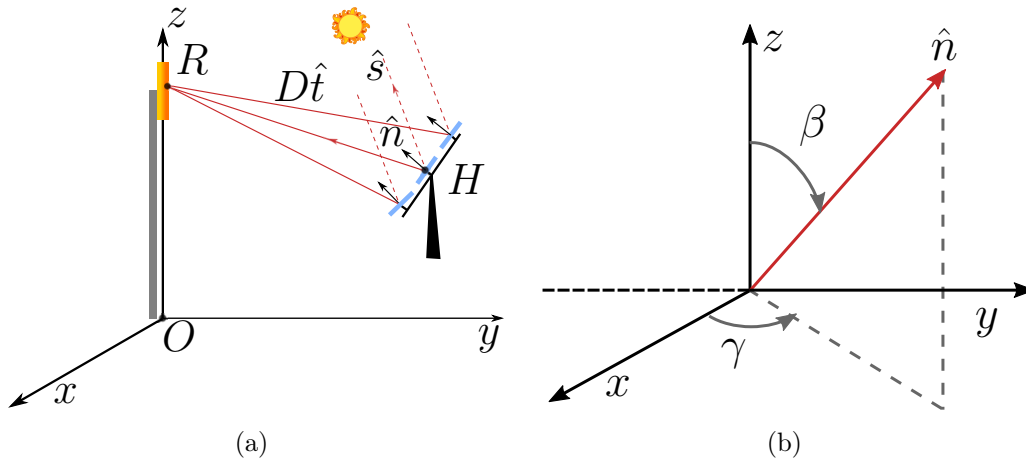


Figura 3.7: Canteo de un helióstato para un día en específico al medio día solar, del lado izquierdo. Sistema de referencia de la normal  $\hat{n}$  central del helióstato, del lado derecho.

consecutivos a la misma hora para que la posición solar no cambie mucho. La faceta central se alinea primero, y luego las facetas restantes son alineadas secuencialmente. Uno puede evaluar el desplazamiento de la posición de cualquier faceta comparando el *spot* observado en el receptor con el ideal.

Para la implementación de esta técnica no es necesario conocer la forma de la superficie del concentrador. La alineación es fácil y relativamente barata. Sin embargo, esta técnica solo puede usarse en sistema de torre central instalados y sólo en periodos despejados. Por otra parte, cuando la faceta se elige e inicia su alineación, las facetas restantes ya enfocadas deben ser cubiertas para prevenir que sus *spots* dificulten el ajuste de la faceta de interés. Aunque este método es cuantitativamente simple, toma más tiempo y requiere más mano de obra además de ser menos preciso que otras alternativas (Chavez y cols., 2012).

El **Método de alineación mecánico** consiste en utilizar una serie de *bloques* para ajustar la posición ideal de cada faceta, la precisión depende del proceso de manufactura de los bloques. Para el proceso de alineación el helióstato es posicionado en forma horizontal, y el plano de referencia es definido por el usuario. El método de alineación con bloques requiere un gran número de estos debido a que se requiere un juego por cada helióstato, o por cada serie de helióstatos a la misma distancia de la torre. Una mejora a este método ha sido el uso del *método del inclinómetro*, el cual sustituye los bloques por el uso de un inclinómetro, con una precisión de hasta 1.5 mrad, mayor que el uso de bloques (Ren y cols., 2014).

El **Método de alineación óptico**, como su nombre lo dice, se basa en la óptica para realizar el canteo mediante el uso del láser, cámaras y de patrones (Chavez y cols., 2012; Ren y cols., 2014). Por ejemplo, mediante el método de *scanning prism laser projection* puede obtener la desviación de la faceta en términos de dos ángulos, consiste de un láser, un divisor del haz, prismas y un detector de posición. Otros métodos son: el *Retro-reflexión de cámara*, el cual utiliza una cámara en la posición focal del helióstato y a partir de la cual pueden ser calculadas las direcciones normales de las facetas; la *Fotogrametría*, el cual

es una herramienta de análisis cuantitativo para determinar la forma de la superficie del concentrador solar a través de fotografías; la *Reflexión de franjas o Deflectometría* es una técnica de medición óptica originada en la industria automotriz y es virtualmente idéntica a la técnica de reflexión de franjas, en esta técnica un patrón de franjas es proyectado sobre una pantalla la cual es reflejada por un espejo y finalmente la imagen reflejada es tomada por una cámara digital, la distorsión de la superficie del espejo es calculada basándose en la distorsión de las franjas de la imagen.

Los métodos de alineación mecánico o óptico permiten realizar el canteo del helióstato en fabrica, conociendo previamente su posición dentro del campo respecto a la torre, para determinar los ángulos de canteo.

### 3.3. Diseño

Un helióstato es diseñado para reflejar la radiación directa a un objetivo con la menor dispersión posible, al menor costo. Para el diseño de la estructura es necesario tener en cuenta las dimensiones del helióstato, las cargas de viento, la precisión del seguimiento que se desea, las deformaciones durante la operación y las cargas de viento máximas. Los helióstatos están expuestos a las condiciones atmosféricas que se presenten en el campo, en donde las cargas de viento pueden causar fallas, si se excede el esfuerzo máximo permisible, pero también pueden fallar por fatiga, al tener fuerzas de viento fluctuantes.

En el diseño de un helióstato se utiliza usualmente espejo de vidrio bajo en acero como material reflectante, de entre 3–4 mm de espesor. El vidrio es la opción más adecuada debido a que es relativamente barato, durable y tiene alta reflectancia. La mayoría de los helióstatos consiste de una estructura de acero y facetas de vidrio. Sin embargo estos dos materiales tiene diferente dilatación térmica, para unirlos existen básicamente dos métodos: mediante una serie de patas con almohadillas o estampado del vidrio en acero.

No existe un tamaño ideal de helióstato (Pfahl y cols., 2017), existe una gama amplia en el mercado. La ventaja de tener helióstatos grandes son la cantidad de partes y una menor cimentación para un área equivalente de espejos. La desventaja sin embargo es que la instalación y el mantenimiento es más difícil en grandes helióstatos, como el remplazo de actuadores o espejos grades (Alexopoulos y Hoffschmidt, 2013). La principal ventaja de los helióstatos pequeños es la reducción de las cargas de viento.

El diseño del acomodo de los helióstatos en el campo es un tema complejo. La decisión respecto a la mejor posición para la localización de los helióstatos relativa al receptor y que tan alto se debe de colocar el receptor por arriba del campo constituye un problema de múltiples fases, en donde el costo y la minimización de mecanismos, para los helióstatos, tienen una gran cantidad de variables, lo que se resuelve por procesos interactivos. La Figura 3.8 muestra varias posiciones del acomodo del campo básicas de acuerdo a la localización de la planta de potencia. En la del *Hemisferio Norte*, la posición del Sol está en el Sur de la planta y el campo debe de ser colocado al norte de la torre, en el *Hemisferio Sur* es de manera opuesta. Cerca del ecuador, donde el Sol pasa justo por encima, se puede usar un *Campo Circundante* en combinación con un receptor abierto. En un sistema con concentrador secundario se debería

usar un *Campo elíptico* (Alexopoulos y Hoffschmidt, 2013).

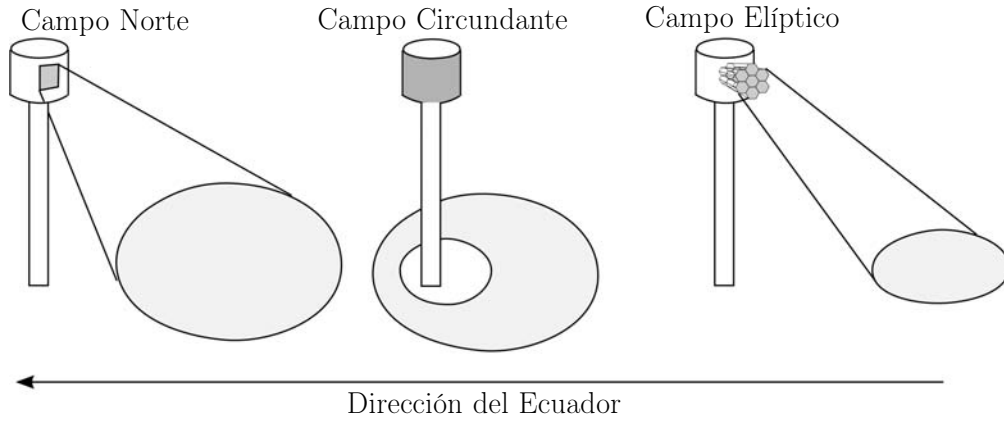


Figura 3.8: Diseño del campo de helióstatos dependiendo de la localización. Adaptada de Alexopoulos y Hoffschmidt (2013).

Al elegir un campo de helióstatos existen muchos tipos de pérdidas que hay que considerar. No solo están las pérdidas ópticas, las pérdidas por coseno y las pérdidas debidas al sombreado, bloqueo, derrame, y atenuación atmosférica, también existen consideraciones técnicas, como la reflectividad del espejo, defectos de superficie en los espejos, precisión del seguimiento, carga de viento, y oscilaciones de la torre (debido a la carga de viento), así como la relación de falla de los helióstatos. La normal de la superficie de los espejos de los helióstatos prácticamente nunca están de frente al Sol, excepto en los días del año cuando en el medio día solar la posición solar está detrás del receptor (en ese caso particular el helióstato debería ser sombreado por la sombra de la torre). Durante el curso de cualquier día la orientación de los ángulos cenit y azimut del helióstato están cambiando cada pocos segundos para garantizar el seguimiento de la trayectoria del Sol. Así que, matemáticamente, el área efectiva del espejo es menos que la área real del espejo. La reducción del área de espejo efectiva es proporcional al coseno del ángulo de incidencia. Las pérdidas por el efecto coseno son las pérdidas más significativas del campo de los helióstatos (Alexopoulos y Hoffschmidt, 2013).

El efecto de bloqueo ocurre para cualquier helióstato que tiene un vecino en frente o a los lados frontales. El bloqueo significa que el helióstato no puede reflejar una porción de rayos de luz al receptor por que el helióstato de enfrente bloquea parte de estos, es decir, que una parte trasera del helióstato es iluminado. Cuando el ángulo de elevación del Sol es pequeño y además un helióstato se encuentra lejos de la torre, es cuando ocurre el mayor bloqueo. El bloqueo se puede disminuir tanto como sea posible al incrementando la distancia entre los helióstatos un  $\Delta x$ , aunque se puede eliminar el bloqueo como lo indica la ecuación (3.5) no es conveniente ya que las  $\Delta x$  se va incrementa conforme el helióstato esta más retirada de la torre y se afectan otros parámetros ópticos como la atenuación atmosférica (Alexopoulos y Hoffschmidt, 2013).

$$\Delta x \approx x \frac{z_s}{z_T} \quad (3.5)$$

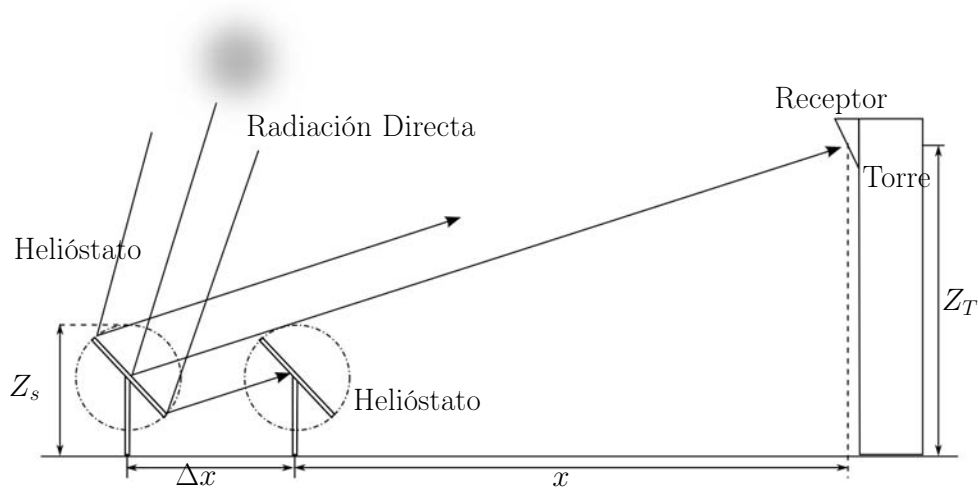


Figura 3.9: Bloqueo. Adaptada de Alexopoulos y Hoffschmidt (2013).

El sombreado es similar al bloqueo excepto que aquí los rayos de luz, inciden en la superficie del espejo, proyectando una sombra sobre el heliostato detrás o hacia los lados posteriores. La Figura 3.10 muestra el sombreado parcial de un espejo.

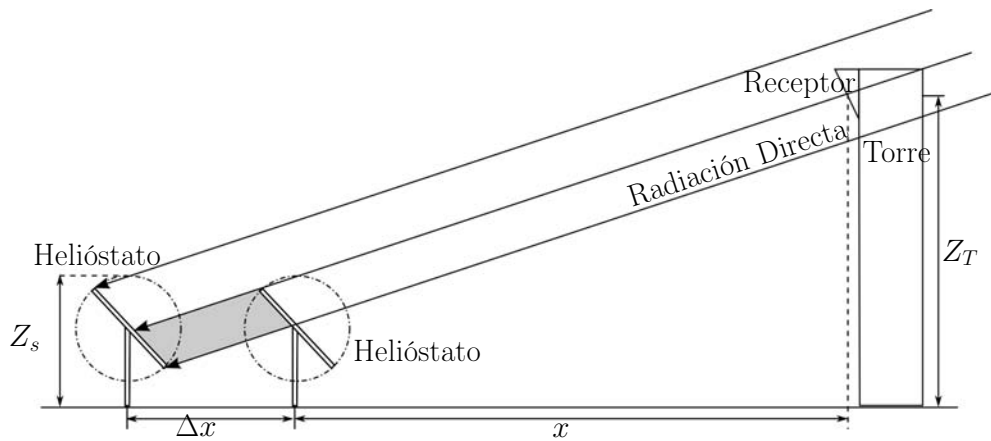


Figura 3.10: Sombreado. Adaptada de Alexopoulos y Hoffschmidt (2013).

Otras pérdidas son debidas al desbordamiento y la atenuación atmosférica. El desbordamiento es la porción de radiación reflejada por los heliostatos que cae fuera de la superficie del receptor, y depende de factores como la calidad del espejo y la precisión del seguimiento. La atenuación atmosférica es debida a la dispersión de la radiación reflejada por el aire que va desde el heliostato hacia el receptor, la cual depende de la distancia de heliostato al receptor.



---

## Capítulo 4

# Seguimiento solar elevación-Fresnel

---

### 4.1. Configuración de seguimiento solar

En la Figura 4.1 se muestra el esquema del seguimiento propuesto. Como se puede observar, todas las facetas rotan alrededor de unos ejes paralelos, que son inclinados a un ángulo  $\xi$  con respecto a la horizontal mediante un mecanismo de elevación común. Todas las facetas giran a la misma velocidad angular  $\mu$ , pero tienen diferente rotación angular  $\psi_j$ . De tal forma que la configuración de rotación es equivalente a un concentrador lineal Fresnel, pero con un movimiento de inclinación adicional. Los ángulos al plano perpendicular son ilustrados en la Figura 4.2.

En este esquema de seguimiento, la faceta central tienen un seguimiento solar preciso en cualquier instante de tiempo, y el resto de las facetas la siguen. La rotación de los ángulos de cada faceta está relacionada al ángulo de la faceta central, de la siguiente forma

$$\psi_j(t) = \psi_0(t) + \Delta\psi_{c,j} \quad (4.1)$$

donde  $\psi_j(t)$  y  $\psi_0(t)$  son los ángulos de la faceta  $j$  y 0, respectivamente, y  $\Delta\psi_{c,j}$  es la diferencia angular constante de la faceta  $j$  con respecto a la faceta 0; i.e., el equivalente al ángulo de canteo. La velocidad angular es idéntica para todas las facetas

$$\frac{d\psi_j}{dt} = \frac{d\psi_0}{dt} = \mu \quad (4.2)$$

La diferencia angular es determinada cuando todas las facetas están perfectamente apuntando<sup>1</sup> al punto focal deseado para un “día de canteo” al mediodía solar. En dicho momento, todas las facetas comparten la misma inclinación

$$\xi_j(t) = \xi_0(t) = \xi(t) \quad (4.3)$$

El objetivo del modelo de seguimiento propuesto es acoplar varios heliostatos pequeños, para operar con sólo dos actuadores; uno de ellos para proveer la elevación  $\xi$  y el segundo para el movimiento común de velocidad angular  $\mu$ . De esta forma, el movimiento independiente

---

<sup>1</sup>Estrictamente hablando en ningún momento las facetas laterales inciden en el objetivo, ya que comparten el ángulo  $\xi$ , pero si muy próximo a él.



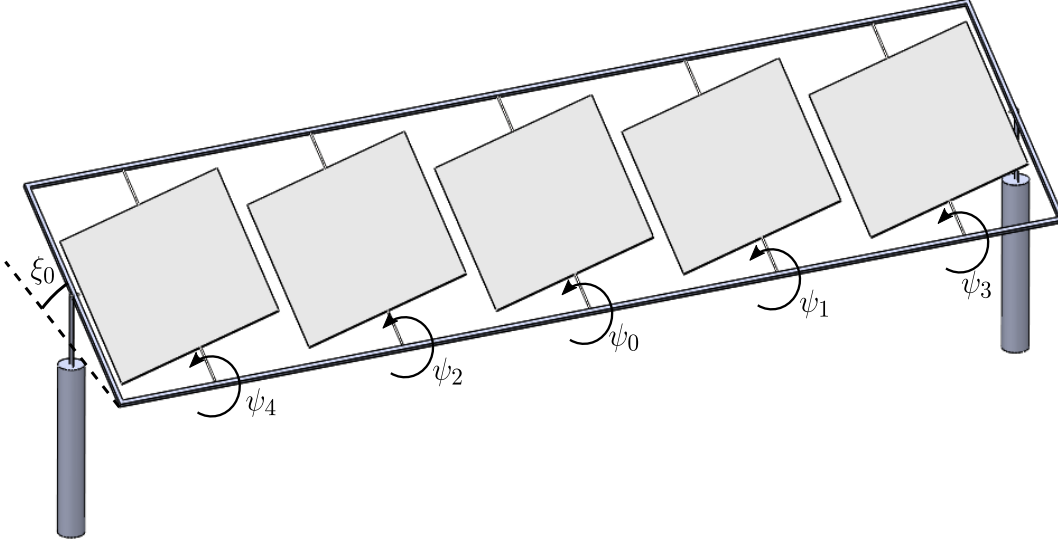


Figura 4.1: Sistema de seguimiento general.

de cada heliostato es sacrificado al convertirlo en un movimiento común de las facetas en un gran sistema. La desventaja es que el apuntamiento de todos, excepto el de la faceta central, será imperfecto en cualquier momento, excepto el día del canteo al mediodía.

## 4.2. Metodología

Como se mencionó con anterioridad, el seguimiento global del sistema es gobernado por los ángulos de la faceta central. Si el arreglo considerado tiene un número par de facetas colocadas simétricamente, no existe una faceta central, el movimiento es calculado para una faceta ficticia localizada exactamente en el centro del arreglo.

En cualquier caso, primeramente es necesario determinar los ángulos de esta faceta central. La Figura 4.3(a) muestra un sistema de referencia global donde  $\hat{s}$  es el vector unitario que apunta al Sol,  $\hat{n}$  es el vector normal de la superficie reflectora,  $\hat{t}$  es el vector unitario de reflexión que apunta a  $R$ , objetivo ubicado en el receptor. El origen de este sistema de referencias es el centro del heliostato, y la torre está localizada al sur.

Para una posición y una hora en específico, si el vector solar  $\hat{s}$  y la posición relativa del objetivo es conocida, el vector unitario  $\hat{t}$  puede ser calculado. La faceta central debería ser orientada para satisfacer la ley de reflexión

$$\hat{n} = \frac{\hat{s} + \hat{t}}{\|\hat{s} + \hat{t}\|} \quad (4.4)$$

Donde tenemos que  $\hat{n} = \hat{n}_0$  y  $\hat{t} = \hat{t}_0$ , son los vectores correspondientes a la faceta central.

Para orientar la faceta central con el mecanismo de elevación-Fresnel, es necesario rotar los ángulos  $\xi_0$  y  $\psi_0$  como lo muestra la Figura 4.3(b): primero una rotación mediante el ángulo

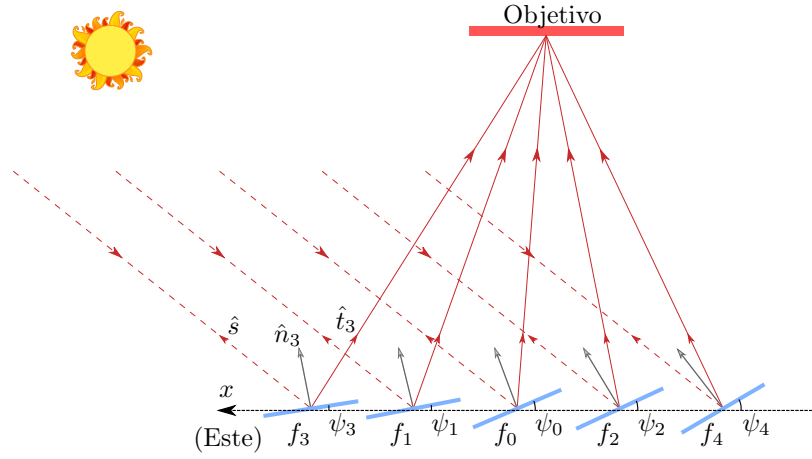


Figura 4.2: Vector de incidencia y reflexión en el plano de rotación de las facetas.

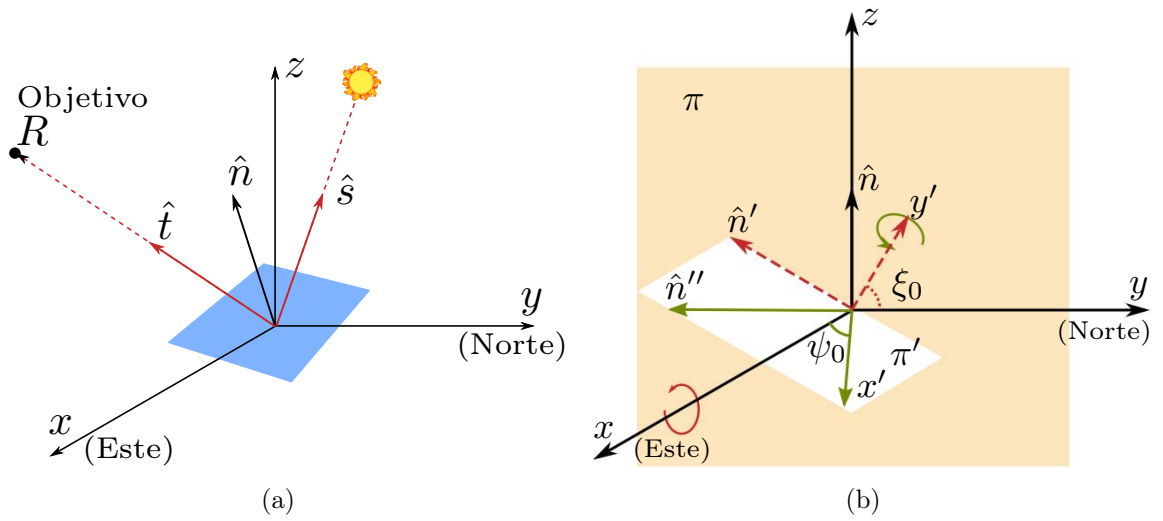


Figura 4.3: (a) Sistema de referencia global en donde se observa el vector solar  $\hat{s}$ , el vector normal  $\hat{n}$ , y el vector de reflexión  $\hat{t}$ . (b) Rotaciones  $\xi_0$  y  $\psi_0$  de la faceta central para el seguimiento de un arreglo de heliostatos.

$\xi_0$  alrededor del eje  $x$  en el plano  $\pi$  (plano  $yz$ ), seguida de una segunda rotación por el ángulo  $\psi_0$  alrededor del eje  $y'$  en el plano  $\pi'$  (plano  $\hat{n}'x$ ). Así, en el sistema de referencia global  $xyz$  de la Figura 4.3(a), el vector normal puede ser expresado en términos de dichos ángulos como

$$\hat{\mathbf{n}}_0 = (\sin \psi_0, -\cos \psi_0 \sin \xi_0, \cos \psi_0 \cos \xi_0) \quad (4.5)$$

De esta ecuación podemos deducir que las componentes del vector normal  $\hat{\mathbf{n}}_0 = (n_{0x}, n_{0y}, n_{0z})$  están relacionadas con los ángulo  $\xi_0$  y  $\psi_0$  por

$$\psi_0 = \arcsin(n_{0x}) \quad (4.6)$$

$$\xi_0 = \arccos\left(\frac{n_{0z}}{\cos \psi_0}\right) \quad (4.7)$$

Con el análisis previo, los ángulos de rotación necesarios para orientar la faceta central  $\psi_0$  y  $\xi_0$  son calculados como función del tiempo, de acuerdo con la evolución requerida del vector normal, calculado por la ecuación (4.4). Ahora únicamente se requiere calcular los ángulos de canteo necesarios de las facetas laterales  $\Delta\psi_{c,j}$ .

El canteo se realiza al elegir un *día de canteo*, y orientar las facetas para que todos los rayos reflejados incidan en el centro del objetivo al mediodía solar. Siempre es posible lograrlo al ajustar los ángulos  $\Delta\psi_{c,j}$  únicamente, porque el sistema llega a ser equivalente a un concentrador lineal en un momento del día. Así, una vez que el ángulo  $\psi_0$  es fijado y los ángulos de canteo  $\Delta\psi_{c,j}$  son obtenidos obligando que la ecuación (4.4) se cumpla para todas las facetas al mediodía solar para este día específico, con  $\hat{\mathbf{n}} = \hat{\mathbf{n}}_{j,\text{noon}}$  y  $\hat{\mathbf{t}} = \hat{\mathbf{t}}_{j,\text{noon}}$ . Análogamente, con la ecuación (4.6) podemos obtener

$$\Delta\psi_{c,j} = \arcsin(n_{jx,\text{noon}}) \quad (4.8)$$

Una vez que tenemos definido el canteo del arreglo de helióstatos, la simulación del seguimiento puede llevarse a cabo. Este es realizado al variar continuamente el vector solar  $\hat{\mathbf{s}}(t)$ , de acuerdo a las ecuaciones de la posición del Sol, y obteniendo los ángulos de seguimiento de la faceta central de las ecuaciones (4.5), (4.6) y 4.7. Entonces los ángulos  $\psi_j$  del resto de las facetas son obtenidos de la ecuación (4.1). Una vez que todas las facetas están rotando de acuerdo a estos ángulos, sus normales pueden ser obtenidas mediante la ecuación (4.5).

Para evaluar la calidad del seguimiento, calculamos el vector de reflexión desde el centro de cada faceta, para determinar el punto de impacto en el objetivo. El vector de reflexión de la faceta  $j$ ,  $\hat{\mathbf{r}}_j$ , es obtenido mediante la ecuación (4.9).

$$\hat{\mathbf{r}}_j = 2(\hat{\mathbf{s}} \cdot \hat{\mathbf{n}}_j)\hat{\mathbf{n}}_j - \hat{\mathbf{s}} \quad (4.9)$$

El punto de impacto en el objetivo  $\mathbf{T}_j$  es obtenido como

$$\mathbf{T}_j = \mathbf{f}_j + D_j \hat{\mathbf{r}}_j \quad (4.10)$$

donde  $\mathbf{f}_j$  es la posición de la faceta  $j$ , y  $D_j$  es la distancia al punto de impacto. Esta distancia es calculada para un objetivo vertical como

$$D_j = \frac{C_y - f_{jy}}{r_{jy}} \quad (4.11)$$

donde  $C_j$ ,  $f_{jy}$ , y  $r_{jy}$  son las componentes en  $y$  del centro del plano objetivo, el centro de la faceta y el vector de reflexión. En el caso de un objetivo horizontal, es obtenido

$$D_j = \frac{C_z - f_{jz}}{r_{jz}} \quad (4.12)$$

Para un receptor diferente al vertical u horizontal, se define un plano con una normal  $\hat{\mathbf{n}}_p$  donde el objetivo se especifica con el punto  $C$  y cualquier punto  $R_j$  dentro del plano define el vector  $\mathbf{u}_j = \overrightarrow{CR_j}$ . El plano es definido por la ecuación  $\hat{\mathbf{n}}_p \cdot \mathbf{u}_j = 0$ . Si el vector  $\mathbf{T}_j = \overrightarrow{PR_j}$  y  $\mathbf{f}_j = \overrightarrow{OP_j}$ , donde  $P_j$  es la posición de la faceta, entonces mediante la ecuación (4.10) y por la definición del plano, la distancia  $D_j$  es calculada de la siguiente manera.

$$D_j = \frac{\sum_{i=1}^3 n_{p,i}(C_i - P_{j,i})}{\sum_{i=1}^3 n_{p,i}r_{j,i}} \quad (4.13)$$

y mediante la ecuación (4.10) el punto  $R$  en el plano es conocido.

Para analizar la desviación del vector de reflexión  $\hat{\mathbf{r}}_j$  de su valor ideal  $\hat{\mathbf{t}}_j$ , obtenemos el error angular  $\varepsilon$  como

$$\varepsilon = \arccos(\hat{\mathbf{r}}_j \cdot \hat{\mathbf{t}}_j) \quad (4.14)$$

Finalmente, para estudiar el efecto del seguimiento imperfecto que la distribución de flujo produce por el heliostato en el objetivo, se utilizó el software **SolTrace**. De los datos obtenidos con el programa en **C**, donde se llevó a cabo la simulación del seguimiento elevación-Fresnel, se obtiene el vector solar  $\hat{\mathbf{s}}$ , la posición de cada una de las facetas y sus normales  $\hat{\mathbf{n}}$  para cada paso de tiempo. Estos son los datos de entrada usados en **SolTrace** mediante el uso de un *script* *LK* (Dopos, 2017). Para la simulaciones realizadas se utilizó un sol Gaussiano, un error de pendiente de 2.64 mrad y  $5 \times 10^6$  rayos. En el receptor se utilizó una malla de  $50 \times 50$ .

La radiación directa utilizada para las simulaciones, para diferentes horas y días, se calculó a partir de un modelo de cielo claro (Dunham y cols., 2013), utilizando:

$$G_b = G_0 \tau^{\text{AM}^{0.678}} \quad (4.15)$$

donde  $G_0$  es la constante solar de  $1,367 \text{ W/m}^2$ ,  $\tau$  es la transmitancia atmosférica, que para condiciones de cielo claro tiene un valor de 0.7, y la masa de aire es obtenida como función de la altitud del lugar, a partir de la siguiente ecuación:

$$\text{AM} = \frac{\exp(-0.0001184 h_{\text{altitud}})}{\cos \theta_z + 0.5057(96.080 - \theta_z)^{-1.634}} \quad (4.16)$$

$\theta_z$  es el ángulo cenital del vector solar, y  $h_{\text{altitud}}$  la altura sobre el nivel del mar.



---

## Capítulo 5

# Programación

---

El código principal para la simulación del arreglo *elevación-Fresnell*, del cual previamente se ha descrito la metodología en la Sec. 4.2, fue programado en C y lleva por nombre `erun`. El programa `erun` es un archivo ejecutable compilado a partir del archivo principal `erun.c`, el cual llama a los *archivos de cabecera* `base.h`, `vectorial.h`, `ecu.h` y `db.h`. Los archivos de cabecera le indican al archivo principal la existencia de las funciones necesarias para la correcta compilación, dichas funciones se encuentran definidas en los correspondientes archivos `base.c`, `vectorial.c`, `ecu.c` y `db.c`. El programa `erun` tiene la capacidad de generar la simulación de un arreglo en un día o a lo largo del año para los días 7, 14, 21 y 28 de cada mes.

Los datos generados a partir de `erun` son datos de entrada para la simulación del trazado de rayos en `SolTrace`, el cual permite realizar `script LK` donde es posible programar el modelo y los parámetros necesarios para la simulación. La gestión de archivos se realizó mediante `shell script` el cual permite, entre otras cosas, gestionar archivos y programas. El análisis de datos se realizó con el programa R que tiene la capacidad de manipular una gran cantidad de datos.

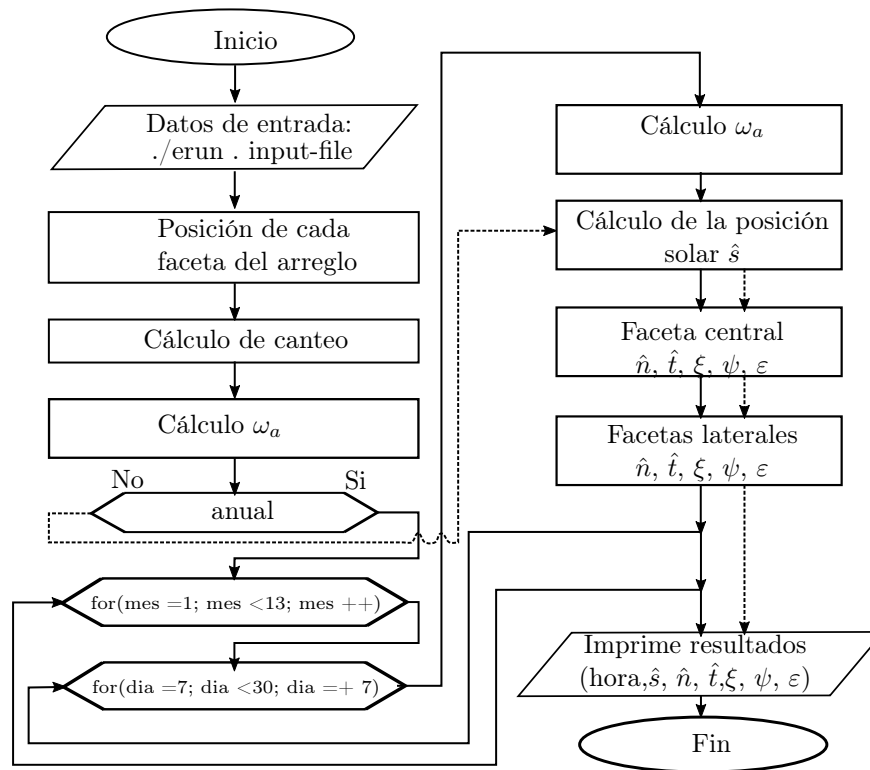
### 5.1. Programación en C

El archivo principal `erun.c`, ver Sec. C.1.1, lee el archivo `file.in` y llama a los archivos de cabecera. Los archivos `base.h` y `vectorial.h` contienen las definiciones de funciones básicas como la conversión de grados a radianes, y viceversa, cálculo de máximos y mínimos, declaración de arreglos, y operaciones vectoriales. El archivo `ecu.h` define las funciones que realizan las operaciones necesarias para simular el seguimiento elevación-Fresnel y calcular la normal  $\hat{n}$  de la faceta  $i$ -ésima, los ángulos  $(\xi, \psi)$ , el impacto en el receptor y el error angular  $\varepsilon$ . Por último el archivo `db.h` define las funciones necesarias para calcular el vector solar  $\hat{s}$  en tiempo solar. La Figura 5.1 muestra el diagrama de flujo del programa `erun.c`.

En el archivo `archivo.in` se indican los parámetros para realizar la simulación, como es: la latitud, la longitud<sup>1</sup>, la posición del receptor, las dimensiones de las facetas, la separación

---

<sup>1</sup>En realidad no es necesario indicar la longitud, ya que la simulación se realiza en tiempo solar, sin embargo el programa tiene la capacidad de calcular la posición solar en tiempo estándar, es por eso que contiene ese parámetro.

Figura 5.1: Diagrama del programa `erun.c`.

entre ellas, el tamaño del arreglo, el día de canteo y el tipo de simulación. Hay que hacer notar que el sistema de referencia global, por facilidad al programarlo,  $xyz$  es  $x$ –Este,  $y$ –Norte y  $z$ –Cenit tiene origen en el centro del arreglo de helióstatos. En el Cuadro 5.1 se muestra un archivo de entrada típico.

Cuadro 5.1: Archivo de entrada de datos `file.in` para el programa `erun`.

```

1  /*****/
2  /* DATOS DE ENTRADA*/
3  /*****/
4  21 12 2017      // dia mes año
5  12 00 00        // hora min seg
6  0               // tiempo solar/estándar (0/1)
7  0               // horario de verano no/si (0/1)
8  29.028452       // latitud +N [grad]
9  111.145475      // longitud +O [grad]
10 105             // longitud estandar [grad]
11 /*****/
12 /* HELIOSTATOS (0, 0, 0)/RECEPTOR*/
13 /*****/
14 0.0 -26.25 35.0 // posición del receptor
15 1.0 1.0 0.2 32  // helióstato largo/alto/gap/facetas
16 0               // rotación del helióstato 0/1, 0 alineado al eje
17 1               // Receptor vertical(1)/horizontal(2)
18 12 21 03        // hrs dia mes de canteo
19 0.0             // desplazamiento del apuntamiento de facetas laterales
20 1               // simulación anual (0/1)

```

A partir de los datos de entrada se calculan las posiciones de cada una de las facetas que conforman el arreglo y se realiza el canteo del arreglo para el día indicado. Para iniciar la simulación se calcula la hora de amanecer y anochecer, la simulación inicia una hora después del amanecer y termina una hora antes del anochecer. Se pueden realizar simulaciones diarias o anuales. A partir de la fecha y hora indicada se calcula el vector solar  $\hat{s}$  y los parámetros  $\hat{n}$ ,  $\hat{t}$ ,  $\xi$ ,  $\psi$ , y  $\varepsilon$  para la faceta central, que son útiles para calcular los mismos parámetros de las facetas laterales, como se vio previamente en la Sección 4. Como resultado se imprimen un `archivo.dat`, el programa genera una carpeta `dat` en donde guarda los resultados con el nombre de `R{file}F{#facetas}T{día}-{mes}.dat`, si la simulación se realiza para un día en específico. Si la simulación es anual entonces se genera una carpeta `dat/rec{file}F{#facetas}` en donde guarda todos los archivos generados con el nombre de `R{día}-{mes}.dat`. El Cuadro 5.2 muestra parte de un archivo de salida generado por el programa, en la primera columna se muestra el número de faceta del arreglo (`faceta`), en la segunda columna la hora de la simulación (`hora:min`), de la columna 3–5 se indica el vector solar utilizado para la simulación (`s0`, `s1`, `s0`), de la columna 6–8 se indica el vector normal de la faceta (`n0`, `n1`, `n2`), en la columna 9 y 10 se indican las rotaciones (`xi`) y (`psi`), de la columna 11–13 se indica las coordenadas del impacto en el receptor (`x`, `y`, `z`), por último en la columna 14 se indica el error angular (`varepsilon`), el siguiente bloque contiene la misma información pero a una hora distinta.

Cuadro 5.2: Sección de archivo de salida, para un arreglo H4, el 21 de junio, colocado a una distancia  $0.75h$ .

---

```

1 // faceta, hora:min, s0, s1, s2, n0, n1, n2, xi, psi, x, y, z, varepsilon
2 0 8:30 0.728 0.077 0.681 0.420 -0.302 0.856 19.445 24.857 -0.000 0.000 0.000 0.000
3 1 8:30 0.728 0.077 0.681 0.414 -0.303 0.858 19.445 24.463 -0.026 0.000 -0.002 0.600
4 2 8:30 0.728 0.077 0.681 0.427 -0.301 0.853 19.445 25.251 0.026 0.000 -0.002 0.600
5 3 8:30 0.728 0.077 0.681 0.402 -0.305 0.864 19.445 23.676 -0.079 0.000 -0.018 1.807
6 4 8:30 0.728 0.077 0.681 0.439 -0.299 0.847 19.445 26.038 0.079 -0.000 -0.018 1.807
7
8
9 0 9:30 0.558 -0.005 0.830 0.306 -0.331 0.893 20.374 17.811 -0.000 0.000 0.000 0.000
10 1 9:30 0.558 -0.005 0.830 0.299 -0.332 0.894 20.374 17.418 -0.023 0.000 -0.002 0.534
11 2 9:30 0.558 -0.005 0.830 0.312 -0.331 0.891 20.374 18.205 0.023 0.000 -0.002 0.534
12 3 9:30 0.558 -0.005 0.830 0.286 -0.334 0.898 20.374 16.631 -0.071 0.000 -0.017 1.612
13 4 9:30 0.558 -0.005 0.830 0.325 -0.329 0.886 20.374 18.992 0.071 0.000 -0.017 1.612

```

---

## 5.2. SolTrace

Como se mencionó previamente, **SolTrace** tiene la ventaja de utilizar **scripts** en lenguaje LK. El lenguaje LK fue diseñado para ser rápido y fácilmente embebido en otras aplicaciones, así como proveer al usuario una funcionalidad extendida de la aplicación.

Para realizar las simulaciones de trazado de rayos de los arreglos de helióstatos con el seguimiento *elevación-Fresnel*, se escribió un **script** que pudiera leer archivos de datos, ver Sec. C.2. El archivo de datos contiene la información necesaria para que **SolTrace** pueda construir el arreglo y orientarlo según las normales que se le proporcionan, y posteriormente realiza la simulación para el vector solar indicado. El archivo de datos que alimenta

---



al **script** LK tiene el arreglo mostrado en el Cuadro 5.3, en donde se le indica el número de facetas, el vector solar de la simulación, la posición de las facetas y sus normales. Esta información se obtiene de los archivos obtenidos del programa **erun** como se puede observar en el Cuadro 5.2.

El archivo del Cuadro 5.3 es para un arreglo que contiene cuatro facetas, y para un vector solar determinado, es decir, si deseamos realizar la simulación de dicho arreglo para cada hora se requieren generar al rededor de 10 archivos, hay que recordar que se calcula la hora de amanecer y a partir de ahí se selecciona el intervalo de simulación. Para la simulación de un arreglo durante un año, simulando únicamente para el 21 de cada mes, se requiere construir 120 archivos aproximadamente. Dado que se tienen seis distintos arreglos, a tres distintas posiciones, en donde además se analiza el comportamiento a distintas latitudes o con diferentes receptores, el número de archivos que se requieren son más de 2,160. Por lo que fue necesario recurrir a la programación del **Shell**<sup>2</sup>, el cual permite generar los archivos que requiere el **script** LK a partir de los datos de salida del programa **erun**.

Cuadro 5.3: Archivo de entrada de **SolTrace** para un arreglo de cuatro facetas, para las 8:30 tiempo solar.

---

```

1 // N. facetas, vector solar, posición facetas, normales
2 4
3 0.728000 0.077000 0.681000
4 0.6000 26.25 0.0
5 -0.600 26.25 0.0
6 1.8000 26.25 0.0
7 -1.800 26.25 0.0
8 0.414000 -0.303000 0.858000
9 0.427000 -0.301000 0.853000
10 0.402000 -0.305000 0.864000
11 0.439000 -0.299000 0.847000

```

---

El **script** LK que se escribió para realizar las simulaciones, ver Apéndice C.2, lee los 12 archivos indicados en la variable **files**, desde las 6:30 hasta las 17:30. Cada archivo contiene el vector solar correspondiente, la posición de las facetas y las normales para la hora indicada. El cálculo de la irradiancia se realiza dentro del **script** de acuerdo a la ecuación 4.15, la variable de orientación del **aimpoint**<sup>3</sup> es calculada internamente, a partir de lo cual se construye el modelo. Para las simulaciones se utilizaron 5,000,000 de rayos y una malla de 50×50 en el receptor. Una vez simulado el modelo, los datos de simulación son exportados a dos archivos, uno contiene la matriz de irradiancia en el receptor en W/m<sup>2</sup> y el otro contendrá parámetros estadísticos de interés: también se guarda el modelo generado en **file.stinput** de **SolTrace**, para cualquier aclaración posterior.

Para ejecutar el programa, el **script** LK se puede abrir con el programa de **SolTrace** y en la sección **Scripting** ejecutarlo. Sin embargo para una gran cantidad de simulaciones, como es este el caso, resulta poco práctico. Otra opción es ejecutar el **script**, en **Linux**, desde la terminal mediante una instrucción desde la línea de comandos:

```
> SolTrace -s file.lk
```

---

<sup>2</sup>Es el intérprete de comandos de Linux.

<sup>3</sup>Para conocer más detalles sobre el funcionamiento de **SolTrace** ver [github.com/imorenoc](https://github.com/imorenoc).

---

Si se puede ejecutar desde la línea de comandos, entonces puede utilizarse un `shell script` para gestionar todos los archivos `file.lk` necesarios para la simulación. Lo que nos permite, mediante una instrucción, ejecutar todas las simulaciones necesarias.



---

## Capítulo 6

# Resultados

---

Con la metodología antes mencionada, Sec. 4.2, los arreglos de 2, 4, 8, 16 y 32 facetas (identificados como H2, H4, H8, H16 y H32, respectivamente) fueron simulados. También un arreglo de una faceta (H1) fue incluida en algunos casos de comparación. Hay que notar que esta última tiene un seguimiento perfecto, y es equivalente a un helióstato ordinario, para el actual propósito. Cada faceta tiene las dimensiones de  $1 \times 1$  m, y se consideró una separación de 20 cm entre facetas. La mayoría de las simulaciones asumen un día de canteo para el día 21 de marzo, si no se especifica otra cosa. Sin embargo, el estudio del canteo para cada 21 de cada mes se estudia para el arreglo H16.

La localización considerada para la presente simulación fue Hermosillo, México, a una latitud de  $29.028452^\circ +N$  y longitud de  $111.145475^\circ +O$ , a una altitud de 200 m. Algunos casos fueron simulados a latitudes de 2, 6, 10, 30 y  $45^\circ +N$ , para estudiar el efecto de este parámetro. Los valores de la simulación son reportados en tiempo solar, iniciando al amanecer ( $-\omega_a$ ) más  $15^\circ$  y terminando al anochecer ( $\omega_a$ ) menos  $15^\circ$ .

Se simuló un receptor colocado a una altura de  $h = 35$  m, con el arreglo de helióstatos localizado a las distancias de  $0.75h$  (26.25 m, caso A),  $2h$  (70 m, caso B) y  $4h$  (140 m, caso C), al norte de la torre (eje  $y$ ), como lo muestra la Figura 6.1.

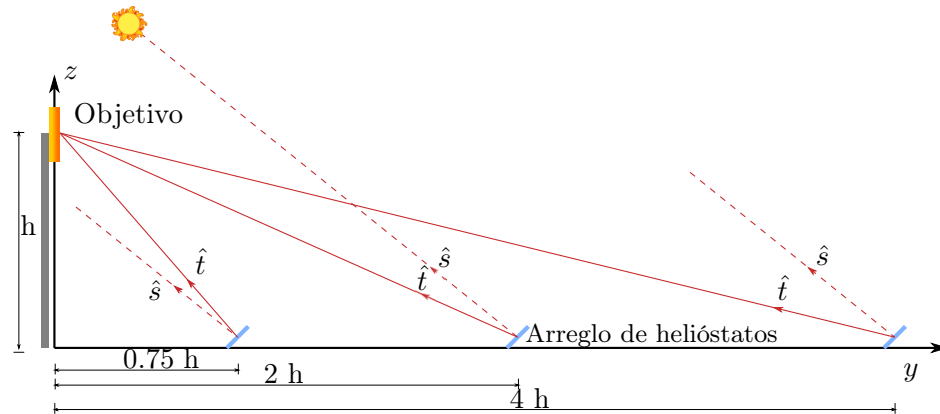


Figura 6.1: Diferentes ubicaciones de los arreglos de helióstatos analizados.

Como primer ejemplo para estudiar las principales características de la operación de esta clase de seguimiento, se seleccionó el arreglo H4. La Figura 6.2 muestra el impacto de los

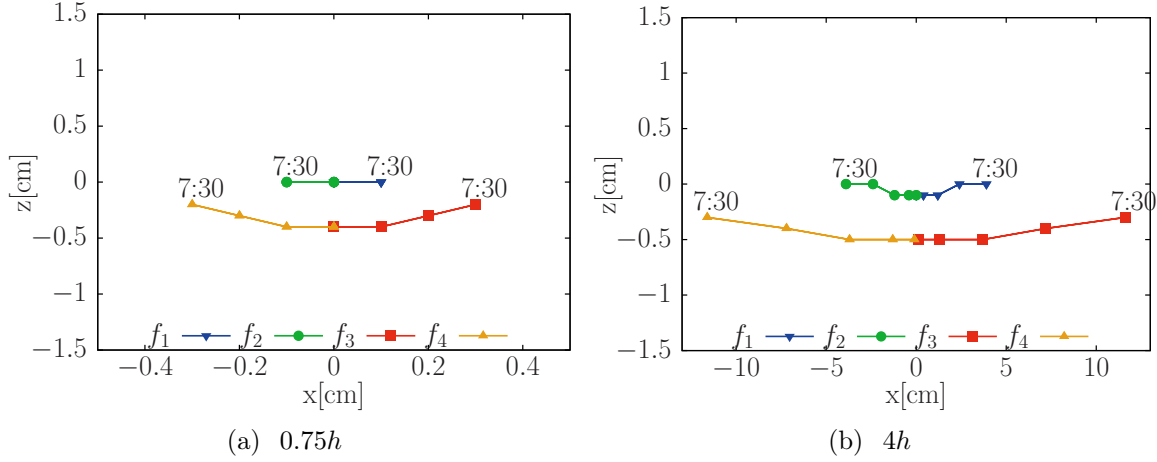


Figura 6.2: Punto de impacto del rayo reflejado de cada una de las facetas en el receptor, para las cuatro facetas  $f_1$ ,  $f_2$ ,  $f_3$  y  $f_4$  del arreglo H4. Simulación para el 21 de marzo. Para las posiciones de  $0.75h$  y  $4h$ , de izquierda a derecha.

puntos del rayo central de cada una de las facetas en el objetivo (plano  $xz$ ), para cada hora desde las 7:30 h hasta las 16:30 h, el 21 de marzo. Debido a la simetría de la trayectoria aparente del Sol y el mecanismo de seguimiento, las trayectorias de los puntos de impacto, después del mediodía solar recorren la misma trayectoria pero en sentido contrario para terminar en el mismo punto de inicio. La Figura 6.2(a)<sup>1</sup> y 6.2(b) muestran el comportamiento de un arreglo de helióstatos localizado a  $0.75h$  y  $4h$ , respectivamente. Note que la escala del eje  $z$  se mantiene en ambas figuras, mientras que la escala del eje  $x$  es mucho más grande en el segundo caso. Puede notarse que el número de puntos que se observan representados en la gráfica son menores a los números de intervalos de tiempo considerados, los cuales deberían ser cinco, uno por cada hora de las 7:30 a las 11:30, ya que la hora correspondiente en la tarde cae en los mismos puntos que en la mañana, como se explicó anteriormente. La razón por la que se observan menos de cinco puntos es porque se encuentran extremadamente cercas entre sí y no es fácil distinguirlos. Por ejemplo, en la Figura 6.2(a), nosotros podemos distinguir únicamente los puntos correspondientes a las 7:30 (que es el mismo de las 16:30) para las facetas 1 y 2, y el resto de los puntos se acumulan en el centro. Para las facetas 3 y 4 podemos observar claramente los primeros tres puntos, y los demás se traslapan. La situación es diferente para la Figura 6.2(b), donde todos los puntos pueden distinguirse fácilmente para todas las facetas.

El comportamiento mostrado anteriormente es sólo para el día de canteo. La Figura 6.3 muestra los puntos de impacto en el objetivo para los días 21 de cada mes, desde junio a diciembre. Note que el comportamiento anual es simétrico respecto a junio. Los intervalos de evaluación varían dependiendo de la hora de amanecer, iniciando una hora después y

<sup>1</sup>Hay que hacer notar que por la cercanía del helióstato H4, a la distancia de  $0.75h$ , los puntos de impacto son muy cercanos al origen, e.g., las facetas  $f_1$  y  $f_2$  prácticamente se comportan con seguimiento perfecto exceptor a las 7:30 y 8:30 por la mañana y a las 15:30 y 16:30 por la tarde.

terminando una hora antes del anochecer, y corresponde a los intervalos de  $[6:30, 17:30]$  en junio,  $[7:30, 16:30]$  en septiembre y  $[8:00, 16:00]$  en diciembre. Los puntos de impacto de las facetas simétricas pares  $f_1, f_2$  y  $f_3, f_4$ , mantienen la simetría, tras el cambio de posición en el receptor a lo largo del año. Puede ser observado que el valor mínimo de dispersión ocurre en el día de canteo, para el 21 de marzo/septiembre, cuando la declinación solar es cero. El valor máximo de dispersión ocurre en junio, de 11.8, 15.1 y 19.6 cm en el eje  $x$  y -2.1, -1.1 y -0.6 cm en el eje  $z$ , para las posiciones de  $0.75h$ ,  $2h$ , y  $4h$ , respectivamente. Los errores angulares máximos asociados son de 2.678, 1.926 y 1.356 mrad, respectivamente, para los mismos casos.

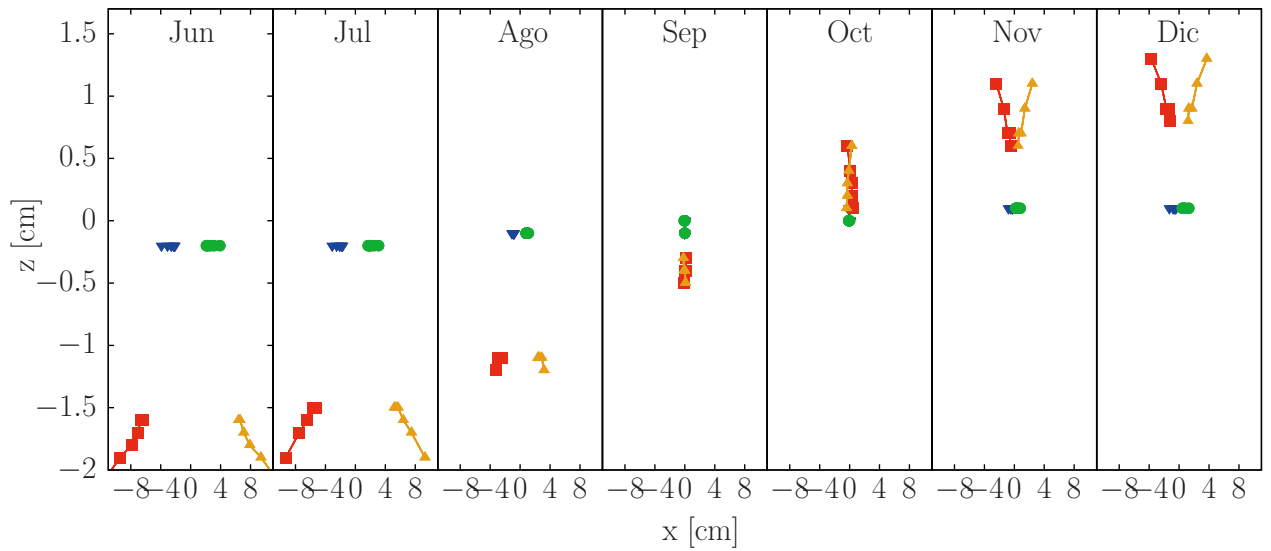


Figura 6.3: Puntos de impactos de los rayos centrales en el objetivo para el arreglo H4 localizado a  $0.75h$ , para el día 21 de cada mes de junio a diciembre.

En las gráficas subsecuentes, para presentar el resultado del comportamiento anual, se calculó la desviación estándar diaria de los puntos de impacto de todas las facetas con respecto al centro del objetivo. Esta es representada como  $\sigma_x$  y  $\sigma_z$  para las direcciones  $x$  y  $z$ , respectivamente. De igual manera para la desviación angular  $\varepsilon$ .

La Figura 6.4 muestra el comportamiento de la desviación estándar diaria a lo largo del año, para el arreglo H4 localizado a  $0.75h$ ,  $2h$  y  $4h$ . Estos tres casos son representados en la figura con los subíndices  $A$ ,  $B$  y  $C$ , respectivamente, como se muestra a continuación. La desviación estándar en el eje  $x$  se incrementa conforme aumenta la distancia del arreglo, contrario a lo que ocurre con la desviación estándar en  $z$ , la cual decrece. En la Figura 6.4(c) se muestra la desviación estándar del error angular  $\sigma_\varepsilon$ , donde se observa que esta cantidad tiende a la uniformidad a lo largo del año conforme aumenta la distancia.

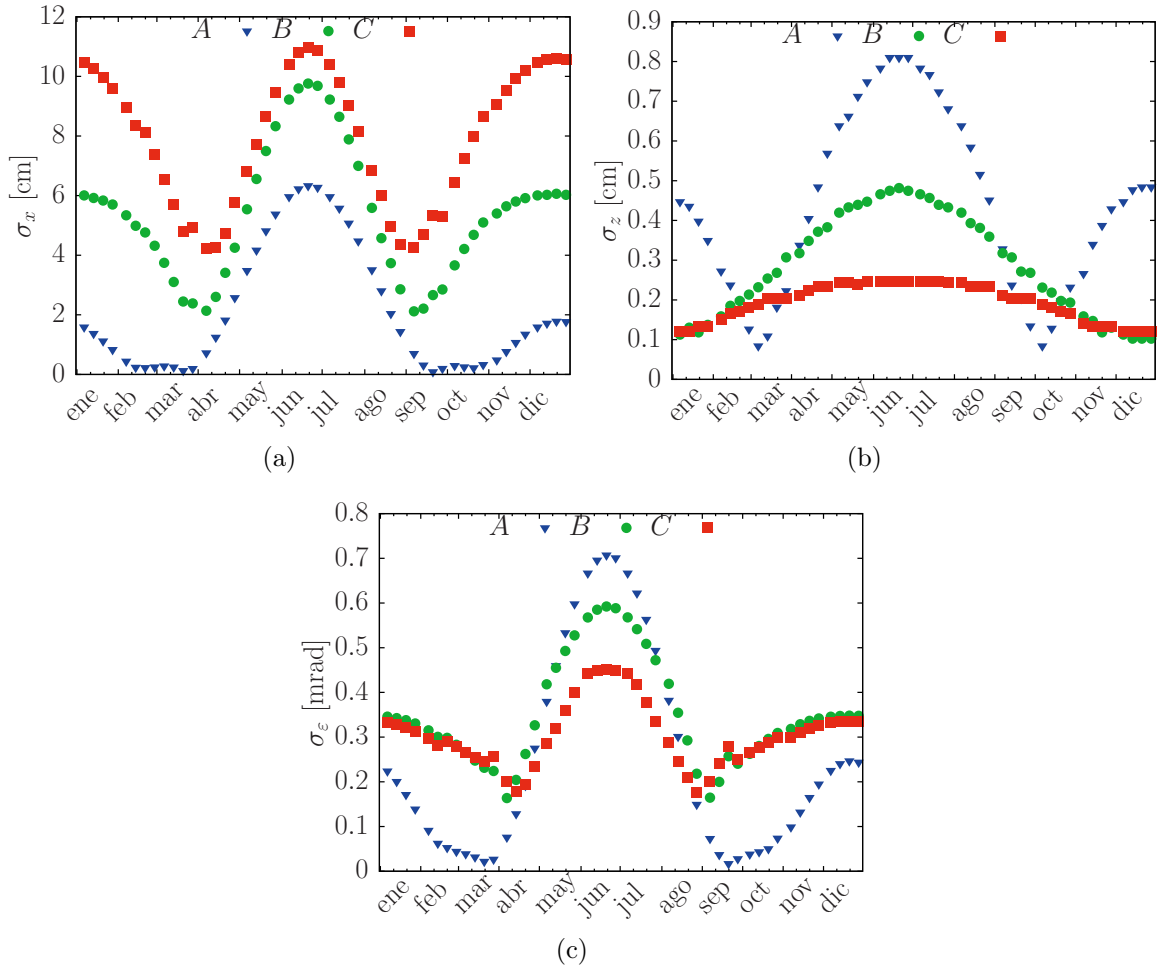


Figura 6.4: Desviación estándar diaria de los puntos de impacto en  $x$  (a) y  $z$  (b), y el error angular (c), para el arreglo H4. Para las posiciones de arreglo ubicado a  $0.75 h$  (A),  $2h$  (B) y  $4h$  (C).

### 6.0.1. Efecto del número de facetas

El número de facetas en el arreglo afecta la distribución de los puntos de impacto en el receptor. En esta sección se analizará el comportamiento de los arreglos H2, H4, H8, H16, H24 y H32, para las posiciones de  $0.75h$  (A),  $2h$  (B) y  $4h$  (C). La Figura 6.5 muestra la desviación estándar diaria para todos los arreglos ubicados a una distancia de  $2h$ . Aparentemente, todas las desviaciones aumentan con el número de facetas. Esta desviación alcanza su máximo valor para el 21 de junio. Recordando que el tamaño lateral del arreglo es proporcional al número de facetas, y el error incrementa prácticamente proporcionalmente a esta dimensión.

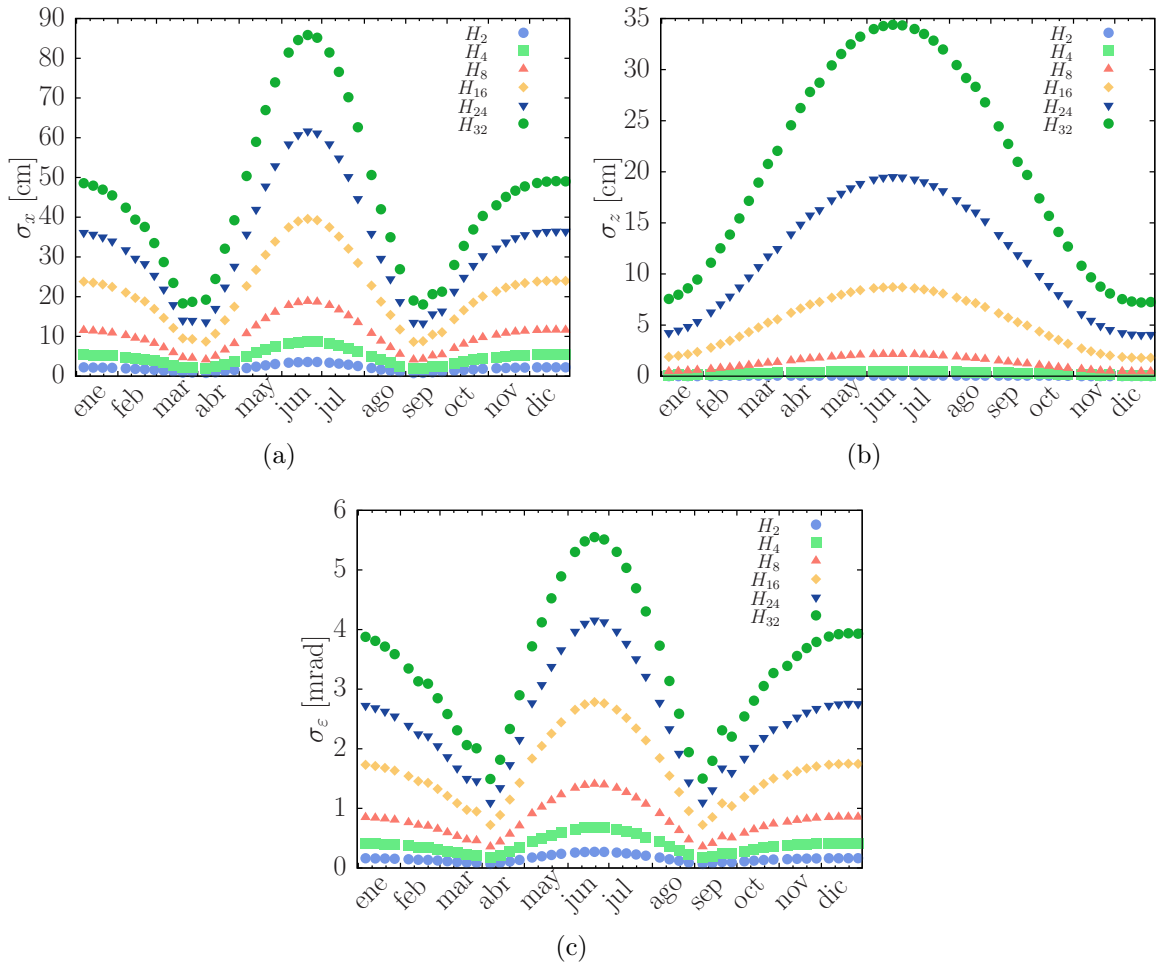


Figura 6.5: Desviación estándar diaria para los arreglos de heliostatos H2, H4, H8, H16, H24, y H32 a lo largo del año, a lo largo de los ejes  $x$  (a) y  $z$  (b), y el error angular (c). Los arreglos están localizados en  $2h$ .

La Figura 6.6 muestra los cuartiles de la desviación estándar diaria a lo largo del año en  $x$  (a),  $z$  (b),  $\epsilon$  (c). En donde se observa los resultados para los arreglos H2, H4, H8, H16, H24 y H32 localizados en las posiciones A, B y C. Se puede observar que la mediana de la



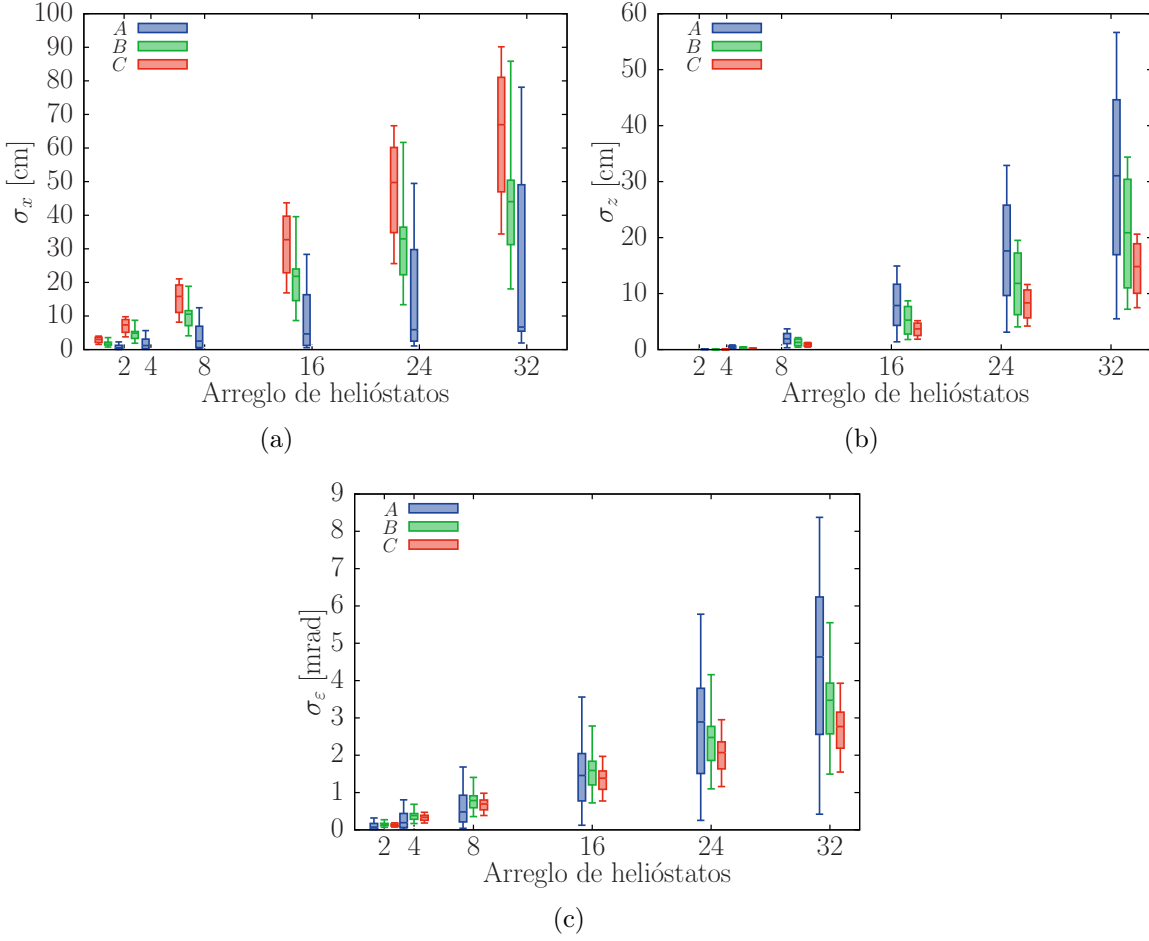


Figura 6.6: Cuartiles de la desviación estándar diaria para los arreglos de helióstatos H2, H4, H8, H16, H24, y H32 en los ejes  $x$  (a) y  $z$  (b), y el error angular (c). Para arreglos colocados en las posiciones de  $A$  ( $0.75h$ ),  $B$  ( $2h$ ) y  $C$  ( $4h$ ).

desviación estándar en  $x$  aumenta cuando el arreglo se mueve lejos de la torre y se observa el efecto contrario para el eje  $z$ , es decir que al incrementar la distancia la desviación estándar en esta dirección disminuye. Para el error angular  $\varepsilon$  también el valor de la mediana de la desviación estándar decrece. En todos los casos, las distribuciones de estos errores son más angostas, a lo largo del año, conforme están más lejos de la torre. Esto es de esperarse ya que los ángulos de proyección desde la torre al helióstato durante el año varían más lentamente con las estaciones para helióstatos distantes.

Para los valores máximos de desviación estándar se ajustó un polinomio para las diferentes localizaciones, como se muestra en la Figura 6.7. En todos los casos existe una desviación grande cuando el número de facetas del arreglo se incrementa. En el caso del eje  $x$ , el crecimiento de la desviación es bien representado por un polinomio de segundo orden para una distancia relativamente corta ( $0.75h$ ), y mediante una función lineal a distancias grandes ( $2h$  y  $4h$ ). La desviación en el eje  $z$  tiene un crecimiento siempre de un polinomio de segundo

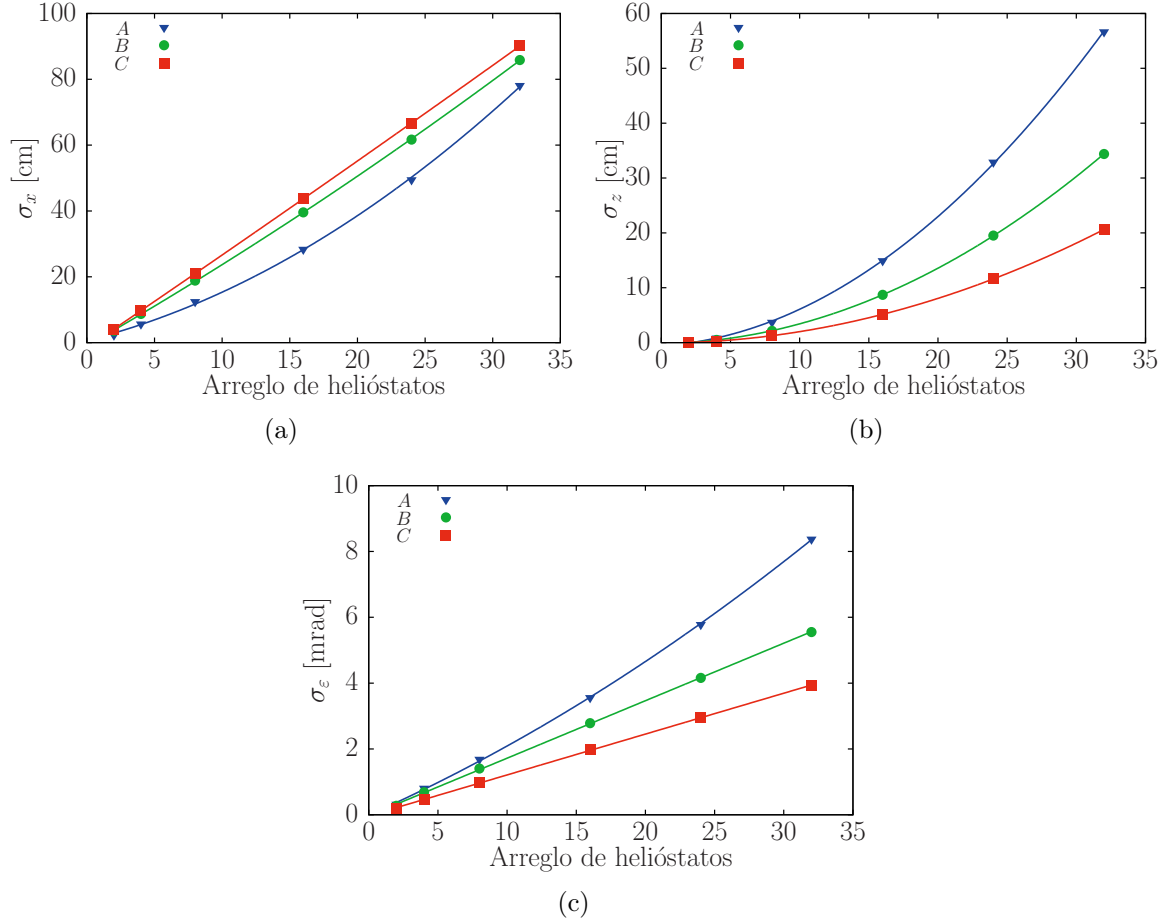


Figura 6.7: Comportamiento de los valores de desviación máxima anual como una función del tamaño del arreglo (puntos), para las posiciones de  $A$ ,  $B$  y  $C$  con sus correspondientes ajustes polinomiales (líneas).

orden.

Para los errores angulares  $\sigma_\epsilon$  se puede observar una tendencia parabólica a distancias cortas y, conforme la distancia incrementa, la tendencia llega a ser lineal con una pendiente decreciente. En todos los casos el error angular incrementa al incrementar el número de facetas en el arreglo.

### 6.0.2. Efecto de la orientación del receptor

El análisis previo fue realizado considerando un receptor vertical. En esta sección se considera el comportamiento para un receptor horizontal. Es evidente que el error angular  $\sigma_\epsilon$  del arreglo de helióstatos, para el mismo día de canteo, es el mismo cuando el receptor cambia debido a que es una característica determinada únicamente por el campo de helióstatos. Sin embargo, la desviación estándar diaria a lo largo de los ejes del receptor se comportan

diferente. La siguiente simulación se llevó acabo considerando los mismos parámetros previos, como el día de canteo, los diferentes arreglos de helióstatos y la localización del receptor.

La Figura 6.8 compara los resultados de la desviación estándar diaria para los receptores horizontales y verticales, para arreglos de helióstatos localizados a distancias diferentes. La Figura 6.8(a) corresponde a la dirección  $x$  (eje este-oeste), que es el eje que comparten ambos receptores. Se observa que los arreglos con 2 a 10 facetas tiene una desviación aproximadamente igual, para todas las distancias. Por otro lado, para arreglos grandes el receptor horizontal tiene una desviación menor. Para la posición de  $0.75h$  y  $2h$  esta diferencia es muy perceptible; no así para distancias mayores a  $4h$ , donde la desviación es prácticamente la misma en ambos casos para todos los tamaños de helióstatos. En ambos tipos de receptor el valor máximo de desviación estándar en  $x$  incrementa con la distancia.

La Figura 6.8(b) compara los resultados del eje  $y$  (norte-sur) para un receptor horizontal, con el eje  $z$  para un receptor vertical. Para un receptor horizontal a una distancia de  $0.75h$  la desviación es más pequeña que para un receptor vertical, aunque después para las distancias de  $2h$  y  $4h$  la desviación es mayor. Para un receptor vertical la desviación máxima en el arreglo a distancias cortas es mayor y decrece con la distancia, mientras para el horizontal es más pequeño y muestra la tendencia opuesta.

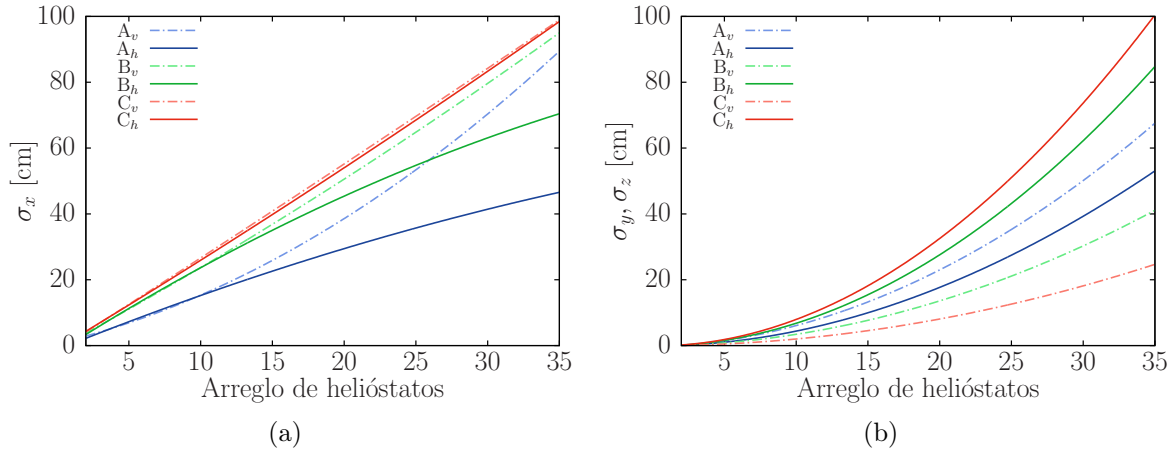


Figura 6.8: La tendencia polinomial de los valores de la desviación estándar máxima para los arreglos colocados a las distancias de  $A$ ,  $B$  y  $C$  para el receptor vertical (línea punteada) y horizontal (línea continua). Los resultados para el eje  $x$  (a) son presentados para ambos receptores, mientras que los ejes  $y$  y  $z$  (b) son presentados para el receptor horizontal y vertical, respectivamente.

La Figura 6.9 muestra el comportamiento de la desviación máxima en  $\sigma_x$  y  $\sigma_z$  en un receptor inclinado  $\Pi$ . El plano del receptor  $\Pi$  es un plano intermedio entre la posición vertical y horizontal que es definido por el vector de posición entre el receptor y el centro del arreglo de helióstatos. Para arreglos demasiado cercanos el plano debería de comportarse como un plano horizontal y para un arreglo muy lejano como un plano vertical. Para nuestro caso de estudio tenemos tres distintos planos  $\Pi$  para las distancias  $A$ ,  $B$  y  $C$ , respectivamente.

La Figura 6.9 muestra el comportamiento del receptor  $\Pi$  a las distancias de  $A$ ,  $B$  y  $C$  así como el intervalo de desviación entre el receptor horizontal y vertical. En la Figura 6.9(a) se muestra la desviación  $\sigma_x$  para el plano  $\Pi$  que tiene valores entre el intervalo de ambos planos. Los valores superiores corresponden al plano vertical y los inferiores al plano horizontal. En el eje  $x$  podemos observar que el plano  $\Pi$  tiende a un valor mínimo en el plano horizontal a distancias cortas de  $0.75h$  y para distancias largas de  $4h$  el comportamiento es igual al de un receptor vertical.

La Figura 6.9(b) muestra el comportamiento del plano  $\Pi$  en el eje  $z$ , o  $y$  para un receptor vertical u horizontal, en donde se puede observar el intervalo entre el receptor vertical y horizontal, en este caso el receptor horizontal es el límite superior. Como se ha mencionado anteriormente la desviación estándar en este eje incrementa en el caso del receptor horizontal. Como en el caso previo para una distancia relativamente lejos de  $4h$  el plano  $\Pi$  ya se comporta como un receptor vertical. Para la distancia de  $0.75h$  hay una reducción en la desviación en este eje.

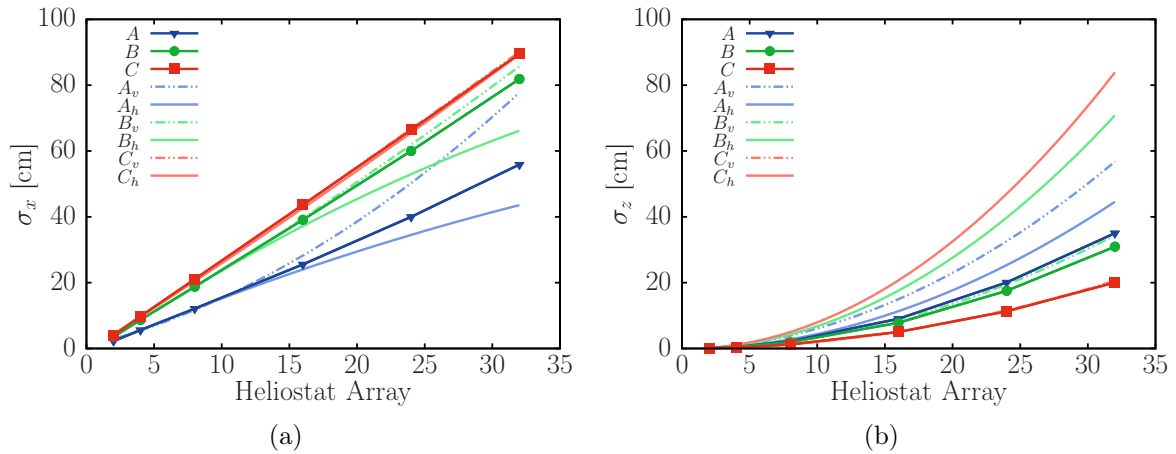


Figura 6.9: Tendencias de la desviación estándar en un receptor plano  $\Pi$  para un arreglo de helióstatos colocado en las posiciones de  $A$ ,  $B$  y  $C$  y sus intervalos limitados por un receptor vertical y horizontal.

La Figura 6.9 nos indica que los valores límites menores de desviación estándar son para un plano horizontal en el eje  $x$  y para un plano  $\Pi$  en el eje  $z$ . Idealmente la combinación de ambos planos debería darnos los valores mínimos de desviación.

### 6.0.3. Efecto de Latitud

La simulación previa se llevó a cabo para una latitud de  $29^\circ + N$ . Sin embargo, debido a la configuración de los arreglos de helióstatos y el tipo de seguimiento se espera que al cambiar la latitud el comportamiento sea diferente. Para verificarlo, se realizaron simulaciones para las latitudes de  $2^\circ$ ,  $6^\circ$ ,  $10^\circ$ ,  $30^\circ$  y  $45^\circ + N$ , para los arreglos H2, H4, H16, H24 y H32, localizados a  $2h$ . Se considero un receptor vertical, y los resultados son presentados en la Figura 6.10.

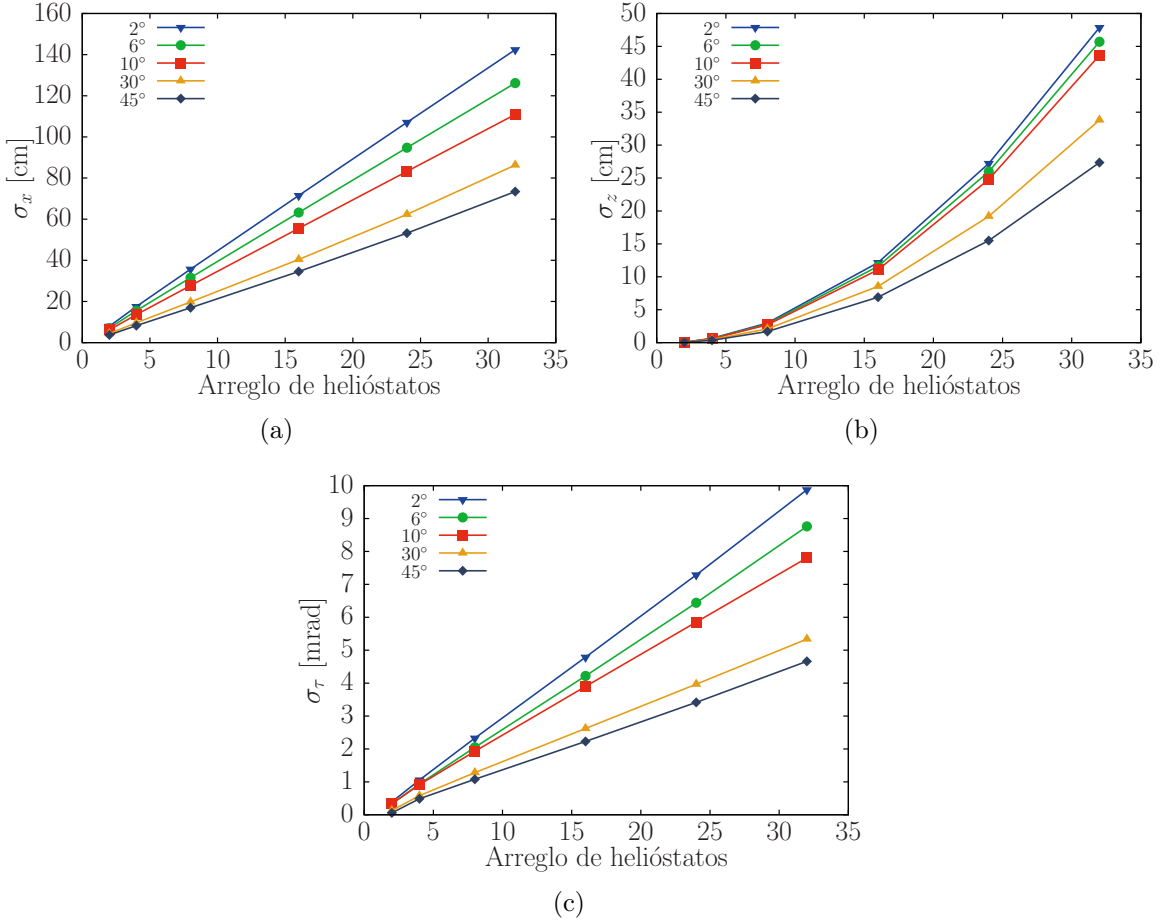


Figura 6.10: Comportamiento del valor máximo de la desviación estándar diaria de cada arreglo de helióstato a las latitudes de 2, 6, 10, 30, y 45°+N, para los ejes  $x$  (a),  $z$  (b) y para el error angular  $\varepsilon$ . Los arreglos son colocados en una posición de  $2h$ .

Se encontró que la desviación estándar máxima en ambos ejes disminuye con la latitud, para todos los tamaños de los arreglos. Este es un comportamiento que se mantiene para todas las posiciones, aunque solo el caso de  $2h$  es presentado en la Figura 6.10. La razón son los pequeños ángulos en promedio entre el sistema Sol-Helióstat-Torre para latitudes grandes.

La diferencia máxima en el eje  $x$  para los arreglos de 2 y 4 facetas, entre latitudes de 2° y 45°, es de 4.19 y 9.27 cm, respectivamente. Sin embargo, este valor incrementa para arreglos mayores alcanzando, para el caso de H32, una diferencia de 68.97 cm. La desviación en la dirección  $z$  para un arreglo de 2 facetas es prácticamente cero, y va aumentando conforme aumenta el número de facetas, para el caso de H32 la diferencia es de 20.5 cm.

La desviación estándar máxima del error angular también disminuye cuando se incrementa la latitud, comportamiento que se mantiene para los diferentes arreglos. La diferencia de  $\varepsilon$  entre la latitud 2 y 45°, para los arreglos de H2 y H32, son de 0.33 y 5.21 mrad, respecti-

vamente, para arreglos colocados a la distancia de  $2h$ .

Para un receptor horizontal el comportamiento es similar, conforme incrementa la latitud la desviación del arreglo decrece. Para visualizar cual es la diferencia de la desviación estándar entre un receptor vertical y uno horizontal se obtiene  $\Delta\sigma = \sigma_{xv} - \sigma_{xh}$ . De esta forma, si el valor es positivo, indica que para la latitud dada la desviación estándar del receptor vertical es mayor, si por el contrario el valor es negativo la desviación estándar en el receptor horizontal es mayor. La Figura 6.11 muestra el comportamiento de  $\Delta\sigma$  entre un receptor vertical y uno horizontal para los arreglos H2, H4, H8, H16, H24 y H32 a las latitudes de 2, 6, 10, 30 y 45° localizadas a  $0.74h$ ,  $2h$  y  $4h$ .

Se observa, en la Figura 6.11, que para la distancia de  $0.75h$  la desviación estándar diaria en un receptor horizontal es menor que para uno vertical y la diferencia aumenta cuando incrementa el tamaño del arreglo esta diferencia se reduce al aumentar la latitud. Por otro lado, al aumentar la distancia, cuando los arreglos están posicionados en  $2h$  y  $4h$  el comportamiento de el receptor vertical mejora en el eje  $x$ , pero en la relación  $\Delta_{z-y}$  muestra que el receptor vertical tiene un valor de desviación estándar mucho menor que el receptor horizontal.

#### 6.0.4. Efecto del día de canteo

Otro parámetro que modifica la desviación estándar es el día de canteo que se elige para el arreglo de helióstatos. Todas las simulaciones previas fueron realizadas con un canteo del 21 de marzo, que fue elegido porque tiene una declinación  $\delta = 0^\circ$ , que representa una trayectoria solar promedio para el año. Una modificación del día de canteo debería cambiar el comportamiento de la curva de desviación estándar diaria. Para analizar este efecto, varias simulaciones fueron realizadas mientras el día de canteo se modificaba para el 21 de cada mes. Para la presente simulación se eligió el arreglo H16. La Figura 6.12 muestra los cuartiles anuales de la desviación estándar diaria para la simulación en las posiciones de  $0.75h$ ,  $2h$  y  $4h$ .

Se observa un comportamiento simétrico respecto al 21 de junio ( $\delta = 23.45^\circ$ ) en ambos ejes. La dispersión mínima se presenta entre marzo y abril. Por lo tanto, es claro que una buena elección es un canteo del 21 de marzo/septiembre ( $\delta = 0$ ) al mediodía solar, para las posiciones de  $2h$  y  $4h$ . Sin embargo, para arreglos colocados relativamente cerca, un canteo para el 21 de abril es adecuado. El efecto del canteo se observa mejor en el eje  $x$ , mientras que en el eje  $z$  no es muy significativo. Nuevamente se observan tendencias opuestas en ambos ejes, mientras la desviación en  $x$  aumentan con la distancia, la desviación en  $z$  disminuye.

#### 6.0.5. Arreglos de Helióstatos Fuera de Eje

Los helióstatos que han sido simulados son arreglos colocados sobre el eje  $y$ , ahora analizaremos el comportamiento de los arreglos fuera de eje que han sido colocados a las mismas distancias de  $0.75h$ ,  $2h$  y  $4h$  en el plano  $xy$ , pero rotados  $45^\circ$  con respecto al eje  $x$ . El receptor se mantiene a una altura de 35 m, y para tener una comparación con respecto a análisis anterior se utiliza un canteo para el 21 de marzo al mediodía solar.

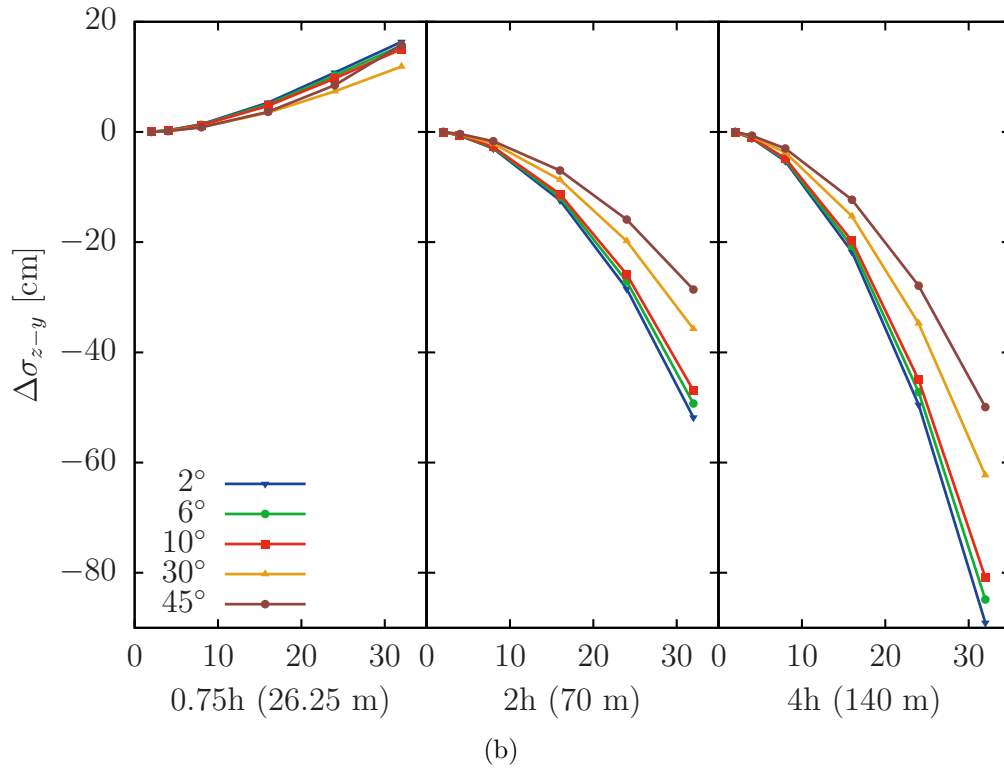
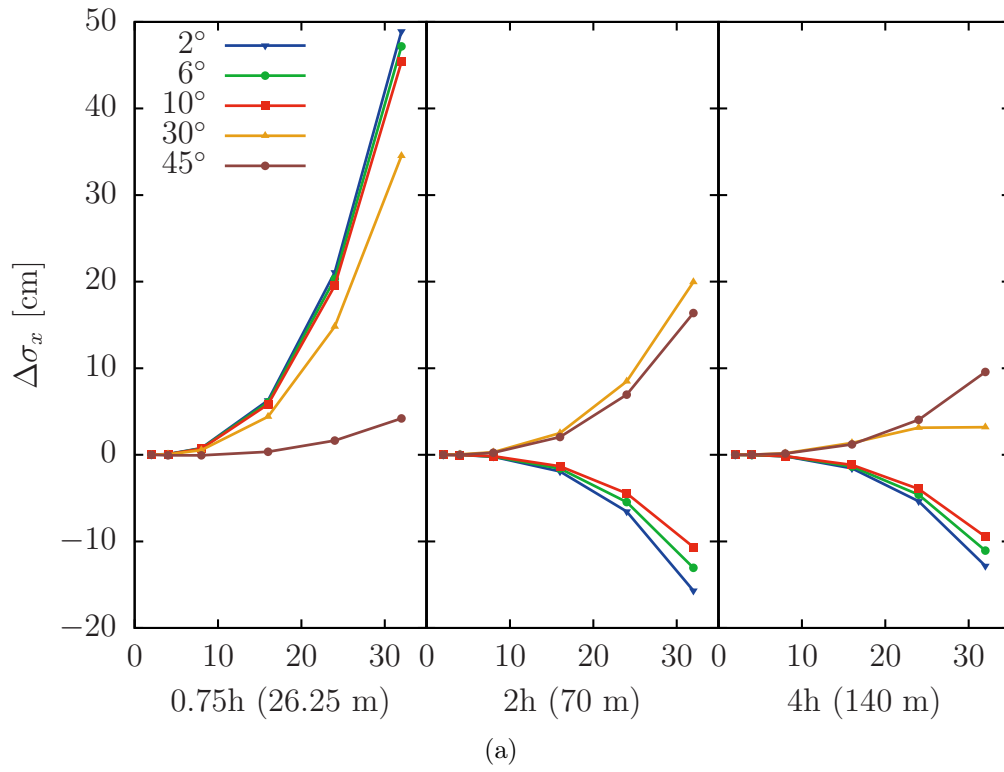


Figura 6.11: Diferencia entre la desviación estándar entre un receptor vertical y un horizontal a diferentes latitudes. (a) Muestra la diferencia de  $\Delta\sigma_x$  (b) la diferencia de  $\Delta\sigma_{z-y}$  para la posición del arreglo a 0.74h, 2h y 4h.

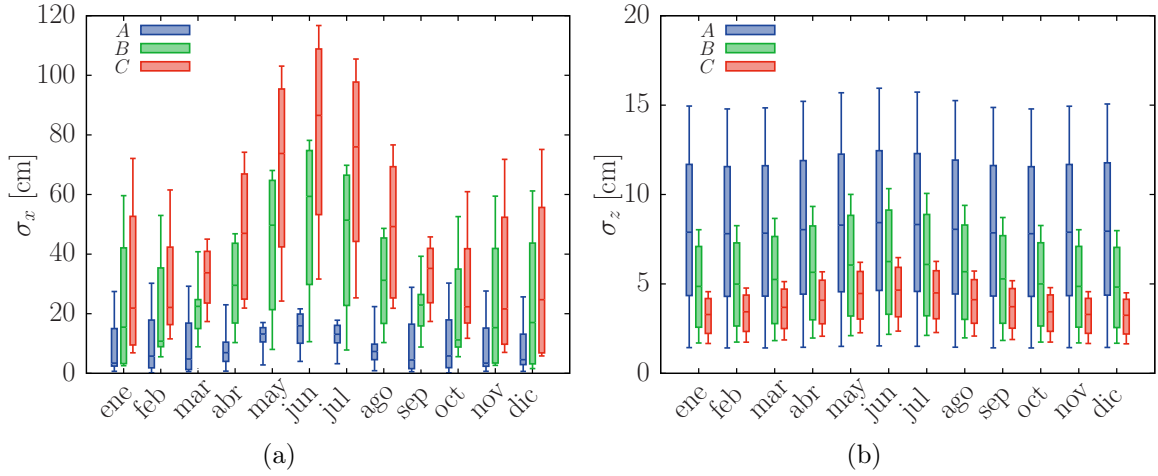


Figura 6.12: Cuartiles de la desviación estándar diaria para el arreglo H16, en el eje  $x$  (a) y  $z$  (b) de un receptor vertical, en función del día de canteo. Para el arreglo a la posición de  $0.75h$  (A),  $2h$  (B) y  $4h$  (C).

La Figura 6.13 muestra el impacto del vector de reflexión del arreglo de helióstatos H4 en un receptor vertical, el día simulado corresponde al día de canteo lo que implica que será el día con menor dispersión en el año. El arreglo fue colocado a una distancia de  $2h$  con lo que se obtiene un intervalo de dispersión de  $[-14.8, 16.2]$  cm y  $[-27.6, 27.2]$  cm en el eje  $x$  y  $y$  respectivamente. La dispersión en el eje  $x$  es de 1.62 veces mayor que el arreglo de helióstatos colocado sobre el eje  $y$ , a la misma distancia. Por otro lado, el valor de dispersión en el eje  $z$  ahora es muy significativo, algo que no ocurren en un arreglo sobre el eje en donde para el caso del arreglo H4 el intervalo de dispersión es de  $[0, -0.7]$  cm, que puede considerarse despreciable, sin embargo para el arreglo H4 fuera de eje el valor de dispersión es mayor que el del eje  $x$ .

Para este arreglo, colocado fuera de eje, se observa que al amanecer el impacto en el receptor es relativamente cerca del objetivo, sin embargo aumenta la dispersión conforme el sol se aproxima al mediodía solar donde tiene la máxima dispersión, después del mediodía solar la dispersión disminuye para terminar en la posición donde inicio.

La Figura 6.14 muestra el comportamiento de la desviación estándar a lo largo del año para los arreglos de helióstatos H2, H4, H8, H16, H24 y H32. En donde se puede observar una gran desviación estándar para los meses centrales, alrededor de junio donde se presenta la desviación máxima. Aunque depende del diseño del receptor, las desviaciones máximas que presentan los arreglos H16, H24 y H32, del orden de metros, resulta poco viable. Los arreglos de helióstatos de hasta ocho facetas pueden resultar atractivas, ya que tienen dispersiones de alrededor de 50 cm, pero a diferencia de los arreglos en eje, las presentas en ambos ejes.

A partir de los valores máximos de la Figura 6.14 podemos obtener la tendencia de crecimiento de la desviación estándar diaria al incrementar el numero de facetas en el arreglo, para las posiciones de  $0.75h$ ,  $2h$  y  $4h$ . En la Figura 6.15 podemos observar que, excepto para la posición de  $0.75h$  para el eje  $x$  que tiene un comportamiento cuadrático, se tiene un



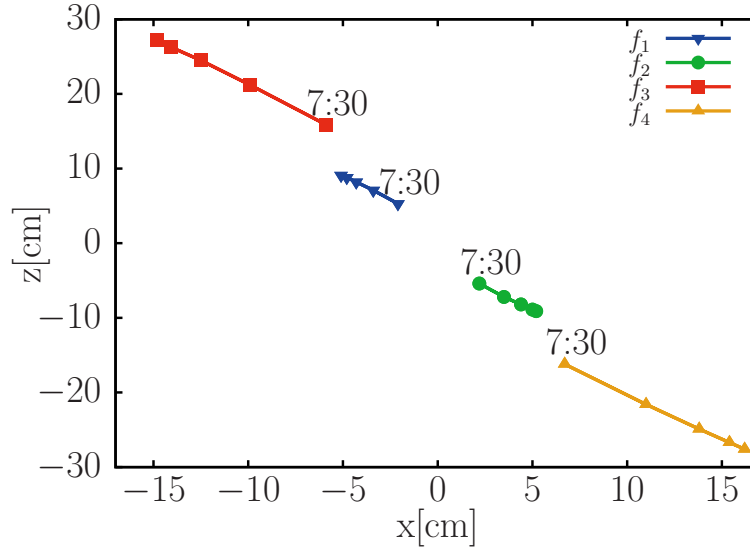


Figura 6.13: Se muestra el comportamiento de los impactos del vector de reflexión central de cada una de las facetas ( $f_1$ ,  $f_2$ ,  $f_3$  y  $f_4$ ) del arreglo H4 en un receptor vertical para el 21 de marzo. Para un arreglo colocado a la distancia  $2h$  cuando el arreglo esta fuera de eje.

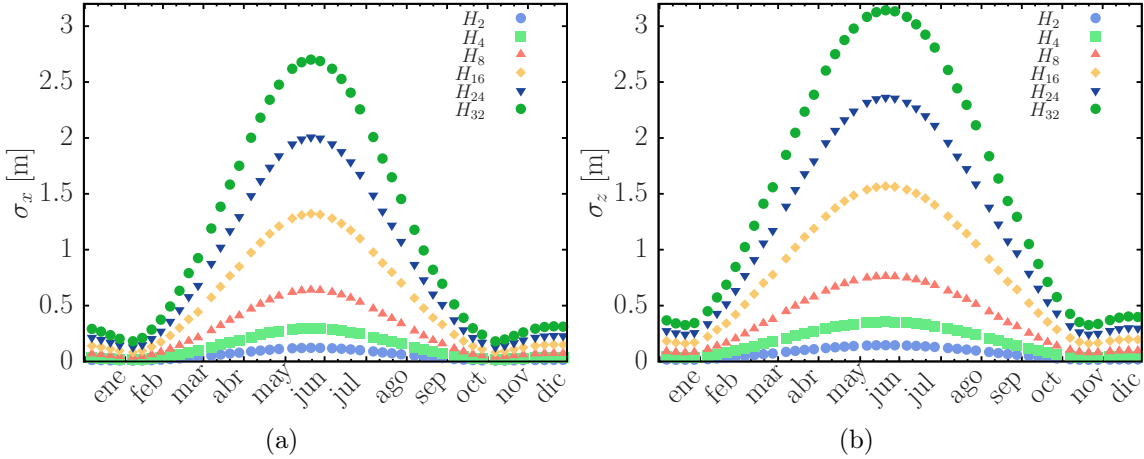


Figura 6.14: Muestra la desviación estándar diaria en un receptor vertical para los arreglos H2, H4, H8, H16, H24 y H32 colocados fuera de eje, en una posición de  $2h$ .

crecimiento lineal, es decir, que la desviación estándar incrementa al aumentar el número de facetas en el arreglo y al aumentar la distancia respecto a la torre.

### 6.0.6. Análisis de distribución de flujo

Para la simulación de trazado de rayos se consideraron los arreglos H1, H2, H4, H8, H16 y H32. Los cuales fueron colocados en las mismas tres posiciones como antes, y la

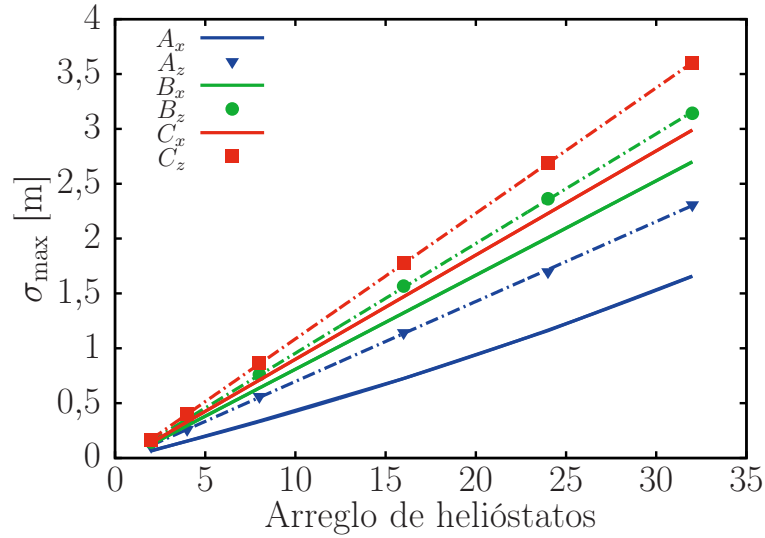


Figura 6.15: Tendencia de crecimiento de la desviación estándar diaria máxima respecto al tipo de arreglo de helióstatos, para las posiciones de 0.75h, 2h y 4h.

potencia radiativa fue concentrada en un receptor vertical con las dimensiones de  $6 \times 6$  m. La Figura 6.16 muestra la distribución de flujo para un arreglo H8, a la posición de 0.75h, para el 21 de junio y diciembre, al mediodía solar. La forma de ambas gráficas se ven muy similares, pero el flujo está más disperso en el segundo caso. El flujo pico calculado cambia de 4.054 a 3.573 kW/m<sup>2</sup> entre junio y diciembre. La potencia evaluada en el receptor fue de 7,400 W para junio y 6,574 W para diciembre, una disminución del 11.16 %.

La gráfica nos muestra la energía contenida en el receptor de  $6 \times 6$  m, donde cerca del 99 % de la energía está contenida. Para una mejor evaluación del tamaño de la mancha solar concentrada (*spot*) calculamos la desviación estándar  $S_x$  y  $S_z$ , a lo largo del eje horizontal. Conociendo estas desviaciones, es posible estimar que cerca del 68 % de la potencia debe estar contenida en una elipse con un área de  $\pi S_x S_z$ .

La Figura 6.17 nos muestra la desviación estándar diaria promedio  $\bar{S}_x$  y  $\bar{S}_z$ , para los arreglos colocados a una distancia de 2h. Se puede observar que con 2, 4, 8, y 16 facetas estas desviaciones diarias permanecen prácticamente constantes a lo largo del año. Sin embargo, para los arreglos de 24 y 32 facetas la desviación en ambos ejes incrementan abruptamente para los meses de mayo, junio, julio y agosto. El radio promedio a lo largo del año para los arreglos H2 es de  $\bar{S}_x = 1.044 \pm 0.067$  y  $\bar{S}_z = 1.022 \pm 0.052$ , mientras que para un arreglo H8 el valor correspondiente es de  $1.034 \pm 0.072$  y  $1.028 \pm 0.058$ , respectivamente. Por lo tanto para arreglos de hasta 16 facetas colocadas a una distancia 2h de la torre, el área que contiene el 68 y 95 % de la energía es de 3.24 y 13.37 m<sup>2</sup>, respectivamente.

El *spot* de H32 es el que posee la desviación estándar más grande, particularmente para el 21 de junio, donde no se tiene un *spot* uniforme, como se puede observar en la Figura 6.17. La Figura 6.18 muestra la evolución del *spot* del arreglo H32 para las 6:30, 8:30 y 12:30 h en un receptor con unas dimensiones de  $6 \times 6$  m. El flujo pico a las 6:30 h es de 2.978 kW/m<sup>2</sup>,

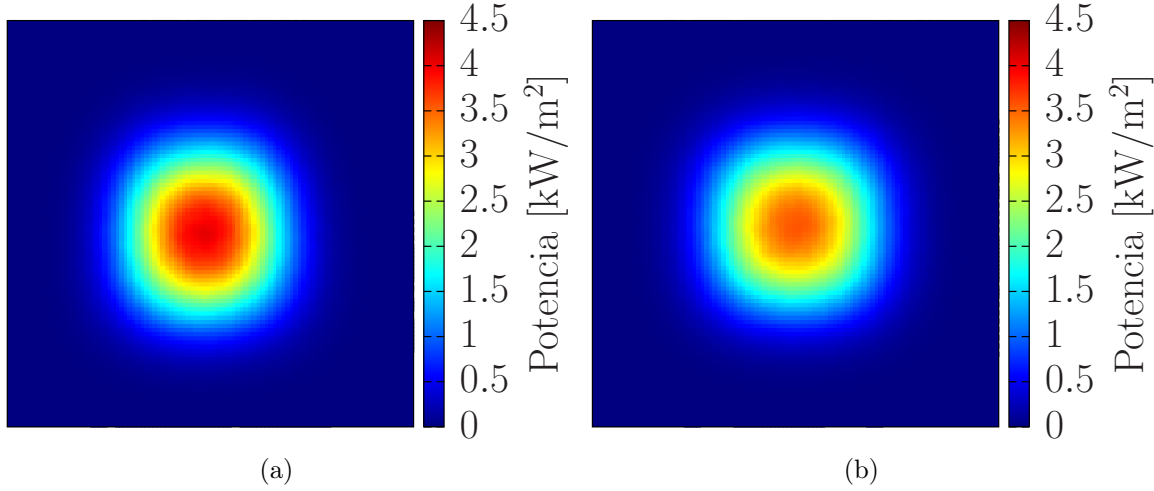


Figura 6.16: La distribución de flujo solar en el receptor vertical ( $6 \times 6$  m) para un arreglo H8 localizado a  $0.75h$ . Al mediodía solar el 21 de junio (a) y diciembre (b).

que incrementa a  $4.579 \text{ kW/m}^2$  para 8:30 h y el valor de  $5.242 \text{ kW/m}^2$  para las 9:30 h se mantiene casi constante hasta el mediodía solar. Esta dispersión presentada en el *spot* en junio limita la concentración, que se demuestra al comparar su comportamiento con el resto de los meses; i.e., el 21 de enero existe un *spot* uniforme al mediodía solar, con un flujo pico de  $9.609 \text{ kW/m}^2$ .

Basado en la distribución de flujo obtenido, la energía diaria fue calculada para el día 21 de cada mes, para todos los arreglos colocados a la misma distancia de  $0.75h$ . La simulación obtenida es mostrada en la Figura 6.19 donde también podemos observar la contribución de un arreglo H1 para tener un punto de referencia. El arreglo H1 corresponde al caso de una simulación de un helióstato individual con un seguimiento cenit-acimut. La gráfica muestra la contribución de energía diaria por faceta en el arreglo; i.e., la energía total suministrada dividida por el número de facetas  $N$  que conforman el arreglo. Como puede ser esperado, debido a las imperfecciones del seguimiento, la energía por faceta decrece al aumentar el número que facetas que conforman el arreglo. Sin embargo, la figura muestra que el valor máximo de decaimiento esta siempre entre H1 y H2, y llega a ser menos significativo conforme el número de facetas que forma el arreglo incrementa. El valor máximo de decaimiento es de 6.01 % al compararlo con H1, mientras el mínimo es de 2.07 %, para los mese de abril y diciembre respectivamente.

El valor de la reducción promedio anual entre el arreglo H1-H2 es de 4.4 %, entre H2-H4 es de 2.3 %, entre H4-H8 es de 1.21 %, entre H8-H16 es de 0.72 %, entre H16-H24 es de 0.39 % y entre H24-H32 es de 0.38 %. La tendencia es que la energía de la faceta para arreglos mayores a H32 tiende a ser la misma, aunque para arreglos de grandes dimensiones el *spot* de la radiación concentrada pierde uniformidad.

Observando nuevamente la Figura 6.5 donde tenemos el valor máximo de desviación estándar en el mes de junio, observamos que el mes con menor energía es diciembre, lo que

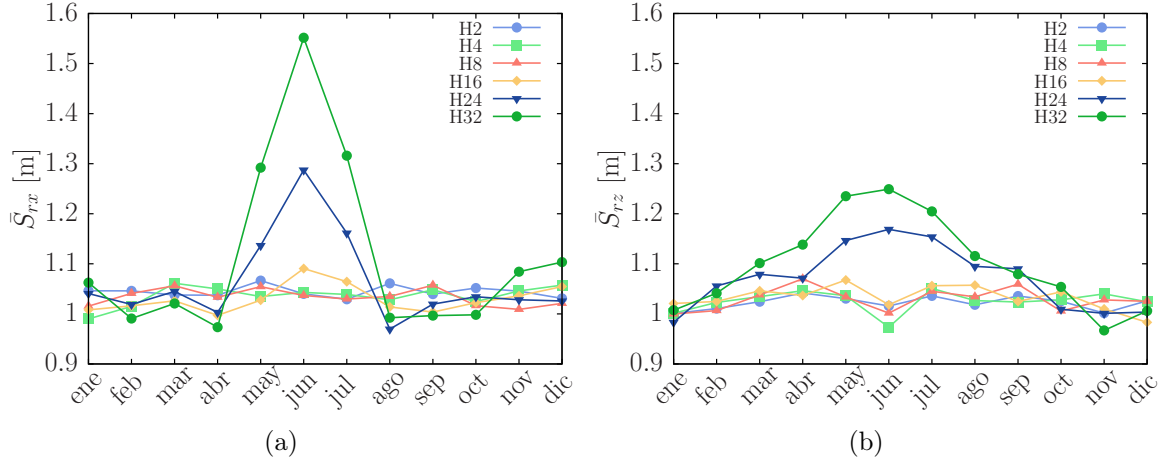


Figura 6.17: El comportamiento diario de la desviación estándar  $S_r$  del *spot*, se muestran en el eje  $x$  y  $z$  a lo largo del año, para diferentes arreglos colocados a una distancia de  $2h$ .

se explica con la compensación del tiempo solar. También se observa que no existe mucha variación entre el valor máximo de junio y septiembre, ya que septiembre tienen la menor dispersión del arreglo y debido a que el canteo fue realizado para el 21 de marzo.

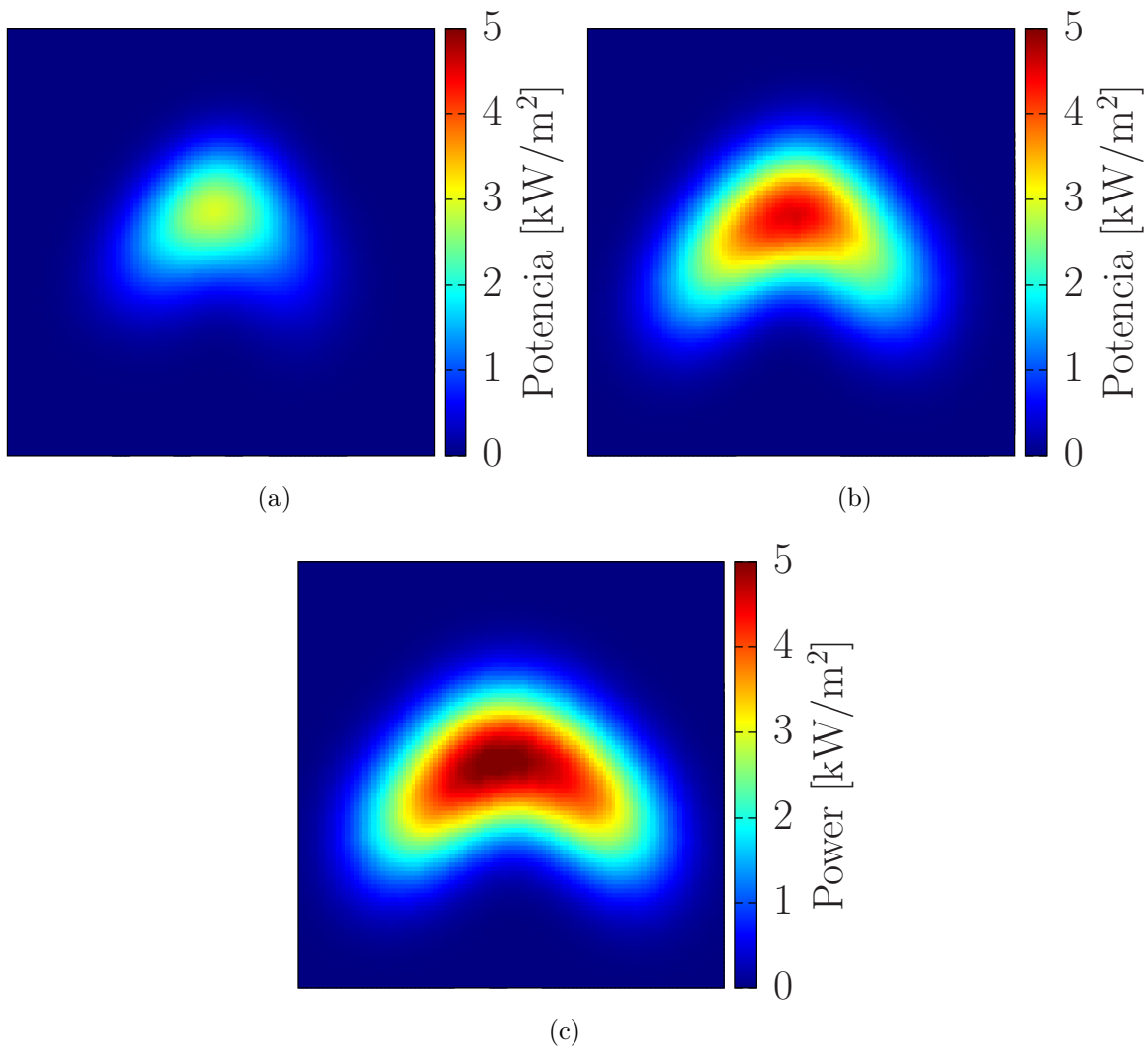


Figura 6.18: Comportamiento del arreglo H32 para el 21 de junio a las 6:30 (a), 8:30 (b) y 12:30 (c). Para un arreglo ubicado a una distancia de  $2h$ .

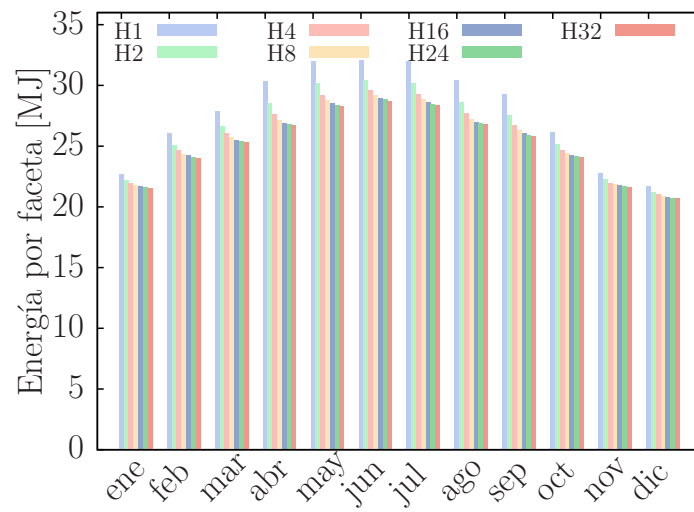


Figura 6.19: Contribución de energía por una faceta del arreglo correspondiente a lo largo del año. Para una distancia fija de  $0.75h$ .



---

## Capítulo 7

# Conclusiones

---

En el presente trabajo se ha estudiado el seguimiento solar de un arreglo de helióstatos con un mecanismo de elevación-Fresnel, así como la influencia de diferentes parámetros en la desviación del seguimiento como el número de facetas de un arreglo, la distancia del arreglo a la torre, la orientación del receptor, la latitud del sitio y el día de canteo.

Considerando la desviación máxima a lo largo del año, se encontró que existen una dependencia lineal o cuadrática respecto a el número de facetas en el arreglo, que es proporcional a el ancho total del arreglo, lo que nos lleva a conocer la desviación esperada a la hora del diseño.

Cuando se comparan los receptores verticales y horizontales, se observa que para arreglos hasta de diez facetas la dispersión en el eje  $x$  es prácticamente el mismo para ambas orientaciones y para todas las distancias. Para arreglos mayores la desviación es menor para un receptor horizontal a distancias cortas, cuando se incrementan estas distancias la diferencia es prácticamente nula entre ambos receptores y para los distintos tipos de arreglos. En el eje vertical del receptor las desviaciones son siempre menores para un plano vertical, las diferencias entre ambos planos comienzan a ser significativas arriba de diez facetas y las diferencias de la desviación entre los dos receptores analizados es proporcional a la distancia de la torre y el arreglo, entre más alejado mayor sera esta diferencia.

Para un receptor inclinado definido como plano II, de la misma forma que los receptores verticales u horizontales, se comporta practicamente igual en el eje  $x$  para arreglos menores de diez facetas. Al aumentar la distancia entre la torre y el arreglo el receptor II tiende a comportarse como un receptor vertical, distancias iguales o mayores de  $4h$ . Podría ser factible combinar los receptores horizontales e inclinado, para distancias cortas menores a  $2h$ , para obtener un mejor comportamiento.

Respecto al comportamiento del arreglo a diferentes latitudes, se observo que la desviación decrece al aumentar la latitud. Para arreglos grandes la reducción es grande, porque el crecimiento de la desviación máxima con el número de facetas tiene un comportamiento lineal con pendiente pequeña para latitudes grandes.

La mayoría de estas simulaciones fueron llevada acabo para un canteo del 21 de marzo. Al cambiar el día de canteo se encontró que el mejor día, con menor desviación, para canteo los arreglos de helióstatos cercanos son el 21 de abril o mayo, para distancias mayores el mejor es siempre el 21 de mayo. Aunque en el mejor día de canteo en ambos ejes es el 21 de marzo, hay que hacer notar que la variación en el eje  $z$  es despreciable.



La desviación del punto de impacto incrementa a partir del día de canteo hacia los solsticios, y llega a ser máxima para el solsticio de verano. Es algo esperado por una altura solar grande e intervalos angulares amplios, este comportamiento estacional es también observado en la desviación estándar.

En el análisis de flujo se encontró que aunque se tienen el valor máximo de dispersión en junio, este es el mes con mayor energía colectada debido a que en esta época los días son más largos. Un comparativo del funcionamiento de la energía por faceta se observa que la energía colectada decrece cuando el número de facetas incrementa. Sin embargo, la reducción más crítica ocurre cuando se pasa de una faceta a un arreglo de dos facetas, mientras que la reducción deja de ser significativa después de la octava faceta.

En el estado actual de la tecnología no es posible hacer una comparación con un campo de helióstatos convencional, o sería injusto ya que no existen herramientas de optimización para este tipo de arreglos. Este trabajo está enfocado para entender a detalle el comportamiento de el seguimiento con la idea de entender el tipo de aplicaciones que este tipo de arreglos podría tener. La construcción de el prototipo está bajo construcción, así que se espera tener resultados experimentales en el corto plazo.

---

---

## Apéndice A

# Cálculo de la posición solar

---

En el uso de algunas aplicaciones de la energía solar existen aplicaciones donde la posición del sol debe de localizarse con gran precisión, i.e., el heliostato de una torre central debe ser posicionado con una precisión de alrededor de  $0.05^\circ$  (Walraven, 1978). En la literatura se encuentra reportados diversos algoritmos para el cálculo de la posición solar, los cuales han sido construidos bajo aproximaciones y suposiciones sobre la geometría terrestre, estos pueden clasificarse en dos grupos. El primero es un grupo de algoritmos relativamente simple que dado el día del año, estiman los parámetros básicos de la posición solar, como la declinación solar o la ecuación del tiempo (Cooper, 1969; Spencer, 1971). El segundo esta formado por algoritmos más complejos (Walraven, 1978; Michalsky, 1988) que dada la posición calculan la posición del sol en la eclíptica, en la esfera celeste, y/o en las coordenadas locales. Típicamente el primer grupo de algoritmos es simples y adecuado para la mayoría de las aplicaciones, como por ejemplo, los paneles fotovoltaicos con seguimiento, los cuales no requieren una precisión extrema, incluso si el panel esta inclinado  $5^\circ$  sobre la normal respecto al vector solar, la salida se reduce un 0.4 % de la salida máxima. Sin embargo los sistemas de concentración solar requieren seguimiento preciso. Algunos autores utiliza 1 mrad ( $0.06^\circ$ ) en sus cálculos, limitados por las precisiones mecánicas (Michalsky, 1988).

La forma de calcular la posición solar más citada y ampliamente utilizada en la literatura (Blanco-Muriel y cols., 2001) es la formula de Spencer (1971). La cual tienen un error menor a 0.6 mrad para la declinación y de 2.5 mrad en la ecuación del tiempo. Sin embargo en los cambios máximos de declinación, que ocurren en los equinoccios, puede existir un error de menos de  $0.5^\circ$ . Cooper (1969) propuso una relación aproximada para calcular la declinación en base al día del año, la cual es menos precisa que la de Spencer (1971). Walraven (1978) presenta un algoritmo completo para el calculo de las coordenadas del sol, basada en el *The Astronomical Ephemeris*<sup>1</sup>, con precisión de 0.6 mrad, sin embargo fueron encontrados errores (Blanco-Muriel y cols., 2001) de más de 4 mrad en acimut y 0.83 mrad en la elevación, cuando se usa este algoritmo para estimar la posición solar en el periodo de 1979–1986. Diez años después de la publicación de Walraven, Michalsky (1988) publica otro algoritmo, basado también en las ecuaciones simplificadas del *Nautical Almanac*, con una precisión de  $0.01^\circ$  y validado hasta el año 2050.

---

<sup>1</sup>Pitman y Vant-Hull presentaron en 1973 un algoritmo de alta precisión, basado también en el *Almanac*, un programa llamado SUNLOC del cual no se provee el código.

Algunas recomendaciones de Blanco-Muriel y cols. (2001):

- El algoritmo de Spencer (1971), por su pobre precisión, no se recomienda para el uso en sistemas de concentración solar
- Los algoritmos de Walraven (1978) y Michalsky (1988) son suficientemente precisos para este propósito, aunque limitados en el periodo de aplicación.

Recientemente han surgido otros códigos que calculan la posición solar con una precisión astronómica como el de Blanco-Muriel y cols. (2001), Reda y Andreas (2004), y el de Grena (2008, 2012). El algoritmo propuesto por Blanco-Muriel y cols. (2001), llamado PSA por la Plataforma Solar de Almería, está validado para un periodo de 1950–2050, con una precisión mayor a los códigos anteriormente de 15 % y 20 % en los cálculos de los ángulos cenit y acimut, respectivamente.

El algoritmo de Reda y Andreas (2004) tiene un periodo de validez que va desde el año 2,000 a.C. hasta el 6,000 d.C. Contiene una gran cantidad de cálculos, por lo que resulta demasiado complejo para aplicaciones solares en general. Tal precisión es necesaria en algunas aplicaciones, como en la calibración de piranómetros, pero existen aproximaciones más sencillas que resultan lo suficientemente precisas, aunque para periodos de validez más cortos.

El algoritmo de Grena (2008) tiene errores máximos de  $0.0027^\circ$ , validado para un periodo de 2003–2023, con un número reducido de cálculos. Cuatro años después, Grena (2012), propone cinco algoritmos, aumentando el periodo de validez (2010–2110) con diferentes niveles de complejidad, y errores asociados máximos de  $0.19^\circ$  a  $0.0027^\circ$ . Grena (2008) nos muestra un resumen de los errores máximos de los principales códigos:

Tabla A.1: Errores máximos en códigos de posicionamiento solar

Código	Error máximo
Spencer	$0.25^\circ$
Pitman & Vant-Hull	$0.02^\circ$
Walraven	$0.013^\circ$
Michalsky	$0.011^\circ$
Blanco-Muriel	$0.008^\circ$

---

---

## Apéndice B

# Cambio de Coordenadas

---

### B.1. Sistemas de referencia

Considérese las rotaciones  $\gamma$  y  $\theta$  en un sistema como el mostrado en Fig. B.1. Para la rotación en  $\gamma$  tenemos que

$$X' = \cos \gamma \hat{i} + \sin \gamma \hat{j} \quad (\text{B.1})$$

$$Y' = -\sin \gamma \hat{i} + \cos \gamma \hat{j} \quad (\text{B.2})$$

$$Z' = \hat{k} \quad (\text{B.3})$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (\text{B.4})$$

Respecto a la rotación en  $\theta$  tenemos que

$$U = \hat{i} \quad (\text{B.5})$$

$$V = \cos \theta \hat{j} + \sin \theta \hat{k} \quad (\text{B.6})$$

$$W = -\sin \theta \hat{j} + \cos \theta \hat{k} \quad (\text{B.7})$$

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (\text{B.8})$$

Considérense tres vectores  $\mathbf{v}$ ,  $\mathbf{v}'$  y  $\mathbf{v}''$ . Donde  $\mathbf{v}$  esta en en sistema el sistema de referencia  $XYZ$ ,  $\mathbf{v}'$  en el sistema  $X'Y'Z'$  y  $\mathbf{v}''$  en el sistema  $UVW$ .

$$\mathbf{v}' = \mathbf{M}\mathbf{v} \quad (\text{B.9})$$

$$\mathbf{v}'' = \mathbf{N}\mathbf{v}' \quad (\text{B.10})$$

$$\mathbf{v}'' = \mathbf{NM}\mathbf{v} \quad (\text{B.11})$$

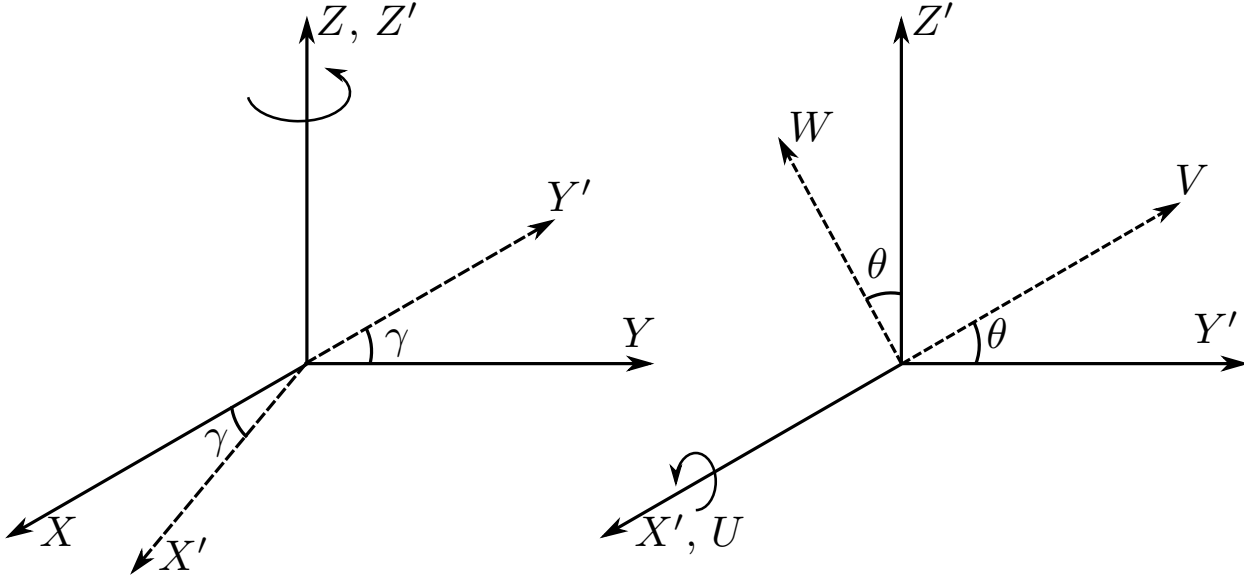


Figura B.1: Cambios de sistemas de referencia

A partir de la Ec. B.11 podemos despejar el vector  $\mathbf{v}$  y obtener que

$$\mathbf{v} = \mathbf{M}^{-1}\mathbf{N}^{-1}\mathbf{v}'' \quad (\text{B.12})$$

Donde  $\mathbf{M}$  y  $\mathbf{N}$  son las matrices de rotación, y las cuales nos permitirán cambiar de sistema de referencia conociendo las rotaciones  $\gamma$  y  $\theta$ , respectivamente.

$$\mathbf{M} = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{N} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (\text{B.13})$$

Si  $\mathbf{v} = (v_1, v_2, v_3)$  y  $\mathbf{v}'' = (v_1'', v_2'', v_3'')$  y realizando la multiplicación de matrices  $\mathbf{NM}$ , obtenemos que

$$v_1'' = v_1 \cos \gamma + v_2 \sin \gamma \quad (\text{B.14})$$

$$v_2'' = -v_1 \cos \theta \sin \gamma + v_2 \cos \theta \cos \gamma + v_3 \sin \theta \quad (\text{B.15})$$

$$v_3'' = v_1 \sin \theta \sin \gamma - v_2 \sin \theta \cos \gamma + v_3 \cos \theta \quad (\text{B.16})$$

De manera similar realizando la multiplicación matricial de  $\mathbf{M}^{-1}\mathbf{N}^{-1}$  tenemos que

$$v_1 = v_1'' \cos \gamma - v_2'' \sin \gamma \cos \theta + v_3'' \sin \gamma \sin \theta \quad (\text{B.17})$$

$$v_2 = v_1'' \sin \gamma + v_2'' \cos \gamma \cos \theta - v_3'' \cos \gamma \sin \theta \quad (\text{B.18})$$

$$v_3 = v_2'' \sin \theta + v_3'' \cos \theta \quad (\text{B.19})$$

---

La razón de reducir las operaciones es para hacer más eficiente el programa, estas últimas son las programadas en las funciones `g2l` y `l2g` que nos permite pasar entre sistemas de referencia.

---



## C.1. Programación en C

### C.1.1. Archivo principal `erun.c`

Cuadro C.1: Código `erun.c`

---

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include <time.h>
5 #include <string.h>
6 #include "ecu.h"
7 #include "base.h"
8 #include "db.h"
9 #include "vectorial.h"
10
11 int main (int argc, char *argv[])
12 {
13     rtimer();
14     char file[BUFSIZ], file_in[BUFSIZ];
15     char dir[BUFSIZ], dir_def[BUFSIZ], home[BUFSIZ];
16     char comentario[80];
17
18     char s_file[80], s_dir[80];
19     int i, j, eje, facetas, mue, result, anual, f2, nn;
20     int ii, tdia;
21     double auxd, auxm, *aux, *p, adot;
22     double lf, gap, h, dl, mu, *dxx, dx, Rx;
23     double xif, psif, *tc;
24     double wa, ha, ho;
25     db_data db;
26     ecuatorial ecu;
27     FILE *f;
28     FILE *fp;
29
30     aux = one_d_double_array(5);
31     p = one_d_double_array(3);
32     tc = one_d_double_array(3);
33
34     /* ----- */
35     /* LECTURA DEL ARCHIVO.IN */
36     /* ----- */
37
38     /* Ruta y archivo file.in */
39     strcpy(home, getenv("HOME"));
```



---

```

40  sprintf(dir_def, "%s", home);
41
42  if (argc < 2) {
43      fprintf(stderr, "Uso: %s [directorio] <archivo>\n", argv[0]);
44      fprintf(stderr, "Directorio por defecto: %s\n", dir_def);
45      exit(1);
46  }
47  else if (argc == 2) {
48      strcpy(dir, dir_def);
49      strcpy(file, argv[1]);
50      printf("dir=%s file_r=%s \n", dir, file);
51  }
52  else {
53      strcpy(dir, argv[1]);
54      strcpy(file, argv[2]);
55      printf("dir=%s file_r=%s \n", dir, file);
56  }
57
58  sprintf(file_in, "./inic/%s.in", file);
59  fp = fopen(file_in, "r");
60
61
62  /* Lectura de datos */
63  for (i = 0; i < 3; i++)
64      lee(fp, CH_LE, "%s", &comentario);
65  lee(fp, CH_LE, "Dia/mes/año: %d %d %d", &db.dia, &db.mes, &db.anno);
66  lee(fp, CH_LE, "Hor/min/seg: %d %d %d", &db.hora, &db.minuto, &db.segundo);
67  lee(fp, CH_LE, "Tiempo solar/est'andar: %d", &db.tp);
68  lee(fp, CH_LE, "Horario de verano: %d", &db.v);
69  lee(fp, CH_LE, "Latitud: %lg", &db.latitud);
70  lee(fp, CH_LE, "Longitud local: %lg", &db.Lloc);
71  lee(fp, CH_LE, "Longitud est'andar: %lg", &db.Lst);
72  db.metodo = 2; /* 1:cooper, 2:spencer */
73
74  /* Inicialización de variables */
75  variablesEc(&ecu);
76  for (i = 0; i < 3; i++)
77      lee(fp, CH_LE, "%s", &comentario);
78  lee(fp, CH_LE, "Receptor: %lg %lg %lg", &ecu.R[0], &ecu.R[1], &ecu.R[2]);
79  lee(fp, CH_LE, "Helioestado: %lg %lg %lg %d", &lf, &h, &gap, &facetas);
80  lee(fp, CH_LE, "Rotacion helioestado: %d", &mue);
81
82  if (mue == 1)
83      mu = -atan2(ecu.R[0], ecu.R[1]);
84  printf("%f\n", R2D(mu));
85
86  lee(fp, CH_LE, "Plano receptor: %d", &eje);
87  auxm = db.mes;
88  auxd = db.dia;
89  db.minuto = 0;
90  db.segundo = 0;
91  lee(fp, CH_LE, "hr/d'ia/mes: %d %d %d", &db.hora, &db.dia, &db.mes);
92  lee(fp, CH_LE, "desplazamiento: %d", &nn);
93  dxx = one_d_double_array(nn);
94  for (i = 0; i < nn; i++)
95      lee(fp, CH_LE, "desplazamiento: %lg", &dxx[i]);
96  lee(fp, CH_LE, "simulacion anual: %d", &anual);
97
98  /* Origen - Receptor */
99  ecu.o[0] = 0.0;
100  ecu.o[1] = 0.0;
101  ecu.o[2] = 0.0;
102
103  /* FIN DE LECTURA DEL ARCHIVO.IN */

```

---

---

```

104  /* ----- */
105
106  /* CÁLCULOS */
107  f2 = 0;
108  if(facetas%2 == 0)
109      f2 = 1;
110  ecu.pc = two_d_double_array(facetas+f2,3);
111  ecu.canteo = two_d_double_array(facetas-1+f2,5);
112
113  ecu.pc = heliostato(lf, gap, facetas, mu);
114
115  imprimePosicionF(facetas, ecu.pc);
116
117  /* vector del Target */
118  for (i = 0; i < 3; i++)
119      ecu.T[i] = ecu.R[i];
120  ecu.t = unitario(ecu.T);
121
122  /* ----- */
123  /* Canteo, se utiliza t/n ideal */
124  /* ----- */
125  result = db_calculate(&db);
126  for (i = 0; i < 3; i++)
127      ecu.s[i] = db.x[i];
128
129  j=0;
130  for (i = 1; i < (facetas+f2); i++) {
131      p[0] = ecu.pc[i][0];
132      p[1] = ecu.pc[i][1];
133      p[2] = ecu.pc[i][2];
134
135      if(i%2 != 0 && nn != 0){
136          dx = dxx[j];
137          j=j+1;
138          printf("dx %d\t%f\n",i, dx);
139      }
140      else if(dxx == 0){
141          dx = 0.0;
142      }
143      Rx = -dx*SIGN(p[0]);
144
145      aux = CalculaCanteo(ecu.R, ecu.T, ecu.s, p, Rx);
146      ecu.canteo[i-1][0] = aux[0];
147      ecu.canteo[i-1][1] = aux[1];
148
149      ecu.canteo[i-1][2] = aux[2];
150      ecu.canteo[i-1][3] = aux[3];
151      ecu.canteo[i-1][4] = aux[4];
152  }
153  db.mes = auxm;
154  db.dia = auxd;
155  //-----
156
157  Mdir("./dat");
158
159  /* Cálculo de la hora de amanecer */
160  wa = hSimulacion(db.dia, db.mes, db.latitud);
161  ha = 12.0 - wa;
162  ho = 12.0 + wa;
163  tdia = (int) (wa*2.0);
164  db.minuto = 0;
165  if (wa - (int) wa == 0.5)
166      db.minuto = 30;
167

```

---

---

```

168  /* ----- */
169  /* SIMULACIÓN ANUAL / DIARIA */
170  /* ----- */
171  if (anual == 0) {
172
173      /* Archivos */
174      sprintf(s_file, "./dat/R%sF%dT%d-%d.dat",file, facetas, db.dia,db.mes);
175      f = fopen(s_file, "w");
176      fprintf(f, "# faceta, hora:min, s0, s1, s2, n0, n1, n2, xi, psi, x, y, z, tau\n");
177
178      for (ii = 0; ii <= tdia; ii++) {
179          db.hora = (int) ha + ii;
180
181          /* Cálculo del vector solar */
182          result = db_calculate(&db);
183          for (i = 0; i < 3; i++)
184              ecu.s[i] = db.x[i];
185
186          /* Espejo central Punto central */
187          ecu.n = Nst(ecu.s, ecu.t);
188          ecu.psi = PSI(ecu.n);
189          ecu.xi = XI(ecu.n, ecu.psi);
190          ecu.tr = tsn(ecu.s, ecu.n);
191          ecu.Tr = PuntoTarget(ecu.R, ecu.o, ecu.tr, eje);
192
193          fprintf(f, "%d\t%d: %d\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n", \
194              0, db.hora, db.minuto, ecu.s[0], ecu.s[1], ecu.s[2], \
195              ecu.n[0], ecu.n[1], ecu.n[2], R2D(ecu.xi), R2D(ecu.psi), \
196              -(ecu.Tr[0]-ecu.R[0]),ecu.Tr[1]-ecu.R[1],ecu.Tr[2]-ecu.R[2], 0.00);
197
198          /* Centros de espejos laterales*/
199          for (i = 1; i < (facetas+f2); i++) {
200              p[0] = ecu.pc[i][0];
201              p[1] = ecu.pc[i][1];
202              p[2] = ecu.pc[i][2];
203
204              tc[0] = ecu.canteo[i-1][2];
205              tc[1] = ecu.canteo[i-1][3];
206              tc[2] = ecu.canteo[i-1][4];
207
208              ecu.prot = rot(p, ecu.xi, 0.0);
209
210              xif = ecu.xi - ecu.canteo[i-1][0];
211              psif = ecu.psi - ecu.canteo[i-1][1];
212              ecu.np = nxipsi(xif, psif);
213              ecu.tp = tsn(ecu.s, ecu.np);
214              ecu.Tp = PuntoTarget(ecu.R, ecu.prot, ecu.tp, eje);
215
216              adot = tc[0]*ecu.tp[0] + tc[1]*ecu.tp[1] + tc[2]*ecu.tp[2];
217              ecu.tau = acos(adot)*1000.;
218              fprintf(f, "%d\t%d: %d\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n", \
219                  i, db.hora, db.minuto, ecu.s[0], ecu.s[1], ecu.s[2], \
220                  ecu.np[0], ecu.np[1], ecu.np[2], R2D(xif), R2D(psif), \
221                  -(ecu.Tp[0]-ecu.R[0]),ecu.Tp[1]-ecu.R[1],ecu.Tp[2]-ecu.R[2], ecu.tau);
222              }
223
224              fprintf(f, "\n\n");
225          }
226      }
227
228      else {
229          sprintf(s_dir, "./dat/rec%sF%d/",file, facetas);
230          Mdir(s_dir);
231

```

---

---

```

232     for (db.mes = 1; db.mes < 13; db.mes++) {
233         for (db.dia = 7; db.dia < 30; db.dia = db.dia + 7) {
234
235             /* Archivos */
236             sprintf(s_dir, "./dat/rec%sF%d/", file, facetas);
237             sprintf(s_file, "R%d-%d.dat", db.dia, db.mes);
238             strcat(s_dir, s_file);
239             f = fopen(s_dir, "w");
240             fprintf(f, "# faceta, hora:min, s0, s1, s2, n0, n1, n2, xi, psi, x, y, z, tau\n");
241
242             /* Calculo de la hora de amanecer */
243             wa = hSimulacion(db.dia, db.mes, db.latitud);
244             ha = 12.0 - wa;
245             ho = 12.0 + wa;
246             tdia = (int) (wa*2.0);
247             db.minuto = 0;
248             if ( (wa - (int) wa) == 0.5)
249                 db.minuto = 30;
250
251             for (ii = 0; ii <= tdia; ii++) {
252                 db.hora = (int) ha + ii;
253
254                 /* Calculo del vector solar */
255                 result = db_calculate(&db);
256                 for (i = 0; i < 3; i++)
257                     ecu.s[i] = db.x[i];
258
259                 /* Espejo central Punto central */
260                 ecu.n = Nst(ecu.s, ecu.t);
261                 ecu.psi = PSI(ecu.n);
262                 ecu.xi = XI(ecu.n, ecu.psi);
263                 ecu.tr = tsu(ecu.s, ecu.n);
264                 ecu.Tr = PuntoTarget(ecu.R, ecu.o, ecu.tr, eje);
265
266                 fprintf(f, "%d\t%d: %d\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n", \
267                     0, db.hora, db.minuto, ecu.s[0], ecu.s[1], ecu.s[2], \
268                     ecu.n[0], ecu.n[1], ecu.n[2], R2D(ecu.xi), R2D(ecu.psi), \
269                     -(ecu.Tr[0]-ecu.R[0]), ecu.Tr[1]-ecu.R[1], ecu.Tr[2]-ecu.R[2], 0.00);
270
271                 /* Centros de espejos laterales */
272                 for (i = 1; i < (facetas+f2); i++) {
273                     p[0] = ecu.pc[i][0];
274                     p[1] = ecu.pc[i][1];
275                     p[2] = ecu.pc[i][2];
276
277                     tc[0] = ecu.canteo[i-1][2];
278                     tc[1] = ecu.canteo[i-1][3];
279                     tc[2] = ecu.canteo[i-1][4];
280
281                     ecu.prot = rot(p, ecu.xi, 0.0);
282                     xif = ecu.xi - ecu.canteo[i-1][0];
283                     psif = ecu.psi - ecu.canteo[i-1][1];
284                     ecu.np = nxipsi( xif, psif);
285                     ecu.tp = tsu(ecu.s, ecu.np);
286                     ecu.Tp = PuntoTarget(ecu.R, ecu.prot, ecu.tp, eje);
287
288                     adot = tc[0]*ecu.tp[0] + tc[1]*ecu.tp[1] + tc[2]*ecu.tp[2];
289                     ecu.tau = acos(adot)*1000.;
290                     fprintf(f, "%d\t%d: %d\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n", \
291                         i, db.hora, db.minuto, ecu.s[0], ecu.s[1], ecu.s[2], \
292                         ecu.np[0], ecu.np[1], ecu.np[2], R2D(xif), R2D(psif), \
293                         -(ecu.Tp[0]-ecu.R[0]), ecu.Tp[1]-ecu.R[1], ecu.Tp[2]-ecu.R[2], ecu.tau);
294                 }
295

```

---

---

```

296     fprintf(f, "\n\n");
297 }
298     }
299 }
300 }
301
302
303     return 0;
304 }

```

---

## C.1.2. Archivos de cabecera

Cuadro C.2: Archivo de cabecera `base.h`

---

```

1  #define PI      3.1415926535897932384626433832795028841971
2
3  /* Grados a radianes */
4  #define D2R(grados) (grados * PI/180.0)
5  /* Radianes a grados */
6  #define R2D(radianes) (radianes * 180.0/PI)
7  /* Regresa el signo -1, 1 o 0 */
8  #define SIGN(X) ((X > 0) - (X < 0))
9
10 #define MIN(a,b) (((a)<(b))?(a):(b))
11 #define MAX(a,b) (((a)>(b))?(a):(b))
12
13 int *one_d_int_array(int n);
14 int **two_d_int_array(int lx, int ly);
15 double *one_d_double_array(int n);
16 double **two_d_double_array(int lx, int ly);
17 double ***three_d_double_array(int lx, int ly, int lz);
18
19 void free_two_d_int_array(int n,int **s);
20 void free_two_d_double_array(int n,double **s);
21 void free_three_d_double_array(int lx, int ly, double ***s);
22
23 double rtimer();
24 double minut_cpu();
25
26 void _make_file_name(char s_aux[20],char s_base[80],char s_pars[150],\
27     char s_term[20],char *s_file);
28
29 void Mdir(char *name);

```

---

Cuadro C.3: Archivo de cabecera `vectorial.h`

---

```

1  double *sumV(double *x, double *y, int sig);
2  double norma(double *x);
3  double *unitario(double *x);
4  double dot(double *x, double *y);
5  double *normal(double *s, double *t);
6  double **sumM(double **A, double **B, int n,int m);

```

---

Cuadro C.4: Archivo de cabecera `db.h`

---

```

1  #ifndef __algoritmo_duffie_beckman
2  #define __algoritmo_duffie_beckman
3
4  typedef struct {
5

```

---

```

6  //-----Variables de entrada-----
7  int tp;           /* tp=0 tiempo solar, tp=1 tiempo estandar */
8  int v;           /* horario de verano, si=1, no=0 */
9  int anno;        /* cuatro digitos */
10 int mes;         /* 1 al 12 */
11 int dia;         /* dos digitos */
12 int hora;        /* 0 a 24 */
13 int minuto;      /* 0 a 59 */
14 double segundo;  /* 0 to <60 */
15 int metodo;      /* 1:cooper, 2:spencer */
16 double latitud;  /* latitud del observador (+N del ecuador) */
17 double Lloc;     /* longitud local */
18 double Lst;      /* meridiano estandar, huso horario * 15 */
19
20
21 //-----Variables de intermedias-----
22 int N;           /* día del año */
23 double B;
24 double declinacion;
25 double E;       /* ecuación del tiempo */
26 double HMS;     /* tiempo en horas */
27 double omega;   /* ángulo solar */
28 double omega_s; /* ángulo de amanecer y anochecer */
29 double dt;      /* tiempo solar - tiempo local */
30
31 //-----Variables de salida-----
32 double cenit;   /* ángulo cenital */
33 double acimut;  /* ángulo acimutal, desde el sur-oeste */
34 double *x;      /* vector solar (x[0],x[1],x[2]) */
35
36 } db_data;
37
38 //----- Funciones útiles -----
39 double r2d(double radianes);
40 double d2r(double grados);
41 double h2r(double horas);
42 double h2d(double horas);
43 double r2s(double radianes);
44 double hms(int hora, int minuto, int segundo);
45
46 int dda(int mes, int dia);
47 double B(int N);
48 double delta(int N, double B, int metodo);
49 double EdT(double B);
50 double omega(double HMS, double dt, int tp);
51 double omega_s(double latitud, double declinacion);
52 double theta_z(double latitud, double declinacion, double omega);
53 double gamma_s(double latitud, double declinacion, double omega, double cenit);
54
55 int db_calculate(db_data *db);
56
57 double amanecer(int dia, int mes, double latitud);
58 double hSimulacion(int dia, int mes, double latitud);
59 #endif

```

Cuadro C.5: Archivo de cabecera ecu.h

```

1 #include <stdio.h>
2
3 /* Leer datos de archivo */
4 #define CH_LE 0
5 int lee(FILE *fp, char , const char *, ...);
6
7 typedef struct{

```

---

```

8  double l, h;
9  double *s, *T, *t, *N, *n, *R, *r;
10 double *o, *prot, *qrot,*np, *tp;
11 double **pc, **canteo;
12 double tau;
13 double xi;
14 double psi;
15 double *tr, *Tr, *Tp;
16 } ecuatorial;
17
18 double **heliostato(double lf, double gap, int facetas, double mu);
19 int imprimePosicionF(int facetas, double **pc);
20 void variablesEc(ecuatorial *ecu);
21 double *rot(double *v, double xi, double psi);
22 double *rotmu(double *v, double mu);
23 double PSI(double *n);
24 double XI(double *n, double psi);
25 double *PuntoTarget(double *R, double *q, double *tr, int eje);
26 double *Nst(double *s, double *t);
27 double *nCanteo(double *s, double *n, double *R, double *p);
28 double *nxipsi(double xi, double psi);
29
30 double *tsn(double *s, double *n);
31 double *CalculaCanteo(double *R, double *T, double *s, double *p, double Rx);
32
33 double areaImagen(double*a, double*b, double *c, double*d);

```

---

### C.1.3. Archivos de funciones

Cuadro C.6: Archivo de funciones `base.c`

---

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <math.h>
5  #include <time.h>
6  #include <sys/types.h>
7  #include <sys/stat.h>
8  #include <unistd.h>
9
10 #define PI      3.1415926535897932384626433832795028841971
11
12 /* ----- */
13 /* Arreglos */
14 /* ----- */
15 /* 1D int */
16 int *one_d_int_array(int n) {
17     int *ptr;
18
19     ptr = (int *) calloc(n,sizeof(int));
20     return ptr;
21 }
22
23 /* 2D int */
24 int **two_d_int_array(int lx, int ly) {
25     int **ptr, i;
26
27     ptr = (int **) calloc(lx,sizeof(int*));
28     for(i=0;i<lx;++i)
29         ptr[i] = (int *) calloc(ly,sizeof(int));
30     return ptr;

```

---

```
31 }
32
33 /* 1D double */
34 double *one_d_double_array(int n) {
35     double *ptr;
36
37     ptr = (double *) calloc(n,sizeof(double));
38     return ptr;
39 }
40
41 /* 2D double */
42 double **two_d_double_array(int lx, int ly) {
43     int i;
44     double **ptr;
45
46     ptr = (double **) calloc(lx,sizeof(double*));
47     for(i=0;i<lx;++i)
48         ptr[i] = (double*) calloc(ly,sizeof(double));
49     return ptr;
50 }
51
52 /* 3D double */
53 double ***three_d_double_array(int lx, int ly, int lz) {
54     int i,j;
55     double ***ptr;
56
57     ptr = (double ***) calloc(lx,sizeof(double**));
58     for(i=0;i<lx;++i) {
59         ptr[i] = (double **) calloc(ly,sizeof(double*));
60         for(j=0;j<ly;++j) {
61             ptr[i][j] = (double *) calloc(lz,sizeof(double));
62         }
63     }
64     return ptr;
65 }
66
67 void free_two_d_int_array(int n,int **s) {
68     int i;
69
70     for (i=0;i<n;++i)
71         free(s[i]);
72 }
73
74 void free_two_d_double_array(int n,double **s) {
75     int i;
76
77     for (i=0;i<n;++i)
78         free(s[i]);
79 }
80
81 void free_three_d_double_array(int lx, int ly, double ***) {
82     int i,j;
83
84     for (i=0;i<lx;++i) {
85         for (j=0;j<ly;++j) {
86             free(s[i][j]);
87         }
88         free(s[i]);
89     }
90 }
91
92 /* ----- */
93 /* Nombre del archivos */
94 /* ----- */
```



```

95 void _make_file_name(char s_aux[20],char s_base[80],char s_pars[150],\
96     char s_term[20],char *s_file){
97     sprintf(s_file,"%s",s_base);
98     strcat(s_file,s_pars);
99     strcat(s_file,s_aux);
100    strcat(s_file,s_term);
101 }
102
103 /* ----- */
104 /* Calculo del tiempo de computo */
105 /* ----- */
106 clock_t qq_0;
107 void rtimer() {
108     qq_0=clock();
109 }
110
111 /* Devuelve el número de minutos transcurridos del CPU hasta la llamada */
112 double minut_cpu() {
113     clock_t qq_1;
114     double a,fakt=(1./60.)*(1./CLOCKS_PER_SEC);
115
116     qq_1=clock();
117     a=fakt*(qq_1-qq_0);
118
119     return(a);
120 }
121
122 /* ----- */
123 /* Crea un directorio */
124 /* ----- */
125 void Mdir(char *name){
126
127     struct stat st = {0};
128
129     if (stat(name, &st) == -1)
130         mkdir(name, 0700);
131 }
132
133 #undef PI

```

---

Cuadro C.7: Archivo de funciones vectorial.c

```

1 #include <math.h>
2 #include "base.h"
3
4 /* ----- */
5 /* Funciones vectoriales */
6 /* ----- */
7
8 /* suma vectorial, x y y vectores, sig = 1 (suma), sig = -1 (resta) */
9 double *sumV(double *x, double *y, int sig){
10     int i;
11     double *z;
12     z = one_d_double_array(3);
13
14     for (i = 0; i < 3; i++)
15         z[i] = x[i] + sig*y[i];
16
17     return z;
18 }
19
20 /* magnitud del vector */
21 double norma(double *x){
22     return sqrt(x[0]*x[0] + x[1]*x[1] + x[2]*x[2]);

```

---

```

23 }
24
25 /* vector unitario */
26 double *unitario(double *x){
27     int i;
28     double n, *u;
29
30     u = one_d_double_array(3);
31
32     n = norma(x);
33     for (i = 0; i < 3; i++)
34         u[i] = x[i]/n;
35
36     return u;
37 }
38
39 /* Producto punto */
40 double dot(double *x, double *y){
41     int i;
42     double sum;
43
44     sum = 0.0;
45     for (i = 0; i < 3; i++)
46         sum += x[i]*y[i];
47
48     return sum;
49 }
50
51 /* Cálculo de un vector normal*/
52 double *normal(double *s, double *t){
53     int i;
54     double *n;
55
56     n = one_d_double_array(3);
57     s = unitario(s);
58
59     for (i = 0; i < 3; i++)
60         n[i] = s[i] + t[i];
61
62     n = unitario(n);
63     return n;
64 }
65
66
67 /* Suma de matrix de nXm */
68 double **sumM(double **A, double **B, int n, int m){
69     int i, j;
70     double **C;
71     C = two_d_double_array(n, m);
72
73     for (i = 0; i < n; i++)
74         for (j = 0; j < m; j++)
75             C[i][j] = A[i][j] + B[i][j];
76
77     return C;
78 }

```

Cuadro C.8: Archivo de funciones `ecu.c`

```

1 #include <math.h>
2 #include "base.h"
3 #include "vectorial.h"
4 #include "ecu.h"
5 #include <stdio.h>

```

```

6
7 /* Inicialización de las variables de ecu */
8 void variablesEc(ecuatorial *ecu){
9     ecu->o = one_d_double_array(3);
10    ecu->s = one_d_double_array(3);
11    ecu->T = one_d_double_array(3);
12    ecu->t = one_d_double_array(3);
13    ecu->N = one_d_double_array(3);
14    ecu->n = one_d_double_array(3);
15    ecu->tr = one_d_double_array(3);
16    ecu->Tr = one_d_double_array(3);
17    ecu->Tp = one_d_double_array(3);
18    ecu->prot = one_d_double_array(3);
19    ecu->qrot = one_d_double_array(3);
20    ecu->np = one_d_double_array(3);
21    ecu->R = one_d_double_array(3);
22 }
23
24 /* Funciones básicas */
25 /* ----- */
26
27 /* Posicionamiento de las facetas en el plano */
28 /* l- longitud del heliostato */
29 /* facetas - número de facetas */
30 /* gap - brecha entre facetas */
31 /* mu - ángulo de rotación */
32 double **heliostato(double lf, double gap, int facetas, double mu){
33     int i, j, n, sig, f;
34     double dl, p, q, **pc;
35
36     f = 0; /* impar */
37     if (facetas%2 == 0)
38         f = 1; /* par */
39
40     /* Facetas impares */
41     if (f == 0) {
42         pc = two_d_double_array(facetas,3);
43         dl = lf + gap;
44         pc[0][0] = 0.0;
45         pc[0][1] = 0.0;
46         pc[0][2] = 0.0;
47         j=1;
48         for (i = 1; i < facetas; i++) {
49             sig = 1;
50             n = i%2;
51             if (n == 0){
52                 sig= -1;
53                 j=j-1;
54             }
55             pc[i][0] = sig*j*dl;
56             pc[i][1] = 0.0;
57             pc[i][2] = 0.0;
58             j++;
59         }
60     }
61
62     /* Facetas pares */
63     else{
64
65         pc = two_d_double_array(facetas+f,3);
66         dl = lf + gap;
67         pc[0][0] = 0.0;
68         pc[0][1] = 0.0;
69         pc[0][2] = 0.0;

```

---

```

70
71     j=1;
72     for (i = 1; i < (facetas+f); i++) {
73         sig = 1;
74         n = i%2;
75         if (n == 0){
76             sig= -1;
77             j=j-1;
78         }
79
80         if (i == 1 || i == 2) {
81             pc[i][0] = sig*j*dl/2.;
82             pc[i][1] = 0.0;
83             pc[i][2] = 0.0;
84         }
85         else{
86             pc[i][0] = sig*(dl/2.+ (j-1)*dl);
87             pc[i][1] = 0.0;
88             pc[i][2] = 0.0;
89         }
90         j++;
91     }
92 }
93
94
95 if (mu != 0) {
96     for (i = 1; i < facetas; i++) {
97         p = pc[i][0]*cos(mu) - pc[i][1]*sin(mu);
98         q = pc[i][0]*sin(mu) + pc[i][1]*cos(mu);
99         pc[i][0] = p;
100        pc[i][1] = q;
101    }
102 }
103
104
105 return pc;
106 }
107
108
109 int imprimePosicionF(int facetas, double **pc){
110     int i, f;
111
112     f = 0;          /* impar */
113     if (facetas%2 == 0)
114         f = 1;      /* par */
115
116     for (i = 0; i < (facetas+f); i++)
117         printf("%d\t%f\t%f\t%f\n", i, pc[i][0], pc[i][1], pc[i][2]);
118
119     return 0;
120 }
121
122
123 /* Cálculo de los ángulos psi/xi a partir de la normal */
124 double PSI(double *n){
125     return asin(n[0]);
126 }
127
128 double XI(double *n, double psi){
129     return acos(n[2]/cos(psi));
130 }
131
132 //rotación del vector v en xi/psi
133 double *rot(double *v, double xi, double psi){

```

---

---

```

134 double *vrot;
135 vrot = one_d_double_array(3);
136 vrot[0] = v[0]*cos(psi)+v[1]*sin(psi)*sin(xi) - v[2]*sin(psi)*cos(xi);
137 vrot[1] = v[1]*cos(xi) + v[2]*sin(xi);
138 vrot[2] = v[0]*sin(psi) - v[1]*cos(psi)*sin(xi) + v[2]*cos(psi)*cos(xi);
139 return vrot;
140 }
141
142 //rotación mu del acomodo de los puntos iniciales
143 double *rotmu(double *v, double mu){
144     double *vrot;
145     vrot = one_d_double_array(3);
146     vrot[0] = v[0]*cos(mu) - v[1]*sin(mu);
147     vrot[1] = v[0]*sin(mu) + v[1]*cos(mu);
148     vrot[2] = v[3];
149     return vrot;
150 }
151
152
153 /* Cálculo de la posición en el target */
154 /* eje 0-x, 1-y, 2-z */
155 double *PuntoTarget(double *R, double *q, double *tr, int eje){
156     int i;
157     double d, *T;
158     T=one_d_double_array(3);
159
160     d = (R[eje]-q[eje]) /tr[eje];
161
162     for (i = 0; i < 3; i++)
163         T[i] = q[i] + d*tr[i];
164
165     return T;
166 }
167
168 /* Calculo de la normal unitaria */
169 double *Nst(double *s, double *t){
170     double *n;
171     n = one_d_double_array(3);
172     n = sumV(s,t,1);
173     n = unitario(n);
174     return n;
175 }
176
177
178 /* Cálculo de los ángulos xi/psi de canteo */
179 /* s-vector solar; R-punto del target; o-origen */
180 double *nCanteo(double *s, double *n, double *R, double *p){
181     double xi, psi;//, epsilon;
182     double *prot, *tq, *nq;
183
184     prot = one_d_double_array(3);
185     tq = one_d_double_array(3);
186     nq = one_d_double_array(3);
187
188     psi = PSI(n);
189     xi = XI(n, psi);
190
191     /* Unicamente rota en xi, ya que psi es independiente */
192     prot = rot(p, xi, 0.0);
193     tq = sumV(R,prot,-1);
194     tq = unitario(tq);
195     nq = Nst(s, tq);
196     return nq;
197 }

```

---

---

```

198
199 double *nxipsi(double xi, double psi){
200     double *np;
201     np = one_d_double_array(3);
202
203     /* Recalcular n y t con errores */
204     np[0] = sin(psi);
205     np[1] = -cos(psi)*sin(xi);
206     np[2] = cos(psi)*cos(xi);
207     return np;
208 }
209
210 /* Plano horizontal, rotación en z */
211 double areaImagen(double*a, double*b, double *c, double*d){
212
213     double area;
214     double *u,*v, *vp;
215     u = one_d_double_array(3);
216     v = one_d_double_array(3);
217     vp = one_d_double_array(3);
218
219
220     u = sumV(d,c,-1);
221     v = sumV(b,c,-1);
222
223     vp[0] = v[1];
224     vp[1] = -v[0];
225     vp[2] = v[2];
226
227     area = dot(u,vp);
228
229     return fabs(area);
230 }
231
232 /* Cálculo de canteo */
233 double *CalculaCanteo(double *R, double *T, double *s, double *p, double Rx){
234     //int i;
235     double xi, psi;
236     double psip, xip;
237     double *t, *tc, *n, *np, *canteo, *R2;
238
239     /* Modificación en el canteo por Rx */
240     R2 = one_d_double_array(3);
241     R2[0] = R[0] - Rx;
242     R2[1] = R[1];
243     R2[2] = R[2];
244
245     t = one_d_double_array(3);
246     tc = one_d_double_array(3);
247     n = one_d_double_array(3);
248     np = one_d_double_array(3);
249     canteo = one_d_double_array(5);
250
251     t = unitario(T);
252     n = Nst(s, t);
253
254     psi = PSI(n);
255     xi = XI(n, psi);
256
257     np = nCanteo(s, n, R2, p);
258     psip = PSI(np);
259     xip = xi;
260     canteo[1] = psi - psip;
261     canteo[0] = 0.0;

```

---

---

```

262
263     /* Recalcular n con psip y xi */
264     np = nxipsi(xi, psip);
265     tc = tsu(s, np);
266     canteo[2]=tc[0];
267     canteo[3]=tc[1];
268     canteo[4]=tc[2];
269
270     return canteo;
271 }
272
273 /* Calcula el vector de reflexión a partir del vector solar y el vector normal */
274 double *tsn(double *s, double *n){
275     int i;
276     double DOT;
277     double *t;
278
279     t = one_d_double_array(3);
280
281     DOT = dot(s,n);
282     for (i = 0; i < 3; i++)
283         t[i] = 2*DOT*n[i]-s[i];
284
285     return t;
286 }

```

---

Cuadro C.9: Archivo de funciones db.c

---

```

1 #include <math.h>
2 #include <stdlib.h>
3 #include "db.h"
4
5 #define PI          3.1415926535897932384626433832795028841971
6
7 /* Funciones básicas */
8 /* ----- */
9
10 /* radianes a grados */
11 double r2d(double radianes){
12     return (180.0/PI)*radianes;
13 }
14
15 /* grados a radianes */
16 double d2r(double grados){
17     return (PI/180.0)*grados;
18 }
19
20 /* horas a radianes */
21 double h2r(double horas){
22     return horas*PI/12.0;
23 }
24
25 /* horas a grados */
26 double h2d(double horas){
27     return horas*180.0/12.0;
28 }
29
30 /* radianes a segundos */
31 double r2s(double radianes){
32     return radianes*12.0/PI*3600.0;
33 }
34
35 /* hora + min + seg, en horas */
36 double hms(int hora, int minuto, int segundo){
37     return hora + minuto/60.0 + segundo/3600.0;
38 }

```

---

---

```

37 }
38
39 /* Validación */
40 /* ----- */
41 int validate_inputs(db_data *db)
42 {
43     if ((db->anno < -2000) || (db->anno > 6000)) return 1;
44     if ((db->mes < 1) || (db->mes > 12)) return 2;
45     if ((db->dia < 1) || (db->dia > 31)) return 3;
46     if ((db->hora < 0) || (db->hora > 24)) return 4;
47     if ((db->minuto < 0) || (db->minuto > 59)) return 5;
48     if ((db->segundo < 0) || (db->segundo >=60)) return 6;
49
50     if (fabs(db->latitud) > 90) return 7;
51
52     return 0;
53 }
54
55 /* Funciones */
56 /* ----- */
57 const int MES[12] = {0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334};
58
59 // día del año
60 int dda(int mes, int dia)
61 {
62     return MES[mes - 1] + dia ;
63 }
64
65 double B(int N){
66     return 2.0*PI*(N-1)/365.0; /* radianes */
67 }
68
69 /* Declinación */
70 double delta(int N, double B, int metodo){
71     double decl;
72
73     switch(metodo){
74         case 1: /* cooper */
75             decl = 23.45 * sin(2.0*PI*(284 + N)/365.0);
76             decl = d2r(decl);
77             break;
78         case 2: /* spencer */
79             decl = 0.006918-0.399912*cos(B)+0.070257*sin(B)- 0.006758*cos(2.0*B)+ 0.000907*sin(2.0*B)- 0.002697*cos
                (3.0*B)+0.001480*sin(3.0*B);
80             break;
81     }
82     return decl;
83 }
84
85 /* Ecuación del tiempo [minutos] */
86 double EdT(double B){
87     return 229.2 * (0.000075 + 0.001868*cos(B) - 0.032077*sin(B) - 0.014615*cos(2*B) - 0.04089*sin(2*B));
88 }
89
90 /* ángulo de amanecer y anochecer [hrs]*/
91 double omega_s(double latitud, double declinacion){
92     double x;
93     x = acos( -tan( d2r(latitud) )*tan(declinacion) );
94     x = r2d(x)/15.0;
95     return x;
96 }
97
98 /* Direfencia entre tiempo solar y estandar [min]*/
99 double dtiempo(double E, double Lst, double Lloc, int v){

```

---



---

```

100     return 4.0*(Lst - Lloc) + E - 60.0*v;
101 }
102
103 /* omega [radianes]*/
104 double omega(double HMS, double dt, int tp){
105     return d2r( 15.0*(HMS-12.0 + tp*dt/60.0) );
106 }
107
108 /* ángulo cenital [radianes] */
109 double theta_z(double latitud, double declinacion, double omega){
110     latitud = d2r(latitud);
111     return acos( cos(latitud)*cos(declinacion)*cos(omega) + sin(latitud)*sin(declinacion));
112 }
113
114 /* ángulo acimital */
115 double gamma_s(double latitud, double declinacion, double omega, double cenit){
116     double Ss, Cs, x, g;
117     latitud = d2r(latitud);
118     Ss = sin(omega)*cos(declinacion)/sin(cenit);
119     Cs = (sin(latitud)*cos(declinacion)*cos(omega) - cos(latitud)*sin(declinacion) )/ sin(cenit);
120     x = sin(omega)*cos(declinacion)/( sin(latitud)*cos(omega)*cos(declinacion) - cos(latitud)*sin(declinacion));
121     g = atan(x);
122     if(Cs >= 0)
123         return g;
124     else if(Cs < 0 & Ss < 0)
125         return g - PI;
126     else if(Cs < 0 & Ss > 0)
127         return g + PI;
128     return 0;
129 }
130
131 double *vsolar(double cenit, double acimut){
132     double *x;
133
134     x = (double *)calloc(3, sizeof(double));
135     x[0] = -sin(cenit)*sin(acimut);
136     x[1] = -sin(cenit)*cos(acimut);
137     x[2] = cos(cenit);
138
139     return x;
140 }
141
142 /* Cálculo de los ángulos solare */
143 /* ----- */
144 int db_calculate(db_data *db)
145 {
146     int result;
147
148     result = validate_inputs(db);
149
150     db->N = dda(db->mes, db->dia);
151     db->B = B(db->N);
152     db->declinacion = delta(db->N, db->B, db->metodo);
153     db->E = EdT(db->B);
154     db->HMS = hms(db->hora, db->minuto, db->segundo);
155     db->dt = dtiempo(db->E, db->Lst, db->Lloc, db->v);
156
157     db->omega = omega(db->HMS, db->dt, db->tp);
158     db->omega_s = omega_s(db->latitud, db->declinacion);
159     db->cenit = theta_z(db->latitud, db->declinacion, db->omega);
160     db->acimut = gamma_s(db->latitud, db->declinacion, db->omega, db->cenit);
161
162     db->x = vsolar(db->cenit, db->acimut);
163

```

---

---

```

164     return result;
165 }
166
167
168 /* Cálculo del amanecer [hrs]*/
169 /* ----- */
170 double amanecer(int dia, int mes, double latitud){
171     int N;
172     double b, d, omegas;
173
174     N = dda(mes, dia);
175     b = B(N);
176     d = delta(N, b, 2);
177
178     omegas = omega_s(latitud, d);
179
180     return omegas;
181 }
182
183
184 /* Calculo de la hora de simulación */
185 /* Apartir de la hora de amanecer - 15 grados */
186 double hSimulacion( int dia, int mes, double latitud){
187     double wa, wae;
188
189     wa = amanecer(dia, mes, latitud);
190     wae = wa - (int) wa;
191     if (wae >= 0.5 )
192         wa = (int) wa + 0.5;
193     else
194         wa = (int) wa;
195
196     /* La simulación se realiza 15grad después del amanecer */
197     /* y 15grad antes del anochecer. 15 grad/hr */
198     wa = wa - 1.0;
199
200     return wa;
201 }

```

---

## C.2. Script LK

Cuadro C.10: Script LK para la simulación del día 21 de junio, para un arreglo H4 a una posición 0.75h.

---

```

1 cwd("~/path/AF4-6");
2 files = ['6:30', '7:30', '8:30', '9:30', '10:30', '11:30', '12:30', '13:30', '14:30', '15:30', '16:30', '17:30'
3         ];
4 // Cálculo de irradiancia
5 G0 = 1367; // Constante solar[W/m2]
6 tau = 0.7; // transmitancia de cielo claro
7 h = 200; // altitud [m]
8
9 AM = define(h, theta){
10     d = cos(theta) + 0.5057*(96.080- theta)^(-1.634);
11     return exp(-0.0001184*h)/d;
12 };
13
14 G = define(G0, tau, am){
15     x = am^0.678;

```

---

---

```

16  return G0*tau^x;
17 };
18
19 for( k=0; k < #files; k++){
20     rut=cwd() + "/";
21     name = ("data_" + files[k]);
22     outln("Nombre del archivo = " + name);
23
24     //Read file
25     file = open(rut + "ini/" + name + ".in", "r");
26     line = "";
27     read_line(file, line);
28     read_line(file, line);
29     nf = to_int(line);
30
31     /*-----
32         Variables
33     -----*/
34
35     // Variable
36     phi = 29.02845; //latitud
37
38     // Vector Solar
39     read_line(file, line);
40     s = split(line, " ");
41     outln("Vector solar = " + s);
42
43     theta = acos(s[2]);
44     I = G(G0, tau, AM(h, theta));
45     outln("Irradiancia = " + I);
46
47     //Receptor
48     //*****
49
50     //Posición
51     T[0] = 0.0;
52     T[1] = 0.0;
53     T[2] = 35.0;
54
55     //aimpoint
56     apT[0] = 0.0;
57     apT[1] = -1.0;
58     apT[2] = T[2];
59
60     //Tamaño
61     w = 6.0;
62     h = 6.0;
63
64     // Helistatos
65     //*****
66     f = alloc(nf,3);
67     n = alloc(nf,3);
68
69     for(i = 0; i < nf; i++){
70         read_line(file, line);
71         v = split(line, " ");
72         //out(v);
73         f[i][0] = to_real(v[0]);
74         f[i][1] = to_real(v[1]);
75         f[i][2] = to_real(v[2]);
76     }
77
78     for(i = 0; i < nf; i++){
79         read_line(file, line);

```

---

---

```

80  v = split(line, " ");
81  n[i][0] = to_real(v[0]);
82  n[i][1] = to_real(v[1]);
83  n[i][2] = to_real(v[2]);
84 }
85
86 aimpoint = alloc(nf, 3);
87 for(i = 0; i < nf; i++){
88     for(j = 0; j < 3; j++){
89         aimpoint[i][j] = f[i][j] + n[i][j];
90     }
91 }
92
93 // dimensiones
94 lw = 1.0;
95 lh = 1.0;
96
97 /*-----
98     Forma solar
99     -----*/
100 Sun.shape = 'g';
101 Sun.sigma = 2.73;
102 Sun.x = s[0];
103 Sun.y = s[1];
104 Sun.z = s[2];
105 sunopt(Sun);
106
107 /*-----
108     Propiedades ópticas
109     -----*/
110 clearoptics();
111
112 addoptic('espejo');
113 opticopt( 'espejo', 1,
114 {'refl' = 1, 'trans' = 0.0001, 'errslope' = 2.64, 'errspec'=0.0001});
115
116 /*-----
117     Etapas
118     -----*/
119
120 clearstages();
121
122 /*-----
123     Helióstató
124     -----*/
125
126 addstage('heliostato');
127 activestage('heliostato');
128
129 for(i = 0; i < nf; i++){
130     addelement(); // adds element to current stage
131     x = f[i][0];
132     y = f[i][1];
133     z = f[i][2];
134     apx = aimpoint[i][0];
135     apy = aimpoint[i][1];
136     apz = aimpoint[i][2];
137     elementopt( i, {'en'=true, 'x' = x , 'y' = y, 'z' = z,
138         'ax' = apx, 'ay' = apy, 'az' = apz,
139         'zrot'=0, 'aper'=['r',lw,lh,0,0,0,0,0,0],
140         'surf'=['f', 0, 0,0,0,0,0,0,0],
141         'optic'='espejo' });
142 }
143

```

---

---

```

144 /*-----
145     Receptor
146     -----*/
147
148 addstage( 'receptor' );
149 activestage( 'receptor' );
150
151 addelement(); // adds element to current stage
152 elementopt( 0, {'en' = true, 'x' = T[0], 'y' = T[1], 'z' = T[2] ,
153     'ax' = apT[0], 'ay' = apT[1], 'az' = apT[2],
154     'zrot'=0, 'aper'=['r', w, h,0,0,0,0,0,0],
155     'surf'=['f',0,0,0,0,0,0,0,0],
156     'optic'='espejo'} );
157
158 rut = rut + name + "/";
159 mkdir(rut, true);
160 close(file);
161
162 /*-----
163     Simulación
164     -----*/
165 // Rayos
166 n =5000000;
167 traceopt({ 'rays' = n } );
168 trace();
169
170 stage = 1;
171 element = 0;
172 nBinX = 50;
173 nBinY = 50;
174 DNI = I;
175 dx = w/nBinX;
176 dy = h/nBinY;
177
178 // Exportar matriz de flujo
179 // *****
180 fs = rut + "matriz.dat";
181 file_out = open(fs, "w");
182
183 // La función 'elementstats'
184 // *****
185 stat = elementstats( stage, element, nBinX, nBinY, DNI, false );
186 da = stat.bin_size_x * stat.bin_size_y;
187
188 write_line(file_out, "## Muestra el numero de rayos que golpean cada bin");
189 write_line(file_out, "## dx =" + dx + ", dy=" + dy);
190 flux = "";
191 for (r = 0 ; r < #stat.flux ; r++){
192     for (c = 0 ; c < #stat.flux[r]; c++){
193         aux = stat.flux[c][r] * stat.power_per_ray/da;
194         flux = flux + aux + "\t";
195     }
196     write_line(file_out, flux);
197     flux = "";
198 }
199 close(file_out);
200
201 // Exportar archivo de estadísticas
202 // *****
203 fs = rut + "stats.dat";
204 file_out = open(fs, "w");
205
206 stat.flux = null;
207 stat.xvalues = null;

```

---

```
208 stat.yvalues = null;
209 keys = @stat;
210 write_line(file, "## Todas las estadisticas:");
211 for (i = 0; i < #keys; i++)
212     write_line( file_out, "# " + keys[i] + " = " + stat{keys[i]} );
213 write_line(file_out, "# ray hits on target: " + rayhits('1', '0'));
214 close(file_out);
215
216 //Guarda el archivo de trabajo
217 //*****
218 name2 = "H" + nf;
219 save_project(rut + name2 + ".stinput");
220
221 }
```

---



# Referencias

---

- Alexopoulos, S., y Hoffschmidt, B. (2013). Concentrating receiver systems (CRS). En C. Richter, D. Lincot, y C. Gueymard (Eds.), *Solar energy* (pp. 29–71). Springer New York.
- Amsbeck, L., Buck, R., Pfahl, A., y Uhlig, R. (2007). Optical performance and weight estimation of a heliostat with ganged facets. *Journal of Solar Energy Engineering*, 130(1), 011010. doi: 10.1115/1.2804630
- Biggs, F., y Vittitoe, C. N. (1979). Helios model for the optical behavior of reflecting solar concentrators. *NASA STI/Recon Technical Report N*, 79, 31806.
- Blanco, M. J., Amieva, J. M., y Mancillas, A. (2005). The tonatiuh software development project: An open source approach to the simulation of solar concentrating systems. En *Asme 2005 international mechanical engineering congress and exposition* (pp. 157–164).
- Blanco-Muriel, M., Alarcón-Padilla, D. C., López-Moratalla, T., y Lara-Coira, M. (2001). Computing the solar vector. *Solar Energy*, 70(5), 431 - 441. doi: 10.1016/S0038-092X(00)00156-0
- Buck, R., y Teufel, E. (2009). Comparison and optimization of heliostat canting methods. *Journal of Solar Energy Engineering*, 131(1), 011001.
- Chavez, K., Sproul, E., y Yellowhair, J. (2012). Development and analysis of the heliostat focusing and canting enhancement technique for full heliostat alignments. En *Asme 2012 6th international conference on energy sustainability* (pp. 237–246).
- Cooper, P. (1969). The absorption of radiation in solar stills. *Solar Energy*, 12(3), 333 - 346. doi: 10.1016/0038-092X(69)90047-4
- Domínguez-Bravo, C.-A., Bode, S.-J., Heiming, G., Richter, P., Carrizosa, E., Fernández-Cara, E., ... Gauché, P. (2016). Field-design optimization with triangular heliostat pods. *AIP Conference Proceedings*, 1734(1), 070006. doi: 10.1063/1.4949153
- Dopos, A. (2017, January). Lk scripting language reference [Manual de software informático].
- Duffie, J. A., y Beckman, W. A. (2013). *Solar engineering of thermal processes*. John Wiley & Sons.



- Dunham, M., Kasetty, R., Mathur, A., y Lipiński, W. (2013). Optical analysis of a heliostat array with linked tracking. *Journal of Solar Energy Engineering*, 135(3), 034501. doi: 10.1115/1.4023593
- Garcia, P., Ferriere, A., y Bezan, J.-J. (2008, marzo). Codes for solar flux calculation dedicated to central receiver system applications: A comparative review. *Solar Energy*, 82(3), 189–197.
- Göttsche, J., Hoffschmidt, B., Schmitz, S., Sauerborn, M., Buck, R., Teufel, E., . . . Rebholz, C. (2010). Solar concentrating systems using small mirror arrays. *Journal of Solar Energy Engineering*, 132(1), 011003. doi: 10.1115/1.4000332
- Grena, R. (2008). An algorithm for the computation of the solar position. *Solar Energy*, 82(5), 462 - 470. doi: 10.1016/j.solener.2007.10.001
- Grena, R. (2012). Five new algorithms for the computation of sun position from 2010 to 2110. *Solar Energy*, 86(5), 1323 - 1337. doi: 10.1016/j.solener.2012.01.024
- Ho, C. K. (2008). Software and codes for analysis of concentrating solar power technologies. *Sandia National Laboratories*.
- IRENA. (2018). *Renewable power generation costs in 2017* (Inf. Téc.). International Renewable Energy Agency.
- Kistler, B. L. (1986). *A user's manual for delsol3: A computer code for calculating the optical performance and optimal system design for solar thermal central receiver plants* (Inf. Téc.). Sandia National Labs., Livermore, CA (USA).
- Kolb, G. J., Davenport, R., Gorman, D., Lumia, R., Thomas, R., y Donnelly, M. (2007). Heliostat cost reduction. *ASME 2007 Energy Sustainability Conference*, 1077–1084. doi: 10.1115/ES2007-36217
- Leonardi, E., y D' Aguanno, B. (2014). Concentrating solar energy technologies. En G. Cao y R. Orrù (Eds.), *Current environmental issues and challenges* (p. 113-126). Springer Netherlands.
- Li, L., Coventry, J., Bader, R., Pye, J., y Lipiński, W. (2016). Optics of solar central receiver systems: a review. *Optics express*, 24(14), A985–A1007. doi: 10.1364/OE.24.00A985
- Michalsky, J. J. (1988). The astronomical almanac's algorithm for approximate solar position (1950–2050). *Solar Energy*, 40(3), 227 - 235. doi: 10.1016/0038-092X(88)90045-X
- Ortega, J. I., Burgaleta, J. I., y Téllez, F. M. (2008). Central receiver system solar power plant using molten salt as heat transfer fluid. *Journal of Solar Energy Engineering*, 130(2), 024501. doi: 10.1115/1.2807210
-

- 
- Pettit, R. B. (1977). Characterization of the reflected beam profile of solar mirror materials. *Solar Energy*, 19(6), 733–741.
- Pfahl, A. (2013, Abr). Survey of heliostat concepts for cost reduction. *Journal of Solar Energy Engineering*, 136(1), 014501. doi: 10.1115/1.4024243
- Pfahl, A., Coventry, J., Röger, M., Wolfertstetter, F., Vásquez-Arango, J. F., Gross, F., ... Liedke, P. (2017). Progress in heliostat development. *Solar Energy*, 152, 3 - 37. (Progress in Solar Energy Special Issue: Concentrating Solar Power (CSP)) doi: 10.1016/j.solener.2017.03.029
- R Core Team. (2015). R: A language and environment for statistical computing [Manual de software informático]. Vienna, Austria.
- Rabl, A. (1985). *Active solar collectors and their applications*. Oxford University Press on Demand.
- Reda, I., y Andreas, A. (2004). Solar position algorithm for solar radiation applications. *Solar Energy*, 76(5), 577 - 589. doi: 10.1016/j.solener.2003.12.003
- Ren, L., Wei, X., Lu, Z., Yu, W., Xu, W., y Shen, Z. (2014). A review of available methods for the alignment of mirror facets of solar concentrator in solar thermal power system. *Renewable and Sustainable Energy Reviews*, 32(0), 76 - 83.
- Schell, S. (2011). Design and evaluation of esolar's heliostat fields. *Solar energy*, 85(4), 614–619. doi: 10.1016/j.solener.2010.01.008
- Spencer, J. (1971). Fourier series representation of the position of the sun. *Search*, 2(5), 172–172.
- Vant-Hull, L. (2012). Central tower concentrating solar power (csp) systems. En *Concentrating solar power technology* (pp. 240–283). Elsevier. doi: 10.1533/9780857096173.2.240
- Walraven, R. (1978). Calculating the position of the sun. *Solar Energy*, 20(5), 393 - 397. doi: 10.1016/0038-092X(78)90155-X
- Wendelin, T., Dobos, A., y Lewandowski, A. (2013). Soltrace: A ray-tracing code for complex solar optical systems. *Contract*, 303, 275–3000.
-