

E.T.S. de Ingeniería Industrial,  
Informática y de Telecomunicación

Desarrollo e implementación de nuevos  
modelos solares para el trazador de rayos  
Tonatiuh



Máster Universitario en  
Ingeniería Industrial

Trabajo Fin de Máster

**Autor:** Javier Arregui Ibáñez

**Director:** Patricia Aranguren

**Codirector:** Iñigo Les Aguerrea

Pamplona, junio de 2018



**CENER**  
ADitech

GENTRO NACIONAL DE  
ENERGÍAS RENOVABLES

*“ Desarrollo e implementación de nuevos modelos  
solares para el trazador de rayos Tonatiuh ”*



## *Abstract*

*Concentrating solar power plants (CSP) use mirrors to concentrate the sun's rays to harness the solar energy available. One of the most important parts of the electricity generation with CSP is how to concentrate the solar rays and understand the optical phenomenon. Solar sunshape has enough influence in solar energy resource with CSP. A new achieve has to be accomplished implementing new solar sunshapes in a ray-tracing software. Tonatiuh was created by Centro Nacional de Energías Renovables (CENER) an must be improved. The implementation of two sunshapes has been made with programming language C++. Another objective of the study is to know how the different solar sun shapes take part in the IAM results. IAM is the indicer angle modifier and it is a coefficient that is used in optical simulations to reduce computational cost.*



**CENER**  
ADitech

GENTRO NACIONAL DE  
ENERGÍAS RENOVABLES

*“ Desarrollo e implementación de nuevos modelos  
solares para el trazador de rayos Tonatiuh ”*



## Índice

Índice de figuras .....	6
Prólogo .....	8
1. Objetivo del proyecto .....	8
2. Introducción .....	9
2.1. Situación energética en el mundo.....	9
2.2. Energías renovables.....	13
2.3. Energía solar térmica y de concentración .....	14
2.3.1. Solar térmica de baja y media temperatura. ....	14
2.3.2. Energía solar termoeléctrica o de concentración.....	15
2.3.3. Otras aplicaciones de las tecnologías de concentración solar .....	22
3. Recurso solar .....	23
3.1. El Sol.....	23
3.2. La atmósfera y su influencia en la radiación solar .....	25
3.3. Instrumentos de medición .....	27
4. Estado del arte en modelos solares.....	29
4.1. Pillbox .....	29
4.2. Buie .....	30
4.3. Solar Limb Darkening.....	32
4.4. Kuiper.....	35
4.5. Elección de los modelos a implementar.....	36
5. Estado del arte en software de simulación .....	37
5.1. Trazador de rayos basados en el método de Monte Carlo .....	40
5.2. Método de convolución.....	44
6. Implementación en Tonatiuh.....	45
6.1. Gaussian sunshape.....	46
6.1.1. Descripción.....	46
6.1.2. Problemática.....	47
6.1.3. Desarrollo matemático .....	47
6.1.4. Método de generación de números aleatorios .....	48
6.1.5. Implementación y funcionamiento del código .....	50
6.2. User defined sunshape.....	51
6.2.1. Descripción.....	51
6.2.2. Desarrollo matemático .....	53
6.2.3. Implementación .....	54



7.	Validación de los modelos .....	55
7.1.	Modelo Gaussiano.....	56
7.1.1.	Validación mediante Tonatiuh .....	56
7.1.2.	Comparación con SolTrace .....	57
7.2.	Modelo User defined.....	61
7.2.1.	Validación mediante Tonatiuh .....	61
7.2.2.	Comparación con SolTrace .....	65
8.	Bibliografía.....	68
	ANEXOS .....	70



## Índice de figuras

Figura 1: Producción global estimada de generación renovable.....	10
Figura 2: Potencia instalada de CSP actual y prevista (Fuente: SolarPACES).....	11
Figura 3: Potencia instalada en cada país de cada tipo de tecnología hasta 2016 (Fuente: SolarPACES).....	11
Figura 4: Proyectos de CSP de torre con almacenamiento ( Fuente: CSP Today Global Tracker).....	12
Figura 5: Perspectiva del coste medio de las plantas CSP (Fuente: Ren21).....	13
Figura 6: Esquema básico de una instalación de energía solar térmica .....	14
Figura 7: Sistema de espejos diseñado por Arquímedes para quemar las velas de los romanos.....	16
Figura 8: Primera planta desaladora solar del mundo, diseñada por Charles Wilson (Chile, 1872).....	17
Figura 9: El experimento de Shuman.....	17
Figura 10: Planta Solar de Almería (PSA).....	18
Figura 11: PS10 y PS20 .....	19
Figura 12: Clasificación tipo de concentradores.....	20
Figura 13: Esquema general de una planta solar termoelectrónica.....	21
Figura 14: Esquema de una planta con sistema de almacenamiento activo directo .....	22
Figura 15: Esquema de una planta con sistema de almacenamiento activo indirecto.....	22
Figura 16: Estructura del Sol.....	24
Figura 17: Variación de la radiación extraterrestre.....	25
Figura 18: Espectro solar y la influencia de la atmósfera .....	25
Figura 19: Componentes de la radiación solar terrestre.....	26
Figura 20: Referencias de la posición solar .....	27
Figura 21: Principio de funcionamiento de los pirheliómetros Kipp & Zonen.....	28
Figura 22: Piranómetro con banda de sombra para medir la radiación .....	28
Figura 23: Perfil de la superficie Pillbox .....	29
Figura 24: Perfil de la superficie Buie .....	31
Figura 25: Representación geométrica del Limb Darkening .....	33
Figura 26: Aplicación de la ecuación 8 para diferentes coeficientes de oscurecimiento.....	33
Figura 27: Aproximación datos empíricos a la ecuación 9 para el computo de todas las longitudes de onda .....	34
Figura 28: Distribución de la intensidad en función del ángulo $\theta$ para el modelo de Limb Darkening.....	34
Figura 29: Perfiles extraterrestres para diferentes longitudes de onda .....	36
Figura 30: Pérdidas ópticas .....	37
Figura 31: Efecto coseno.....	38
Figura 32: Desbordamiento óptico.....	38



Figura 33: Pérdidas anuales en un campo de heliostatos .....	39
Figura 34: Valores del error típicos .....	40
Figura 35: Método de Monte Carlo.....	41
Figura 36: Solar ray tracing (Fuente: SFERA) .....	42
Figura 37: Representación del escenario con los rayos .....	42
Figura 38: Software que utilizan el método de trazador de rayos.....	43
Figura 39: Software que utilizan el método de convolución.....	45
Figura 40: Comparativa Ray-tracing vs Convolution .....	45
Figura 41: Sistema de coordenadas locales usado para la implementación del modelo Gaussiano .....	46
Figura 42: Función de densidad de probabilidad .....	48
Figura 43: Procedimiento aceptación-rechazo .....	49
Figura 44: Método de aceptación y rechazo .....	49
Figura 45: Diagrama de bloques del algoritmo Gaussian Sunshape.....	51
Figura 46: Función de densidad de probabilidad .....	54
Figura 47: Diagrama de bloques del algoritmo User Defined Sunshape .....	55
Figura 48: Distribución Gaussiana del flujo incidente en un heliostato .....	56
Figura 49: Comparación entre distribución Gaussiana y datos obtenidos .....	57
Figura 50: Error entre el $\sigma$ teórico y el $\sigma$ obtenido tras simulación .....	57
Figura 51: Disposición de los elementos en el sistema.....	58
Figura 52: Distribución del flujo Tonatiuh vs SolTrace .....	59
Figura 53: Ajuste de los datos obtenidos a sendas distribuciones normales.....	60
Figura 54: Distribución del flujo sobre un heliostato .....	62
Figura 55. Comparativa datos teóricos y datos reales.....	62
Figura 56: Error relativo en función de theta .....	63
Figura 57: Partición de una malla .....	63
Figura 58: Escenario alternativo .....	64
Figura 59: Errores relativos correspondiente a un escenario relativo .....	65
Figura 60: Forma solar (“User Defined Sunshape“) tras simulación.....	66
Figura 61: Comparativa misma forma solar y diferentes herramientas .....	66



## Prólogo

El Trabajo Fin de Máster (TFM) que hay detrás de estas páginas, es el resultado de cientos de horas de trabajo y esfuerzo en la Cátedra de Energías Renovables de la Universidad Pública de Navarra (UPNA), desarrollando el trabajo en su totalidad en el Centro Nacional de Energías Renovables (CENER).

Debido a ser un proyecto de larga duración (10 meses) el proyecto se engloba en una primera parte que consiste en el TFM y una segunda parte que es el complemento al TFM. A parte de esta memoria, existe un segundo trabajo que complementa a este titulado *“Impact of incident angle dependant optical properties and dirtiness on the optical performance of parabolic trough collectors”*. Para la comprensión del complemento se recomienda una breve lectura y comprensión de este primer trabajo.

Una vez orientado el contexto del proyecto al lector no quiero terminar sin agradecer la oportunidad que se me ha brindado en poder desarrollar los conocimientos adquiridos a lo largo de mi vida como estudiante e incluso ampliarlos en el mundo laboral a todas las personas implicadas tanto desde la Universidad como desde CENER.

## 1. Objetivo del proyecto

La finalidad del proyecto que se expone es la continuación en el desarrollo y mantenimiento del software Tonatiuh. El software tiene una gran cantidad de funcionalidades pero en concreto se enfoca en el desarrollo de nuevos modelos para la forma del sol y el perfil de irradiancia. Las tareas a realizar son:

- Búsqueda de bibliografía de los modelos del Sol y perfiles de irradiancia existentes.
- Familiarizarse con el entorno de desarrollo de Tonatiuh.
- Definición de las especificaciones para el desarrollo de las nuevas funcionalidades.
- Implementación del código en C++.
- Pruebas y validación.



## 2. Introducción

El sector de la energía termosolar abarca una gran cantidad de disciplinas entre los que están los ingenieros, economistas, abogados, periodistas, etc. Dentro del mundo ingenieril la temática existente es enorme abarcando desde sistemas de torre, lineales, Fresnel y de disco, pasando por análisis de ciclos de potencia, almacenamiento térmico, análisis del recurso solar, instrumentación, materiales avanzados, integración en la red hasta hibridación de sistemas. El proyecto se puede catalogar en la temática de herramientas computacionales para la simulación y análisis de plantas de concentración solar.

El proyecto se enfoca entorno a Tonatiuh. El software es una herramienta diseñada por CENER (Centro Nacional de Energías Renovables) que sirve para el análisis óptico-energético de una gran variedad de sistemas. En el apartado 5.1 se explica detalladamente.

Existe un gran interés en desarrollar herramientas que faciliten el estudio de los efectos ópticos que se producen tanto en el colector como en el receptor para conocer cómo se comporta ante la exposición al Sol. Por un lado facilita la mejora de elementos ya diseñados como la introducción de otros nuevos con unas cualidades que mejoren los rendimientos ópticos del sistema. Por otro lado, las herramientas computacionales facilitan el estudio de un escenario en concreto sin tener que hacerlo físicamente lo que supone un ahorro económico.

Como bien se ha especificado en el apartado de objetivos, uno de ello es implementar otras formas solares. El software tiene implementados dos de los muchos que existen de forma teórica. Son modelos más o menos básicos que no se ajustan a la realidad por lo que existe cierta necesidad en actualizar los modelos solares e introducir unos más precisos.

Con la correcta realización de este proyecto el beneficio potencial sea enorme. El programa es de código abierto y además se puede descargar desde Github (<https://github.com/iat-cener/tonatiuh>) de forma gratuita por lo que de forma directa e indirectamente el beneficiado es la sociedad.

### 2.1. Situación energética en el mundo

Los datos expuestos a continuación provienen de artículos y análisis de la situación energética en el mundo y en España hasta 2016.

Entre finales del siglo XX y principios del XXI se han creado diversos protocolos para la reducción de emisiones contaminantes entre los que cabe destacar el tratado de Kioto (1997) y el tratado de París (2015). En ellos se impone a todos los firmantes la

obligación de introducir tecnologías que conlleve la reducción de emisiones de gases contaminantes en el transporte, generación eléctrica, industria, etc.

En la parte de generación eléctrica las tecnologías son múltiples. Desde las primeras que se introdujeron a gran escala como son la hidráulica, eólica y fotovoltaica hasta otras que se han ido implantando en el mix energético mundial más adelante como geotérmica, tecnologías mediante concentradores solares o termosolar y biomasa. Luego existen otras en fase experimental como pueden ser aquella que capta la energía del mar como son la mareomotriz (movimiento de mareas) y la undimotriz (fuerza de las olas).

Según los últimos estudios, del total de la energía generada sólo el 24.5 % proviene de las energías renovables siendo mayoritariamente energía hidráulica seguido de eólica, fotovoltaica, biomasa y por último CSP (Concentrated Solar Power)[1].

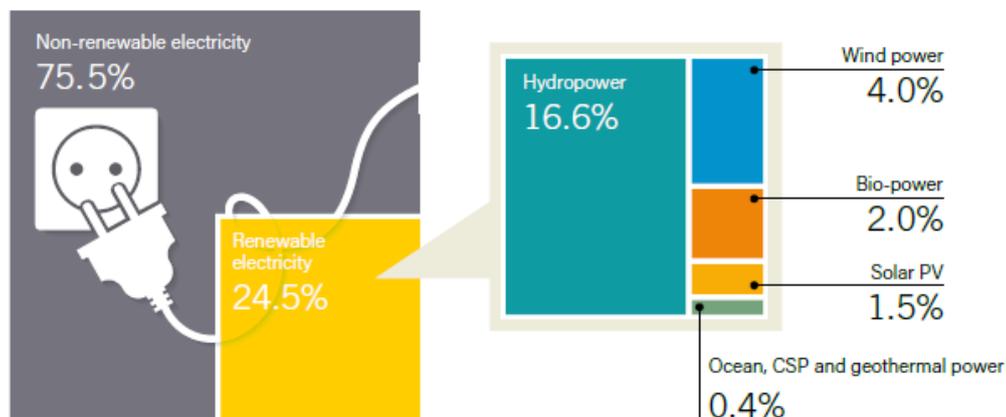


Figura 1: Producción global estimada de generación renovable

Respecto a la tecnología CSP, durante el año 2016 se pusieron en marcha 110MW más, lo que hace un total de 4.8 GW a final de año. Esto supone el incremento más bajo de los últimos años (2%) [2]. Durante el 2017 se prevé un crecimiento muy fuerte con la puesta en marcha de 900 MW. España sigue siendo el primer país mundial en potencia instalada (2.3 GW) de esta tecnología. El mercado está empezando a diversificarse y se aleja del tradicional mercado Español y Estadounidense (80% de la potencia instalada). Países como Marruecos, China, Arabia Saudí o Sudáfrica están entrando fuertemente en el mercado, incluso la mayor parte de las plantas están incorporando sistemas TES (Thermal Energy Storage).

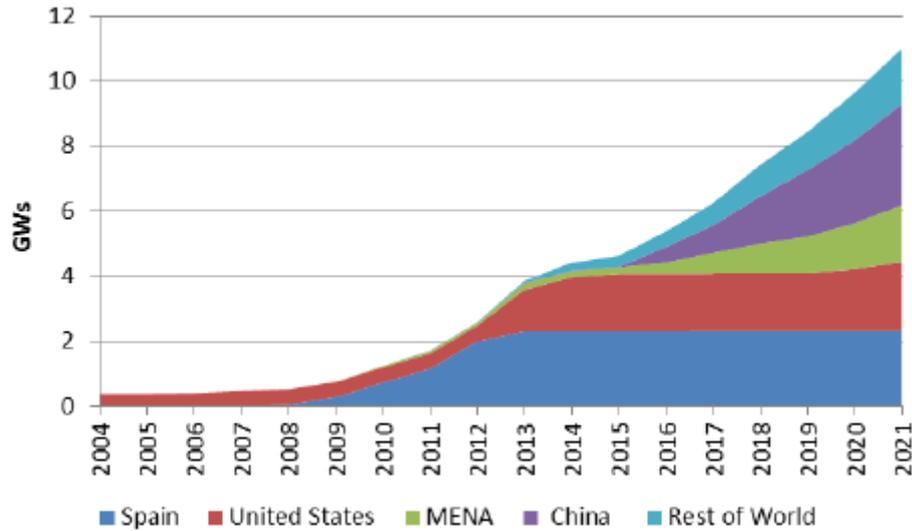


Figura 2: Potencia instalada de CSP actual y prevista (Fuente: SolarPACES)

Entre los diferentes tipos de concentradores, los reflectores parabólicos y la tecnología de torre son los dominantes en el mercado. Las tecnologías Fresnel y disco parabólico todavía tienen su importancia pero se utilizan en plantas de pequeña potencia que están en desarrollo y fase de construcción.

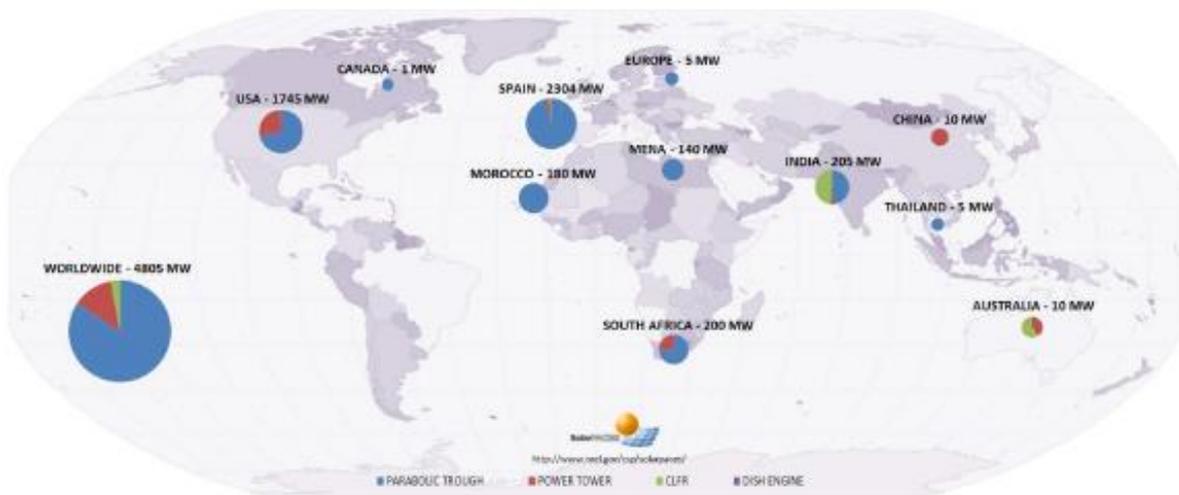


Figura 3: Potencia instalada en cada país de cada tipo de tecnología hasta 2016 (Fuente: SolarPACES)

Abengoa (España) ha sido el gran desarrollador y constructor de instalaciones CSP pero durante el 2016 tuvo problemas de solvencia económica que supuso una reestructuración de la propiedad. A parte de eso, el 2016 fue un año relativamente tranquilo para este sector. Entre las empresas más significativas que se dedican a la construcción, fabricación u operación de plantas CSP se encuentran Rioglass Solar (Bélgica); Supcon (China); Acciona, ACS Cobra, Sener y TSK (todas en España); y Brightsource, GE y Reserva solar (todas en EEUU).



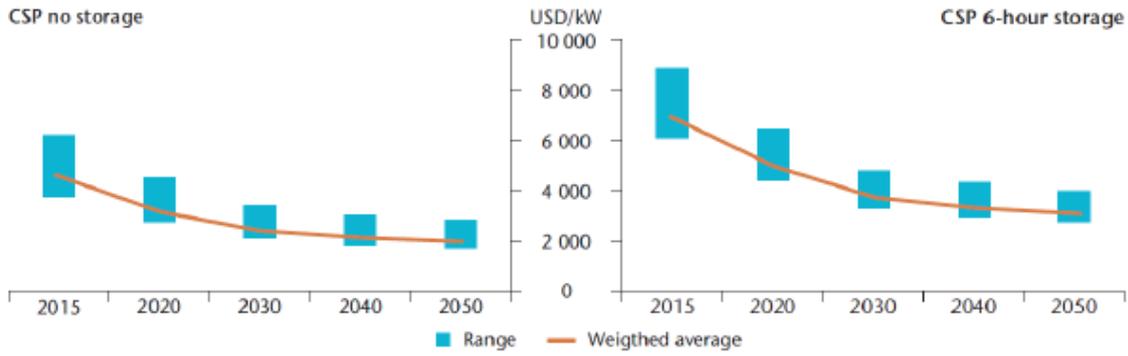
Aunque el coste de generación de energía con CSP ha disminuido significativamente, la implementación de nuevas plantas se ve obstaculizada por la rápida reducción de los precios de la fotovoltaica y eólica. Por ello, el punto fuerte de la tecnología CSP está en el desarrollo y mejora de los sistemas TES que facilitan la gestionabilidad del recurso solar durante la noche, incluso durante el día pudiendo entregar mayor potencia si la demanda es mayor. Un punto a su favor ya que la fotovoltaica necesitaría de sistema de almacenamiento mediante baterías lo que actualmente elevaría los costos y no sería tan rentable su implementación.

Las instalaciones más grandes con tecnología TES se han seguido desarrollando con la capacidad de seguir produciendo energía al atardecer e incluso a la noche. La siguiente figura indica la reducción de los costes de construcción en proyectos para los próximos 5 años.

Project	Country	MWe	Storage Capacity (hours)	Power Purchase Agreement Price (PPA) (¢/kWh)	Status	Completion Date
Crescent Dunes	U.S.	110	10	13.7	Operation	Q4 2015
Noor III	Morocco	150	7.5	16.3	Construction	Q4 2017
Redstone	South Africa	100	12	12.5 (PPA to be signed)	Development	Q3 2018
DEWA CSP Project Phase I	United Arab Emirates	200	12	8.0 (targeted)	Planning	Q2 2021
Copiapo	Chile	240	14	6.3 (bid)	Planning	TBA

Figura 4: Proyectos de CSP de torre con almacenamiento ( Fuente: CSP Today Global Tracker)

Aunque es difícil de comparar los costes de cada planta a través del LCOE (Levelized cost of Energy) se puede hacer una aproximación de los costes para las próximas décadas.



### LCoE for plants including storage

USD/MWh	2015	2020	2025	2030	2035	2040	2045	2050
Minimum	146	116	96	86	72	69	66	64
Average	168	130	109	98	80	77	72	71
Maximum	213	169	124	112	105	101	96	94

Figura 5: Perspectiva del coste medio de las plantas CSP (Fuente: Ren21)

Como dato curioso decir que se estima que con un 1% de la superficie desértica del planeta sería suficiente para cubrir la demanda eléctrica mundial con sistemas de concentración solar térmica. En España se mantendría dicha proporción.

## 2.2. Energías renovables

Durante millones de años, el clima de la Tierra se ha mantenido a una temperatura media relativamente estable, lo que ha permitido el desarrollo de la vida. Los gases de efecto invernadero han conservado su equilibrio gracias, fundamentalmente, a la acción de la lluvia y de la vegetación, que regulan las cantidades de dióxido de carbono en la atmósfera.

Sin embargo, en los últimos 50 años, las concentraciones de gases de efecto invernadero están creciendo rápidamente como consecuencia de la acción humana. El uso generalizado de los combustibles fósiles, el debilitamiento de la capa de ozono y la destrucción de las masas forestales están favoreciendo el aumento de la temperatura de la Tierra, provocando cambios drásticos en el clima mundial y haciéndolo cada vez más impredecible.

A medida que una sociedad está más desarrollada consume más energía. Pero la energía que se obtiene del carbón, del petróleo y del gas no se renueva y se va agotando año tras año además de aumentar las emisiones de gases de efecto invernadero. Es necesario buscar otras fuentes de energía que sean más limpias e ilimitadas.

La energía proveniente del viento, del sol, residuos, etc. las cuales son renovables, no se agotan y además no contaminan el ambiente, lo que significa una doble ventaja

para los ciudadanos. Hoy en día constituyen un complemento a las energías convencionales fósiles (carbón, petróleo, gas natural, etc.) En la siguiente tabla se muestran las diferencias entre las energías renovables y no convencionales.

	E.Renovables	E.Convencionales
<b>Ventajas</b>	Son limpias	Contaminan
	Sin residuos	Generan emisiones y residuos
	Inagotables	Son limitadas
	Autóctonas	Provocan dependencia del exterior
<b>Inconvenientes</b>	No gestionable	Gestionable

### 2.3. Energía solar térmica y de concentración

La energía solar térmica consiste en el aprovechamiento de la energía procedente del Sol para transferirla a un medio portador de calor. Generalmente se clasifican los colectores en función del rango de temperaturas para la cual se calienta el fluido: bajas temperaturas, medias temperaturas y altas temperaturas.

#### 2.3.1. Solar térmica de baja y media temperatura.

Son instalaciones relativamente sencillas que aprovechan la radiación solar de una forma poco eficaz pero que se utilizan en zonas residenciales con un coste bajo. El aprovechamiento de energía es solo en forma de calor. A continuación se muestra el funcionamiento de estas instalaciones:

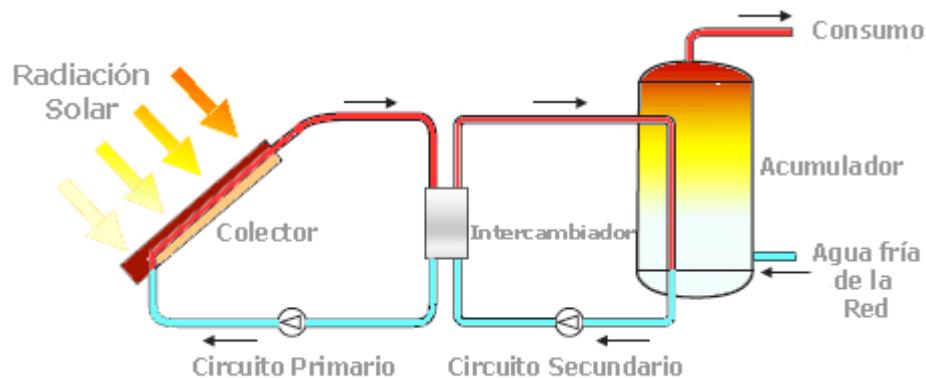


Figura 6: Esquema básico de una instalación de energía solar térmica



Un colector solar es un dispositivo diseñado para recoger la energía radiada por el sol y convertirla en energía térmica. Los colectores de baja temperatura fundamentalmente se utilizan en sistemas domésticos de calefacción y ACS (agua caliente sanitaria)[3].

Estos colectores consisten en una caja plana metálica que aloja los dispositivos necesarios para la circulación de un fluido, que se calienta a su paso por el panel debido al efecto invernadero. La luz visible calienta la placa colectora que, a su vez, se convierte en emisora de radiación de onda larga. El vidrio actúa como filtro para ciertas longitudes de onda de la luz solar: deja pasar fundamentalmente la luz visible, y es menos transparente a las ondas infrarrojas de menor energía, por lo que, a pesar de las pérdidas por transmisión (el vidrio es mal conductor térmico) el recinto de la caja se calienta por encima de la temperatura exterior.

A la salida del colector, el fluido caliente pasa por un intercambiador de calor que transfiere el calor a un flujo de agua fría, el cual se calienta y se almacena en un acumulador para su posterior consumo.

Los colectores de media temperatura captan mayor cantidad de energía solar. El funcionamiento es prácticamente igual que los colectores de baja temperatura solo que el área de captación es mayor. Entre sus aplicaciones está el calentar agua para edificios residenciales y comerciales. También se utiliza en aplicaciones industriales, sobre todo el alimenticio en las que se necesitan calor para la limpieza y secado de alimentos o secado de productos en cualquier sector industrial.

### 2.3.2. Energía solar termoelectrica o de concentración

- *Breve historia*

Dentro de las tecnologías que utilizan el sol como recurso principal para la obtención de energía eléctrica están los sistemas de captación solar mediante concentración. Por ello, el Sol es una de las fuentes de energía que más peso está cogiendo en los últimos años, de ahí la importancia de comprender el comportamiento que tiene y comprender qué fenómenos externos afectan desde que sale la energía del sol hasta que llega al suelo terrestre.



Figura 7: Sistema de espejos diseñado por Arquímedes para quemar las velas de los romanos

Antes de todo es de gran interés conocer la evolución histórica de la energía solar térmica de concentración. Se tiene constancia que los griegos (siglo III a.C) fueron los primeros en concentrar los rayos del sol para poder encender las antorchas para sus rituales religiosos mediante la *Skaphia*, un elemento parabólico que concentraba los rayos del sol. Incluso unos años después (213-211 a.C) se utilizó esta técnica de concentración para usos militares como fue la famosa batalla de Siracusa en la que Arquímedes consiguió un sistema de espejos que concentraban los rayos de forma adecuada para quemar las velas de los barcos romanos (Figura 7).

El famoso Leonardo Da Vinci, el gran hombre del renacimiento, también mostró su interés en las propiedades de la concentración solar. En 1515 emprendió un gran proyecto por el cual quería producir vapor industrial. Solo se tiene constancia en las notas que él mismo dejó.

Uno de los primeros grandes pasos en lo que hoy conocemos como concentradores solares fue la invención del sueco Charles Wilson (1872). La instalación consistía en una agrupación de arcas poco profundas llenas de agua salada, cubiertas por un techo oblicuo de vidrio (Figura 8). El agua era elevada desde los pozos hasta un estanque, mediante una bomba accionada por una turbina eólica. El sol calentaba el agua de las arcas por los rayos que pasaban a través de los vidrios, de modo que se producía vapor, el cual era condensado y conducido hasta un tanque de almacenamiento [4].



Figura 8: Primera planta desaladora solar del mundo, diseñada por Charles Wilson (Chile, 1872)

Las primeras instalaciones prototipo de generación eléctrica mediante la concentración solar eran colectores solares. Tiene gran relevancia el prototipo creado por el ingeniero Frank Shuman quién puso en marcha tras prueba y error y después de convencer a los inversores una planta en 1912 en el Sahara de 52.600 km<sup>2</sup> de colectores solares para producir 198 MW, lo que hubiera sido el equivalente al consumo de energía mundial en 1908.



Figura 9: El experimento de Shuman

Durante los años 60, el abaratamiento de los combustibles fósiles hizo que se dedicase poca atención al tema de la energía solar pero fue en 1973 cuando, como consecuencia de la cuarta guerra árabe-israelí, la organización de Países Exportadores de Petróleo (OPEP) decidió elevar enormemente los precios del petróleo y volvió el interés hacia la energía solar. Fue entonces cuando distintos gobiernos establecieron grupos de trabajo para investigar en energías solares de concentración y desarrollar tecnología que permitiese generar electricidad en grandes cantidades y a precios competitivos. Uno de los proyectos más importantes y que todavía sigue en



funcionamiento es la denominada Plataforma Solar de Almería (PSA), donde participaron inicialmente Alemania, Austria, Bélgica, Estados Unidos, Grecia, Italia, Suiza y Suecia. A medida que los precios del petróleo se fueron deshinchando, distintos países se retiraron del proyecto. Hoy en día el proyecto depende del Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT) y se ha considerado en los últimos 30 años como el principal referente de esta tecnología a nivel mundial.



Figura 10: Planta Solar de Almería (PSA)

El 30 de Marzo de 2007 se inauguró en España la primera planta con tecnología de torre que produce la electricidad de una manera estable y comercial también conocida como la PS10. La central solar tiene una potencia de 10 MW y es capaz de abastecer 5.500 hogares y ahorrar 6.700 toneladas de CO<sub>2</sub>. Dos años después se construyó la segunda planta en el mundo con este sistema tecnológico con una potencia de 20 MW anexa a la primera. Ambas fueron creadas por el grupo Abengoa Solar.



Figura 11: PS10 y PS20

- *Tecnologías actuales de concentración solar*

Los sistemas solares termoelectricos se pueden clasificar atendiendo a diferentes criterios según sea el tipo de concentración, la estructura del receptor, o la movilidad del receptor.

*Clasificación en función del tipo de concentración.*

**Concentración lineal:** es aquella que permite un seguimiento solar en un solo eje. Es más fácil de instalar al tener menos grados de libertad, pero tiene un factor de concentración menor por lo tanto puede alcanzar menores temperaturas que la tecnología de concentración puntual. Las tecnologías basadas en la concentración lineal son la Cilindro-Parabólica (CCP) y la Fresnel.

**Concentración puntual:** es aquella cuya concentración de la radiación solar se realiza sobre una superficie puntual puesto que en toda ella la temperatura es prácticamente uniforme. El seguimiento solar es en dos ejes. En estas tecnologías, la concentración que se alcanza es mayor, siendo estas: Torre central y Discos parabólicos.

- *Clasificación en función de la movilidad del foco receptor.*

También se pueden clasificar en función de la movilidad del foco receptor de la radiación concentrada.

**Receptor fijo:** para realizar el seguimiento solar, únicamente se mueven los espejos reflectores, mientras que el foco receptor permanece fijo. La principal ventaja de este sistema es que es más fácil la circulación del fluido calorportador, al no tener movimiento el absorbedor.

**Receptor móvil:** en este caso, tanto el reflector como el receptor se mueven para realizar el seguimiento solar. Por ello, la instalación para la circulación del fluido calorportador por el receptor es más compleja, en el caso de las CCP.

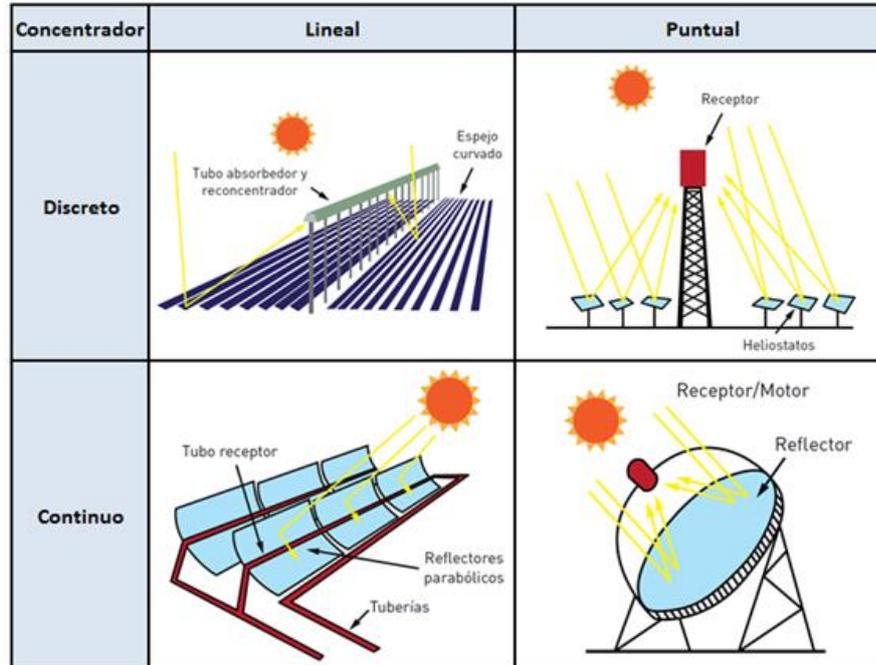


Figura 12: Clasificación tipo de concentradores

- *Planta de potencia de una central termosolar*

Tras concentrar los rayos del sol y calentar un fluido que puede ser agua, aceite sintético o sales fundidas las siguientes etapas que sigue serían como las de una central convencional. Se utiliza un ciclo Rankine; primero se calienta el agua hasta que se transforma en vapor sobrecalentado a alta presión. Se turbinan con la ayuda de múltiples etapas y a la salida se extrae el calor del agua para poder cerrar el ciclo. Por lo tanto la única novedad de estos sistemas es la fuente primaria con la que más adelante se genera la electricidad.

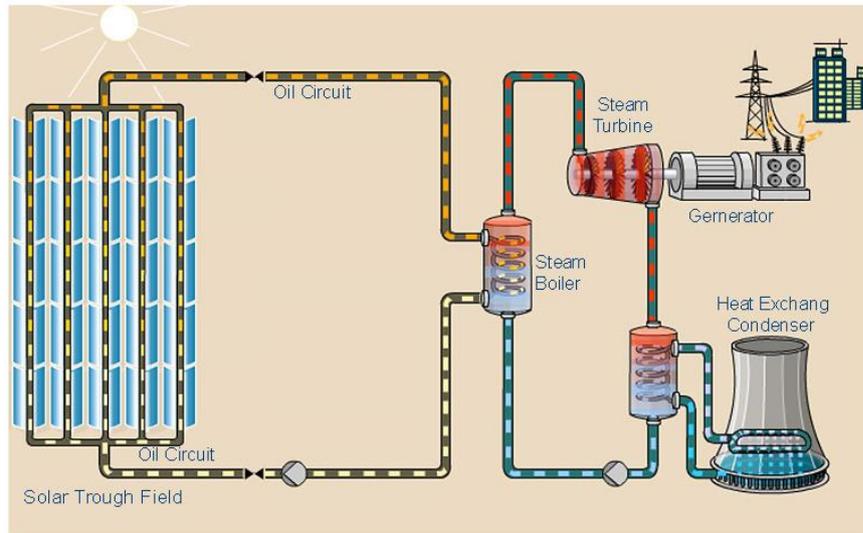


Figura 13: Esquema general de una planta solar termoeléctrica

A las planta de potencia se le puede introducir un sistema de almacenamiento de energía térmica conocido en sus siglas en inglés como TES. Son más caros pero permite prolongar el tiempo de utilización del bloque de potencia, aumentar la eficiencia y además aumenta el ciclo de vida de los componentes debido a la reducción de los transitorios térmicos.

La capacidad de almacenamiento de energía de la planta, depende directamente de la capacidad de producción de energía de la misma, es decir, un aumento del número de horas de almacenamiento térmico requerirá un mayor campo solar. Los sistemas de almacenamiento térmico pueden dividirse en sistemas activos y sistemas pasivos.

En los sistemas activos el medio de almacenamiento de calor es un fluido circulante que almacena y proporciona calor sensible. Este fluido es almacenado a temperaturas elevadas, en unos tanques aislados térmicamente.

En los sistemas de almacenamiento activo directo, parte del fluido de transferencia, que proviene del campo solar es directamente bombeado a unos tanques de almacenamiento que están altamente aislados. Es decir, el fluido de transferencia es también el medio de almacenamiento.

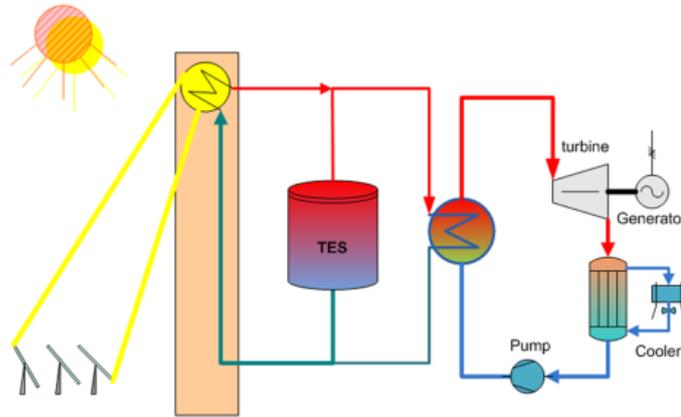


Figura 14: Esquema de una planta con sistema de almacenamiento activo directo

En los sistemas de almacenamiento indirecto el medio de almacenamiento térmico y el fluido de transferencia (calorportador) son diferentes. El calor obtenido en el fluido calorportador se transfiere al sistema de almacenamiento con la ayuda de un intercambiador de calor.

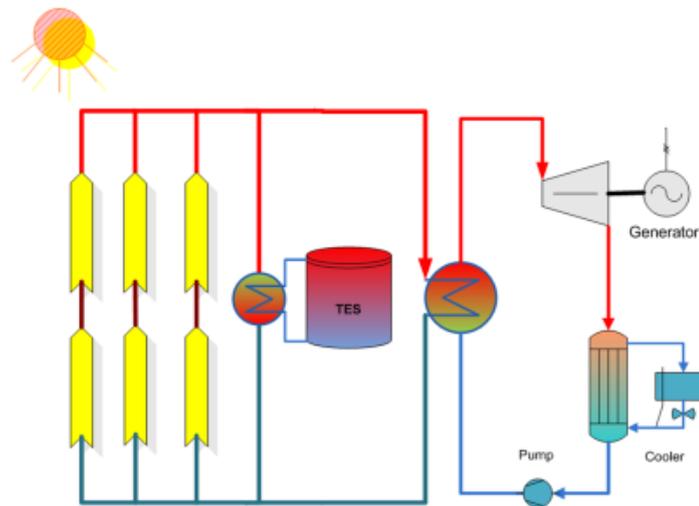


Figura 15: Esquema de una planta con sistema de almacenamiento activo indirecto

Los sistemas de almacenamiento pasivo de calor sensible emplean medios de almacenamiento sólidos (Hormigón o cerámica), a donde se bombea el fluido de transferencia para transferir la energía térmica o viceversa.

### 2.3.3. Otras aplicaciones de las tecnologías de concentración solar

A parte de la utilización de las tecnologías de concentración solar en aplicaciones de generación eléctrica existes una gran cantidad de aplicaciones industriales y en la propia sociedad donde se pueden emplear.



En muchas industrias se requieren grandes cantidades de calor para llevar a cabo muchas de sus aplicaciones y procesos, tales como la esterilización, calefacción, sistemas de refrigeración mediante máquinas de absorción, limpieza, secado, evaporización y destilación, escaldado, pasteurización, fusión, pintura y tratamiento de superficies. Un estudio encargado por la Agencia Internacional de la Energía en 2008 determinó, que para varios sectores industriales, como el de la alimentación, equipos de transporte, maquinaria industrial, textil, papelería, siderurgia, plásticos y químicos, el 27 % de sus necesidades de calor son de temperaturas medias (100°C a 400°C) y el 43% de altas temperaturas (por encima de 400°C).

Los colectores cilíndrico-parabólicos y sistemas lineales Fresnel son los más adecuados para la captura de calor para estos procesos industriales. Podrían ser considerados como una opción económica para instalar in-situ en una amplia gama de tipos de industrias, que requieren calores medios y elevados. El uso de torres de concentración o discos parabólicos para procesos a altas temperaturas, como la requerida en la industria de la cerámica está bajo investigación.

Los lugares con grandes cantidades de radiación solar son a menudo también lugares con problemas de abastecimiento de agua. Según un estudio de 2007 del Centro Aeroespacial Alemán (DLR), la tecnología solar de concentración para la desalinización del agua de mar tiene un gran potencial para proveer de agua a los grandes centros urbanos de los países mediterráneos, Oriente Medio y norte de África. El estudio encontró que el recurso solar en la región es más que suficiente para proporcionar energía para la desalación que satisfaga el creciente déficit de agua de estas áreas.

Otro punto de especial atención se centra en el hidrógeno (H<sub>2</sub>), una potencial alternativa a los combustibles fósiles, especialmente adecuada para el transporte. Hoy en día más del 90 % del hidrógeno se produce utilizando calor procedente de la combustión de combustibles fósiles, principalmente gas natural. Si el hidrógeno se generase a partir de energía solar, sería una tecnología completamente limpia, sin residuos peligrosos ni emisiones de gases contaminantes.

### **3. Recurso solar**

#### **3.1. El Sol**

Como el propósito del proyecto en cuestión es el desarrollo de modelos del sol para implementar en el *software* Tonatiuh es de gran importancia tener unos conocimientos básicos del comportamiento y emisión de la energía liberada hacia el espacio.

Es bien sabido que el sol es una esfera gaseosa con un diámetro de  $1.39 \times 10^9$  y que está a una distancia media de  $1.5 \times 10^{11}$  m respecto de la tierra. La distancia varía a lo largo del año porque la tierra describe una elipse alrededor del sol. La estrella se

comporta como un cuerpo negro a una temperatura de 5777 °K aunque la temperatura en su interior varía según estimaciones entre  $8 \times 10^6$  hasta  $40 \times 10^6$  K [5]. Al final el sol es un reactor de fusión continuo en el que la reacción principal es hidrógeno y helio que libera una energía ingente que se irradia hacia el espacio.

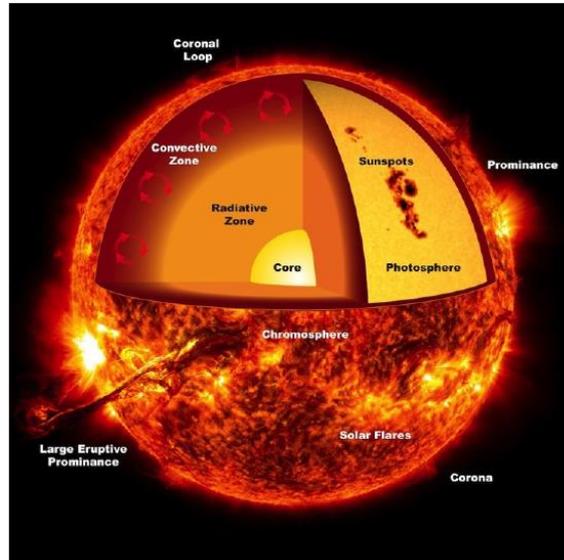


Figura 16: Estructura del Sol

Una vez que la energía solar sale del cuerpo, se transmite a lo largo del espacio y llega hasta la tierra. La distancia entre los dos cuerpos es de  $1.5 \times 10^{11} \text{ m} \pm 1.7 \%$  [6]. La constante solar  $G$ , es la energía del sol por unidad de tiempo que llega perpendicular a una unidad de área en la capa exterior de la atmósfera. Antes de la existencia de vuelos espaciales las mediciones de irradiancia se hacía desde la superficie terrestre. Lógicamente los resultado dependían de las condiciones atmosféricas por la dispersión de los rayos del sol con los diferentes compuestos que hay suspendidos en la atmósfera. Más adelante se utilizaron globos, aviones o mediciones en alta montaña para poder aproximarse a los valores de irradiancia que llegaban desde el espacio a la termosfera (hasta 690 km respecto de la superficie terrestre). Actualmente “The World Radiaton Center” ha adoptado un valor de  $1367 \text{ W/m}^2$  no obstante este valor varía y no siempre es constante.

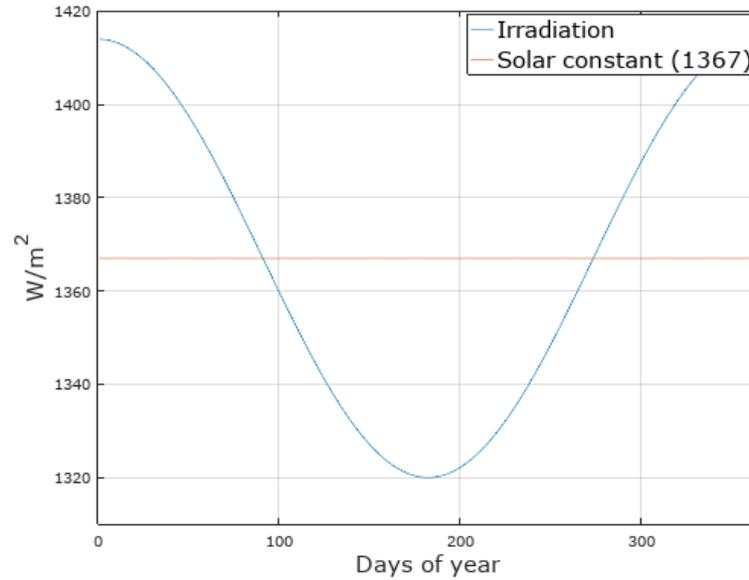


Figura 17: Variación de la radiación extraterrestre

### 3.2. La atmósfera y su influencia en la radiación solar

La atmósfera es al final una envoltura de gases que rodean la tierra, compuesta principalmente por O<sub>2</sub>, N<sub>2</sub>, CO<sub>2</sub>, agua y pequeñas partículas de polvo y sólidos suspendidos. Cuando la radiación solar alcanza la atmósfera, una parte de ella es absorbida y otra dispersada. El grado en que la radiación es dispersada depende de la transmisividad de los gases y sólidos que conforman la capa gaseosa.

La dispersión de la radiación solar (conocido como scattering), cuando pasa a través de la atmósfera, es causada por la interacción de los rayos con las moléculas de aire, el agua (en forma de vapor y pequeñas gotas) y el polvo. El grado con el cual se produce la dispersión es función del número de partículas a través de las que debe pasar la radiación y del tamaño de las partículas relativo a la longitud de onda. La distancia que debe recorrer la luz del sol a través de las moléculas de aire esta descrita por la masa de aire[7].

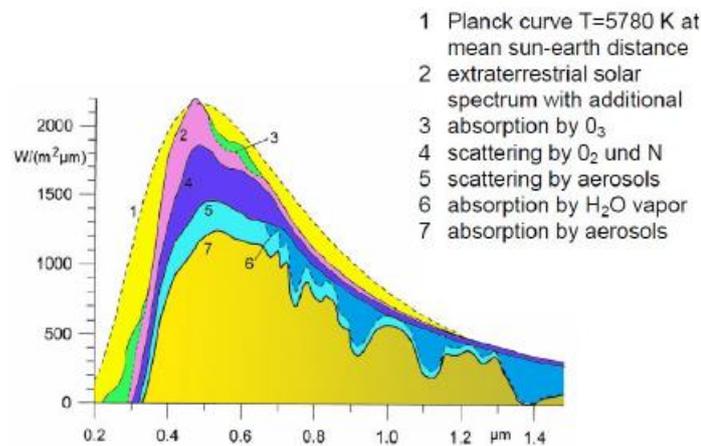


Figura 18: Espectro solar y la influencia de la atmósfera

Los efectos de la atmósfera en la dispersión y la absorción de la radiación son variables con respecto al tiempo, así como con las condiciones atmosféricas y los cambios de la masa gaseosa. Es útil definir un día claro aquel en que la dispersión es baja. Existen 3 componentes de interés de la radiación solar:

- **Irradiancia directa:** es la cantidad de radiación solar recibida por unidad de área por una superficie que siempre será perpendicular a los rayos que proceden en línea recta con el Sol.
- **Irradiancia difusa:** es la cantidad de radiación solar recibida por unidad de área que no llega directamente desde el Sol pero que ha sido dispersada por moléculas y partículas en la atmósfera.
- **Irradiancia global:** es la cantidad total de radiación que recibe por unidad de área y es la suma de radiación directa, difusa y reflejada.

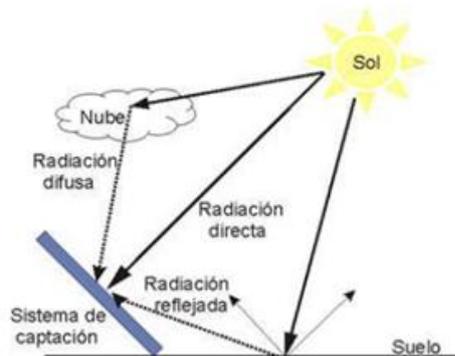


Figura 19: Componentes de la radiación solar terrestre

La posición del Sol depende de la localización de un punto en la Tierra, su latitud y longitud que varía a cualquier hora del día y cada día del año.

- **Ángulo horario  $\omega$ :** describe la hora solar en términos trigonométricos como un ángulo. Una hora corresponde a un ángulo de  $360^{\circ}/24h = 15^{\circ}$ .
- **Latitud  $\Phi$ :** Es el ángulo que localiza si estamos al norte o al sur del ecuador;  $-90^{\circ} < \Phi < 90^{\circ}$ .
- **Declinación  $\delta$ :** Ángulo comprendido entre el rayo del sol y el plano del ecuador;  $-23.45^{\circ} < \delta < 23.45^{\circ}$ .
- **Inclinación  $\alpha_p$ :** Ángulo entre el plano de la superficie en cuestión y la horizontal;  $0^{\circ} < \alpha_p < 90^{\circ}$

- **Ángulo azimutal  $\alpha_s$** : Es la desviación de la proyección en un plano horizontal de la superficie desde el meridiano local. Donde el cero corresponde con el Norte.
- **Elevación o altitud  $\gamma_s$** : Ángulo vertical comprendido entre la superficie terrestre y el Sol
- **Ángulo de incidencia  $\Theta$** : Ángulo comprendido entre el Sol y la normal de la superficie.

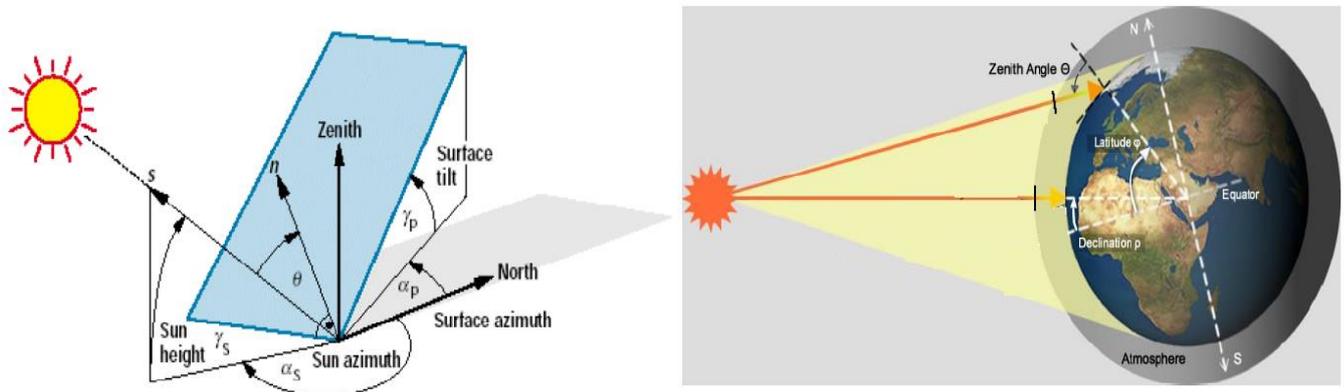


Figura 20: Referencias de la posición solar

### 3.3. Instrumentos de medición

Para conocer más a fondo el comportamiento del Sol se utilizan diferentes instrumentos que evalúan los diversos componentes definidos en el apartado 3.2.

El instrumento de medición más comúnmente utilizado para medir la radiación directa es el pirheliómetro. Este instrumento emplea un detector colimado para medir la radiación solar que proviene directa del sol y de una pequeña porción del cielo alrededor del sol con una incidencia normal.

Los pirheliómetros estándares no son fáciles de usar, por lo que se han establecido instrumentos estándar secundarios para calibrar los instrumentos de campo. Los instrumentos operacionales o de campo están calibrados contra estándares secundarios y son la fuente de la mayoría de los datos sobre los cuales deben estar basados los diseños de ingeniería de procesos solares. El más utilizado en Europa es el Kipp & Zonen [8].

El actinómetro Kipp & Zonen está basado en el diseño de Linke-Feussner y utiliza una termopila de constatan-manganina de 40 uniones, con las uniones calientes calentadas por la radiación y las uniones frías en buen contacto térmico con el cuerpo del instrumento. En estos equipos el ensamble de diafragmas y cuerpo de cobre tienen una capacidad térmica muy grande, de órdenes de magnitud mayores que las uniones calientes. Con la exposición a la radiación solar, las uniones calientes alcanzan

rápidamente temperaturas mayores que las uniones frías; la diferencia en las temperaturas provee una medición de la radiación (efecto peltier).

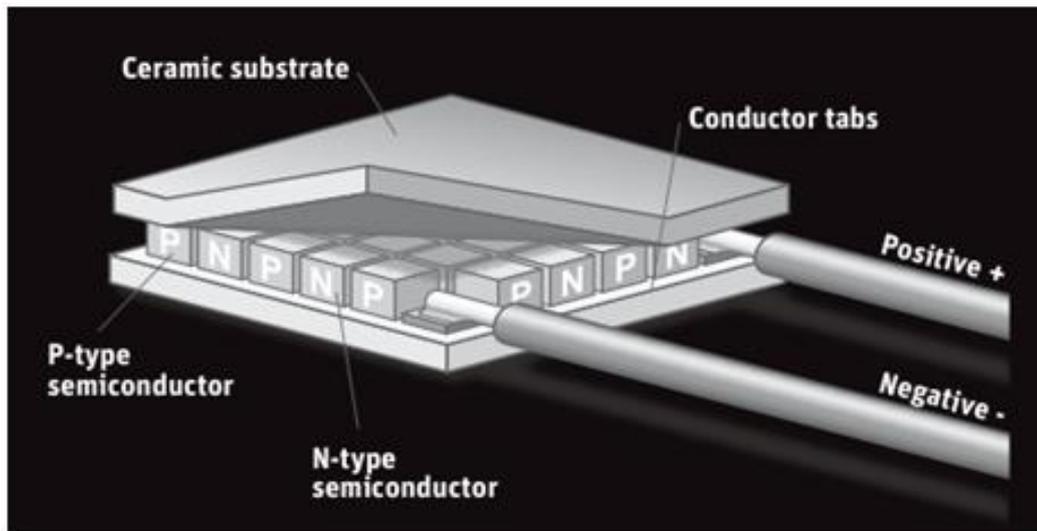


Figura 21: Principio de funcionamiento de los pirheliómetros Kipp & Zonen

Las mediciones de radiación difusa pueden ser realizadas con Piranómetros comunes sombreándolos de la radiación directa. Esto es usualmente hecho por medio de un anillo sombreador, que permite el registro de constantes de radiación difusa sin la necesidad de posicionamiento continuo, sólo se necesita hacer ajustes por la declinación pero esto se hace cada cierto tiempo [9].



Figura 22: Piranómetro con banda de sombra para medir la radiación

Las mediciones de radiación global se hacen con los mismos Piranómetros pero sin añadirle el anillo sombreador, obtienen la mayoría de los datos de radiación disponible.

## 4. Estado del arte en modelos solares

El perfil de distribución o forma solar es el concepto que recoge las variaciones en la distribución radial de energía que presenta el Sol debido a que se considera como una fuente luminosa no puntual. Diversos autores han estudiado este fenómeno, y han propuesto expresiones que se ajustan con mayor o menor precisión al perfil de emisión de energía del disco solar.

La relación geométrica entre el radio del Sol y su distancia media a la Tierra hace que este sea visto como un disco cuyo diámetro se subtende bajo un ángulo de 9.3 mrad. Esto implica, que a cada punto de la Tierra llega un rango de rayos procedentes de cada uno de los puntos de dicho disco, formando un cono solar con una apertura definida.

### 4.1. Pillbox

En una primera aproximación, el Sol, como esfera uniforme emisora de energía a temperatura equivalente de 5777 K, se configura como un cuerpo lambertiano<sup>1</sup>, y así, un observador lejano lo percibiría como un disco perfecto de brillo constante en toda su superficie [10]. Esto daría lugar a obtener una distribución energética angular constante también llamada Pillbox. La distribución es independiente de las condiciones atmosféricas.

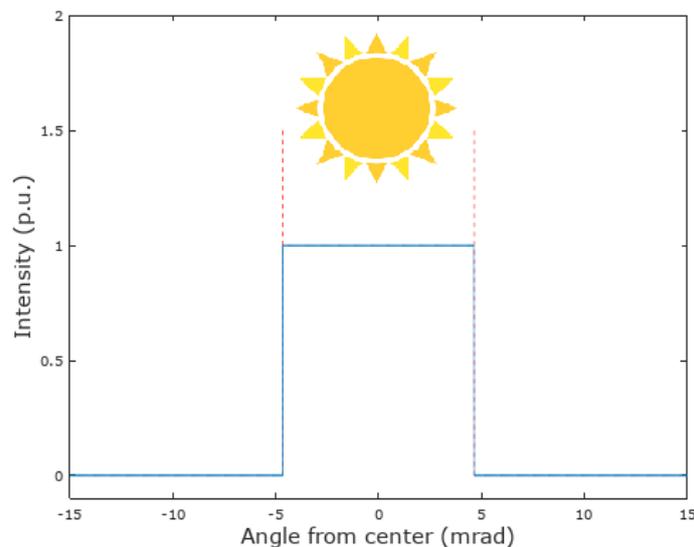


Figura 23: Perfil de la superficie Pillbox

<sup>1</sup> En física, una superficie lambertiana es una superficie ideal que refleja la energía incidente desde una dirección igual en todas direcciones, por lo cual al variar el punto de vista, su luminancia no cambia.



El modelo Pillbox es válido para concentradores con un ratio de concentración pequeño, para concentraciones mayores (>10000) se necesitan perfiles de la superficie solar mejor definidas[11].

## 4.2. Buie

Otro de los modelos ya implementados es la superficie solar Buie. Para la implementación de este modelo se tiene en cuenta factores atmosféricos.

Como ya se sabe los rayos del sol interactúan con las partículas atmosféricas [5]. Cuando el radio de las partículas es grande (entre 0.1  $\mu\text{m}$  hasta 1  $\mu\text{m}$ ) en comparación con la longitud de onda del rayo entonces se producen pequeñas desviaciones del ángulo de incidencia formando la aureola solar llamado a partir de ahora circumsolar o corona solar. El total de la energía en la corona solar es importante por dos motivos. La primera, depende del ángulo de inclinación de un sistema de concentración solar (CST) haciendo que se pueda sobreestimar la potencia extraída si se asume que toda la energía proviene del disco solar[9]. Lo segundo es que el perfil de energía radial (sunshape) puede desempeñar un papel nada despreciable en la determinación de la distribución general del flujo en el plano focal de los sistemas de concentración.

Las grandes partículas en la troposfera son creadas generalmente por la influencia del viento y la actividad del agua y además por la contribución de las partículas de los aerosoles, la gran mayoría de las grandes industrias. Estos sistemas crean partículas solubles y otras insolubles. Las partículas solubles suelen aumentar de tamaño por la humedad y la temperatura del ambiente. La cantidad de grandes partículas disminuye de forma exponencial cuanto mayor es la altitud. Esto junto a la irregularidad de la sunshape hace que la dispersión del rayo este fuertemente influenciado con las condiciones atmosféricas.

En 1992, Neumann y Schubnell con la ayuda de una CCD (Charge coupled device camera) cuantificaron el grado en que una variación en la sunshape podía afectar a la eficiencia óptica de los concentradores. Al simular los datos obtenidos y comparando con el modelo Pill-box en diferentes dispositivos de concentración solar con la ayuda de un trazador de rayos obtuvieron una hasta una variación del 20 % en el rendimiento óptico [12].

El ratio de la corona solar (CSR) se define como la relación entre la radiación en la corona solar ( $\Phi_{cs}$ ) y la radiación total incidente ( $\Phi_i$ ):

$$CSR = \frac{\Phi_{cs}}{\Phi_i} \quad \text{Ec. 1}$$

Si se asume que el perfil de intensidad solar  $\Phi(\theta)$  tiene una simetría radial, algo que se asumirá en los modelos a implementar. Tanto la radiación en el disco solar como el total se pueden calcular usando la siguiente expresión:

$$\Phi_{cs,i} = 2 \pi \int_{0, \theta_{disk}}^{\theta_{out}} \Phi(\theta) * \sin(\theta) d \theta \approx 2 \pi \int_{0, \theta_{disk}}^{\theta_{out}} \Phi(\theta) \theta d \theta \quad \text{Ec. 2}$$

Siendo  $\theta$  una pequeña variación del ángulo radial y  $\theta_{disk}$  y  $\theta_{out}$  son los límites radiales del disco solar y la región de la corona solar.

El valor del CSR no permanece constante sino que varía a lo largo del año debido a que el disco solar no permanece constante pero para obtener un modelo genérico del sol, es esencial tomar un único valor para el borde del disco solar y por lo tanto se introducirá un pequeño error en los cálculos. El valor aceptado para  $\theta_{disk}$  es una variación respecto del centro del sol de 4.65 mrad.

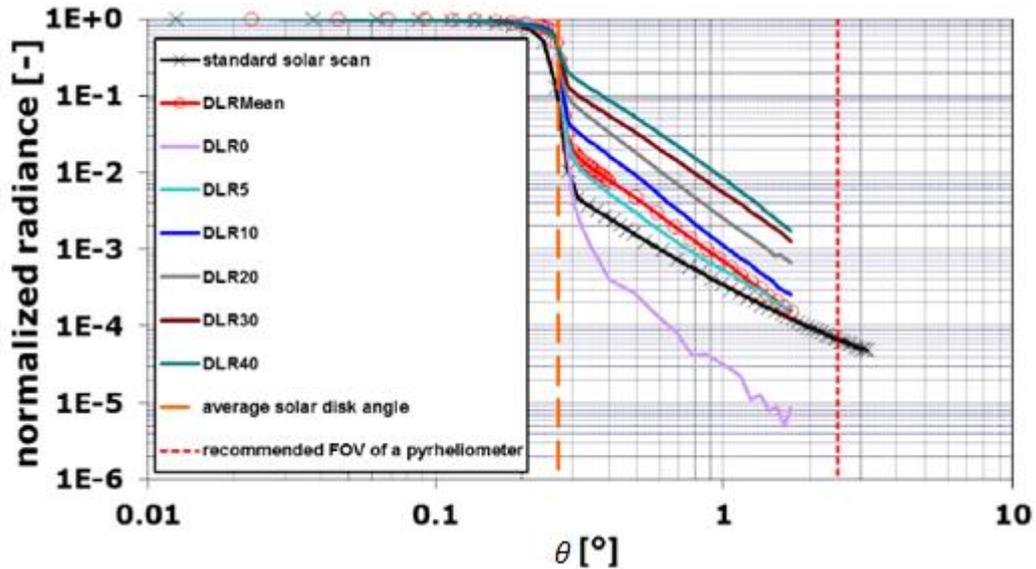


Figura 24: Perfil de la superficie Buie

La distribución de la energía del disco solar depende del oscurecimiento del exterior del astro (efecto óptico) y la atenuación atmosférica. El modelo establecido por Neumann representa el perfil terrestre del disco solar de la forma más adecuada.

$$\Phi(\theta) = \frac{\cos(0.326 \theta)}{\cos(0.308 \theta)} \quad \{\theta \in \mathbb{R} | 0 \leq \theta \leq \theta_{disk}\} \quad \text{Ec. 3}$$

La distribución de la energía de la corona solar fue desarrollada por Noring a partir de la relación lineal entre la intensidad solar y la distribución angular de la aureola en base logarítmica y se puede representar con la siguiente función:

$$\Phi(\theta) = e^{k\theta} \quad \{\theta \in \mathbb{R} | \theta_{disk} \leq \theta\} \quad \text{Ec. 4}$$



Donde  $\gamma$  es el gradiente de la curva en base logarítmica y  $K$  es la intersección de esa curva en una variación angular respecto del cero.

$$\kappa = 0.9 * \ln(13.5 * CSR) CSR^{-0.3} \quad \text{Ec. 5}$$

$$\gamma = 2.2 * \ln(0.52 * CSR) CSR^{0.43} - 0.1 \quad \text{Ec. 6}$$

$\Phi(\theta)$  está en milirradiantes.

Utilizando la distribución del disco solar es suficiente para la mayoría de las localizaciones y bajo la mayoría de las condiciones atmosféricas pero se ha demostrado que combinando las expresiones (3) y (4) los resultados obtenidos tienen mayor precisión si los comparamos con mediciones reales debido a que la componente del rayo directo es crítico para determinados CSR's. Además, en el borde del disco solar se produce un cambio significativo en la intensidad del flujo- que si se desprecia puede dar lugar a resultados falseados de la CSR.

De esta manera se normaliza la distribución del brillo solar  $\Phi(\theta)$ - que solo va a depender del valor de CSR.

### 4.3. Solar Limb Darkening

Limb darkening es un efecto óptico que se ve en las estrellas. La astrofísica define este fenómeno como el decrecimiento del brillo del disco del sol u otra estrella observando desde el centro hasta la periferia. Es un fenómeno que se aprecia fácilmente en las imágenes obtenidas del sol [13]. El oscurecimiento se debe a que en las capas donde la profundidad es mayor la temperatura es mayor y por lo tanto el brillo es mayor mientras que las capas menos profundas la temperatura es menor y por lo tanto se produce un oscurecimiento de la superficie.

Una buena manera de representar el fenómeno de Limb Darkening de forma empírica sería con la siguiente ecuación:

$$I(r) = I(0) \left[ 1 - u \left( 1 - \sqrt{\frac{a^2 - r^2}{a^2}} \right) \right] \quad \text{Ec. 7}$$

Donde  $a$  es el radio del disco solar,  $r$  es la distancia radial respecto del centro del disco y  $u$  es el coeficiente de oscurecimiento. Normalmente esta expresión se escribe en función de  $\theta$  o con  $\mu = \cos \theta$ .

$$I(\theta) = I(0)[1 - u(1 - \cos \theta)] = I(0)[1 - u(1 - \cos \mu)] \quad \text{Ec. 8}$$

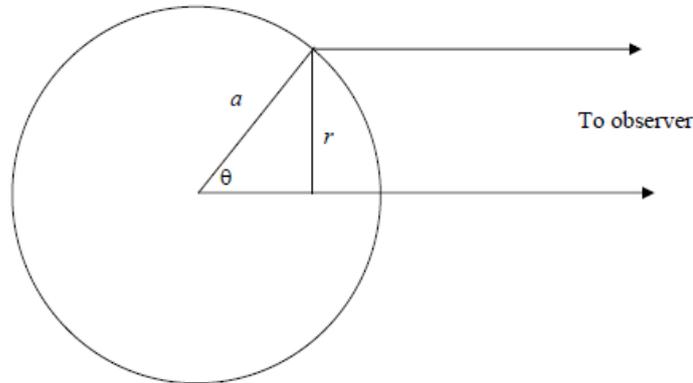


Figura 25: Representación geométrica del Limb Darkening

En las Ec. 7 y Ec. 8 ,  $I(0)$  representa la intensidad específica en el centro del disco. La intensidad específica en el extremo (donde  $r=a$  o  $\theta=90^0$ ) es  $I(0)(1-u)$ . El coeficiente de oscurecimiento puede ser escrito como  $u = [I(\text{centro}) - I(\text{Limb})] / I(\text{centro})$ . La Ec. 7 se representa en la siguiente gráfica para diferentes coeficientes de oscurecimiento ( $u$ ).

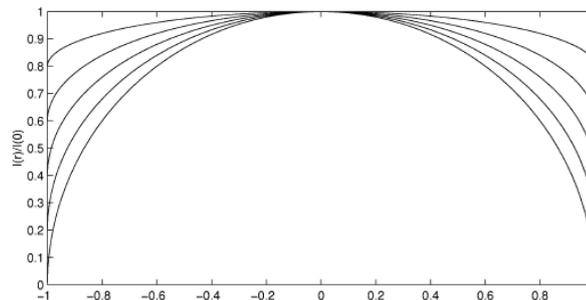


Figura 26: Aplicación de la ecuación 8 para diferentes coeficientes de oscurecimiento.

Desde la curva más pequeña hasta la superior el valor de  $u$  va disminuyendo con  $u = \{1.0, 0.8, 0.6, 0.4, 0.2, 0.0\}$ . Para  $u=1$  se representa como un círculo y se ha tomado como radio del disco igual a 1. Dependiendo de la longitud de onda los coeficientes de oscurecimiento ( $u$ ) varían siendo mayor el valor del coeficiente cuanto menor es la longitud de onda [14]. Para una mejor representación empírica se utiliza la siguiente expresión:

$$\frac{I(\theta)}{I(0)} = 1 - u - v + u \cos \theta + v \cos^2 \theta \quad \text{Ec. 9}$$

El ratio  $\frac{I(\theta)}{I(0)}$  depende de la longitud de onda pero mediante mediciones empíricas se calcula el ratio de la intensidad para toda la banda de longitudes de onda. Para el

cálculo en función de  $\theta$  es necesario introducir en la ecuación 10 los valores de  $u=0.84$  y  $v=-0.2$ .

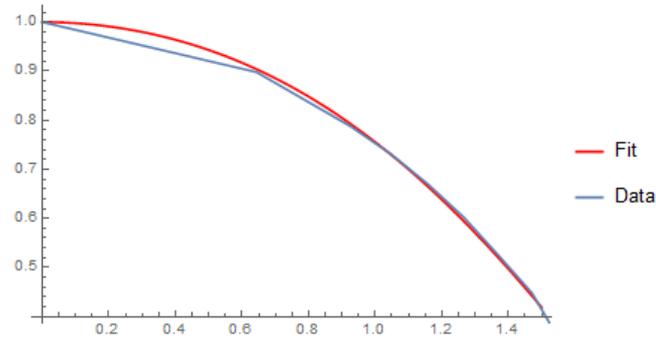


Figura 27: Aproximación datos empíricos a la ecuación 9 para el computo de todas las longitudes de onda

El modelo de la irradiancia para un valor de  $\theta_{max}$  determinado y normalizado se describe mediante la siguiente ecuación:

$$I(\theta) = 0.36 + 0.84 \left(1 - \frac{\theta^2}{\theta_{max}^2}\right)^{0.5} - 0.2 \left(1 - \frac{\theta^2}{\theta_{max}^2}\right) \quad \text{Ec. 10}$$

Simplificando:

$$I(\theta) = 1 - 0.5138 \left(\frac{\theta}{\theta_{max}}\right)^4 \quad \text{Ec. 11}$$

Un valor típico que se suele utilizar para  $\theta_{max}$  es 4.65 mrad [15]. El perfil de irradiancia queda de la siguiente manera para el modelo de Limb Darkening:

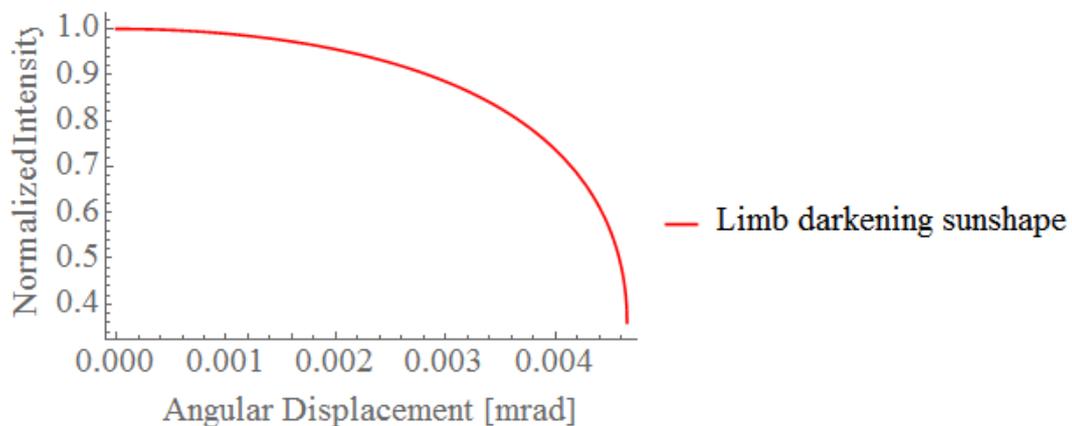


Figura 28: Distribución de la intensidad en función del ángulo  $\theta$  para el modelo de Limb Darkening



### 4.4. Kuiper

Aun eliminando la influencia de la atmósfera terrestre, la irradiancia solar disminuye con el incremento del ángulo desde el centro del Sol. El modelo Kuiper es el más utilizado para describir el perfil de la irradiancia del Sol en el extraterrestre. Este modelo es una evolución del modelo Limb Darkening en el que se tiene en cuenta la influencia de la longitud de onda en el perfil de distribución de la superficie solar [15]. Como es conocido, la estrella es una bola de gas en la que la temperatura disminuye al incrementar la distancia angular con el centro del Sol y por lo tanto la irradiancia también disminuye. Probablemente el modelo Kuiper sea el más utilizado para definir la superficie solar en el extraterrestre para los campos de CSP. La expresión que lo describe es:

$$\frac{L(\theta)}{L(\theta = 0)} = \frac{1 + \beta_k * \sqrt{1 - \frac{\tan^2(\theta)}{\tan^2(\theta_{disk})}}}{1 + \beta_k} \tag{Ec. 12}$$

donde  $L(\theta)$  es la irradiancia espectral que varía en función de la distancia angular respecto del centro del Sol y  $\beta_k$  es un parámetro que depende de la longitud de onda conocido como coeficiente de oscurecimiento

$$\beta_k = \frac{3 * h * c * \sqrt[4]{2}}{8 * k * \lambda * T_s} \tag{Ec. 13}$$

donde  $\lambda$  es la longitud de onda en m,  $h$  es la constante de Planck ( $6.63 * 10^{-34}$  J s),  $c$  es la velocidad de la luz ( $2.998 * 10^8$  m s<sup>-1</sup>),  $k$  es la constante de Boltzmann ( $1.38 * 10^{-23}$ ) y  $T_s$  es la temperatura del Sol en la superficie ( $5740$  °K) [16].

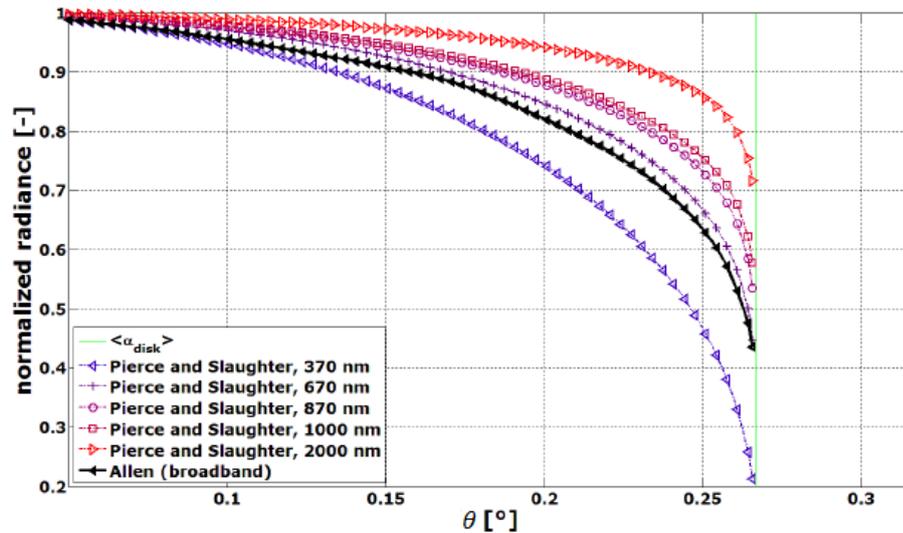


Figura 29: Perfiles extraterrestres para diferentes longitudes de onda

#### 4.5. Elección de los modelos a implementar

Una vez expuesto los diferentes modelos que se han formulado y ver que el software Tonatiuh ya tiene implementado los modelos Pillbox y Buie se ha decidido implementar la forma solar que sigue una distribución Gaussiana.

El motivo por el que se ha elegido una forma solar mediante una distribución Gaussiana es que como se comenta posteriormente en el apartado 5.2 los métodos de convolución utilizan una distribución Gaussiana. De esta forma se puede definir una distribución cualesquiera que tenga las mismas características que con lo software de convolución pero con las ventajas de los software de trazadores de rayos.

Además, existen programas comerciales que ya tienen implementados esta forma solar por lo que servirá de gran utilidad para la validación del modelo implementado en Tonatiuh. El proceso de validación se realiza en el apartado 7 donde se analiza los flujos en el receptor para unas condiciones iniciales.

Para darle mayor flexibilidad al usuario y poder introducir infinidad de forma solares se añade una segunda herramienta. El objetivo de esta funcionalidad es poder personalizar la forma solar. Introduciendo los datos de los puntos- en una tabla para crear la forma solar es más que suficiente. Softwares como SolTrace tienen implementado esta funcionalidad por lo que es de gran ayuda para la- comparación y posterior validación.

## 5. Estado del arte en software de simulación

El cálculo y diseño de instalaciones con tecnología CSP- es ardua y compleja con métodos de cálculo convencionales. Par calcular estos sistemas existen diversas variables a tener en cuenta. Por un lado es de gran importancia conocer el comportamiento del recurso solar (ver apartados 3 y 4): forma solar, intensidad, posición solar, etc.

Por otra parte las pérdidas ópticas y eficiencia de los concentradores ópticos. No toda la radiación que llega a los heliostatos es reflejada y dirigida al receptor (torre, absorbedor, etc.) Las pérdidas se deben a múltiples factores [17].

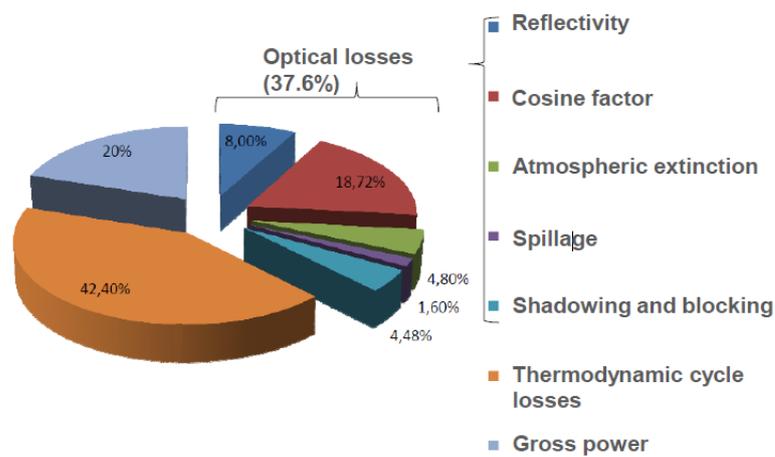


Figura 30: Pérdidas ópticas

Entre las más importantes y en orden de mayor a menor en cuanto a su influencia en las pérdidas están: el efecto coseno, reflectividad del espejo, atenuación- de la atmósfera, spillage- losses (pérdidas por dispersión), pérdidas por bloqueo o sombreado. De las más importantes, el efecto coseno- es la reducción debida al coseno del ángulo entre la radiación solar y una superficie normal. La superficie efectiva varía cíclicamente.

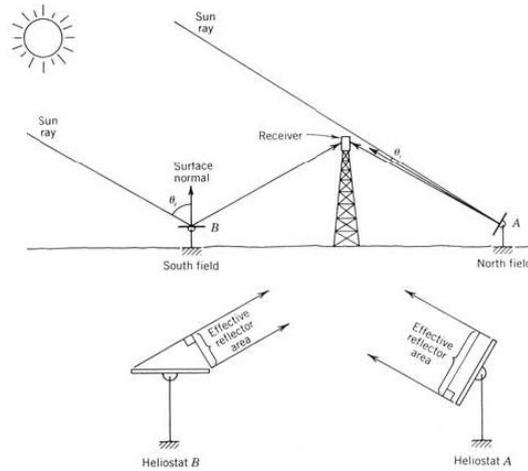


Figura 31: Efecto coseno

Los espejos utilizados en los heliostatos no son perfectos, microscópicamente no son lisos y aunque la reflectividad mediante un tratamiento superficial con plata es mejor que con aluminio para un amplio ancho de banda sigue existiendo anomalías.

El Spillage losses es debido a la difusión del rayo reflejado y el área limitada del receptor. Las pérdidas por difusión provienen de:

- El tamaño del Sol.
- Los errores de la superficie del espejo
- Deformación de la estructura
- Errores de orientación (track, sensores)
- Astigmatismo

Es necesario buscar una solución intermedia entre la calidad óptica de los espejos y el tamaño del receptor. Existe un compromiso económico.

Algunas de las pérdidas comentadas son temporales y depende del momento del día. Por ejemplo las pérdidas debidas por bloqueo y/o sombreado. Las pérdidas de sombra son aquellas en las que los rayos que emiten el sol no llegan al espejo porque algún objeto se interpone en el camino. En cambio las pérdidas por bloqueo son aquellas en las que un objeto se interpone entre el espejo y el receptor. Además, las características de la superficie de los espejos y el absorbedor varían constantemente por

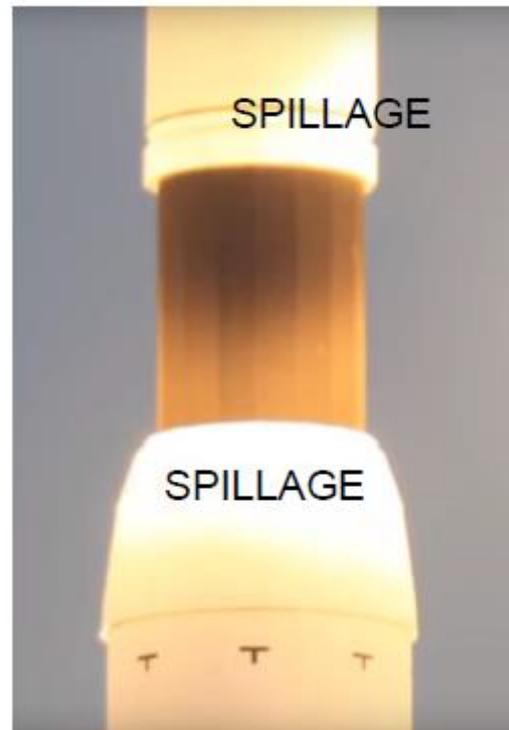


Figura 32: Desbordamiento óptico

las condiciones climatológicas (arena, suciedad, etc.). Los espejos se limpian aproximadamente cada semana pero también hay que tener en cuenta la corrosión.

Existen anomalías en cuanto al seguimiento de la referencia. Una mala inclinación debida a una orientación o- curvatura incorrecta. Un error de astigmatismo ocurre cuando los rayos incidentes son oblicuos crenado 2 puntos focales. Uno en la dirección sagital<sup>2</sup> y el otro en la dirección tangencial, generando un mapa de flujo de forma elíptica. Los errores de tracking son comunes y no se pueden evitar ya que los sensores de seguimiento no son ideales [18].

También es necesario diseñar y optimizar el layout del campo de heliostatos para maximizar el flujo incidente en el receptor.

### Annual losses and efficiency

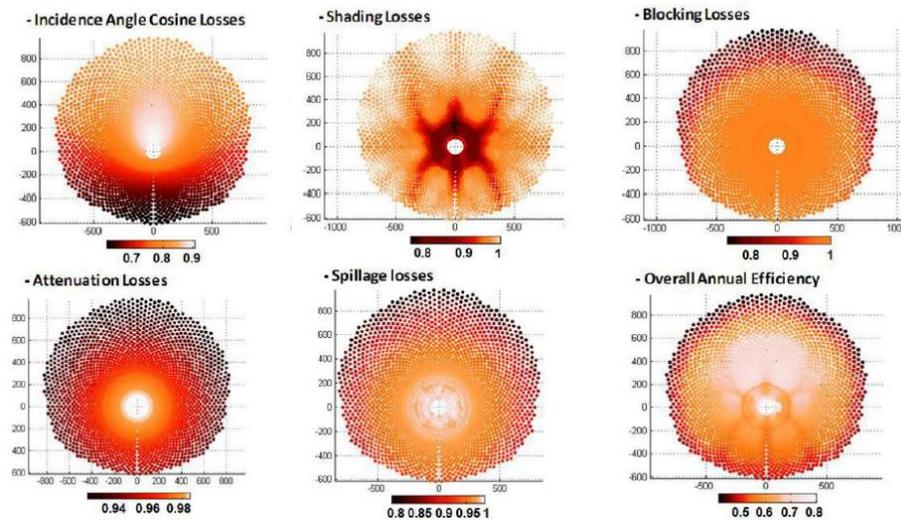


Figura 33: Pérdidas anuales en un campo de heliostatos

Existen más factores que disminuyen el valor de la eficiencia de los heliostatos además de los ya mencionados anteriormente y que son los más claros ya que son fenómenos físicos. A continuación se exponen otro tipo de errores y un valor orientativo del error que puede generar para la reflexión de los rayos [19]:

<sup>2</sup> La línea sagital es una vertical de referencia que teóricamente cruza el cuerpo por la parte media y central, a modo de plomada imaginaria.



Error type	Error source	Error (mrad)
General	Mirror surface error (curvature, waviness, etc.) at 25 °C	1.5
	Canting	1.2
	Adjustment of rotation axis (non orthogonality, horizontality of foundation, etc.)	0.5
	Electronic control (encoder, resolution, sun algorithm, ...)	1.0
	Thermal deformation of structure	0.3
	Partial sum	2.24
Mass forces for elevation axis motions (between 30 ° and 90 °)	Bending of pedestal	0.6
	Support structure deformation	1.55
	Mirror deformation	0.4
	Elasticity of drive mechanism	-
	Partial sum	1.71
Wind load (18 km/h)	Bending and torsion of pedestal	0.4
	Support structure deformation	0.1
	Facet deformation	0.2
	Drive mechanism tolerance	0.3
	Elasticity of drive mechanism	0.4
	Partial sum	0.68
	Total	2.9

Figura 34:- Valores del error típicos

Una vez que se tiene en cuenta las diferentes variables que entran en juego, actualmente existen dos tendencias para la simulación óptica de plantas CSP. Por un lado están los softwares que hacen una aproximación estadística conocidos como software de trazadores de rayos (ray-tracing) que generalmente permiten mayores precisiones en sus resultados y otros que hacen una aproximación analítica que son conocidos como los softwares de convolución.

### 5.1. Trazador de rayos basados en el método de Monte Carlo

Los programas que se basan en el trazador de rayos o ray-tracing utilizan un método estadístico en el que se elige de forma aleatoria una cantidad de rayos que provienen de una superficie1 y determina cuál de todos ellos llegan a la superficie2. La irradiancia de cada superficie elemental es proporcional al número de rayos impactados [19]. En el caso de que exista un heliostato y un receptor, el algoritmo se aplica dos veces, o tantas como número de intersecciones haya hasta que el rayo no interaccione con ninguna superficie o sea absorbido. El primero entre el sol y la superficie donde se va a reflejar los rayos que tendrá una distribución de energía con la forma solar y luego entre el heliostato y el receptor que tendrá una distribución de la energía combinando la del sol con el error de distribución debido a la superficie del heliostato [20].

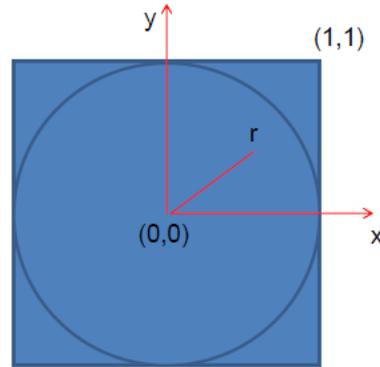
El método de Monte Carlo es una técnica numérica para calcular probabilidades utilizando secuencias de números aleatorios. Para el caso de una sola variable el procedimiento es el siguiente:

- Generar una serie de números aleatorios  $n_1, n_2, \dots, n_m$ , uniformemente distribuidos en  $[0,1]$

- Usar esta secuencia para producir otra secuencia  $x_1, x_2, \dots, x_m$  distribuida de acuerdo a la función de probabilidad (pdf) que se está interesado.

## Monte Carlo method

Find  $\pi$  with numerical experiment



$$\pi = 4 \frac{S_{circle}}{S_{square}} \quad (\text{by definition})$$

$$= 4 \frac{\text{probability to hit circle}}{\text{probability to hit square}}$$

```

N_tot=10^6 % total No. of random tests
N_cir=0
for i = 1..N_tot
  x=rand() % take random number for x
  y=rand()
  r=sqrt(x^2+y^2) % location with radius
  if r <= 1 % random test
    N_cir = N_cir+1 % hit the circle
  end
end
pi = 4*N_cir/N_tot % answer

```

rand() is random number generator in range (0,1)

Figura 35: Método de Monte Carlo

Para realizar gráficos digitales, el trazador de rayos es una técnica de representación de imágenes que trazan el camino de la luz como píxeles que intersecciona con objetos virtuales y simula los efectos que se producen en él. La técnica es capaz de producir un alto grado de realismo visual. El trazador de rayos es capaz de simular una amplia variedad de efectos ópticos, como la reflexión, refracción y dispersión.

La combinación del método de Monte Carlo con el trazador de rayos en softwares de simulación- convierten estos softwares en una herramienta visual porque consiste en lanzar un haz de rayos a un sistema determinado por el usuario [21]. Cada rayo emitido es un fotón en el que tiene una información determinada (dirección, intensidad). El cómputo de todos los rayos emitidos sigue una función de probabilidad determinada por el usuario.

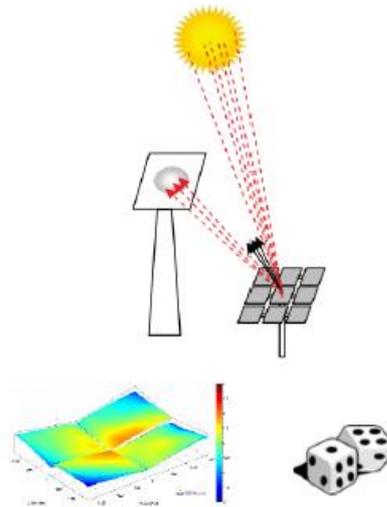


Figura 36: Solar ray tracing (Fuente: SFERA)

Los rayos son emitidos hacia un sistema compuesto por un conjunto de colectores y uno o varios receptores. La ventaja de este software es que se puede implementar colectores de diferentes formas y se analizan las intersecciones que se producen entre los rayos y los elementos del sistema con la finalidad de recabar información útil. Con este método se analiza el comportamiento óptico y energético y proporciona información en detalle de las distribuciones de flujo solar incidentes en las superficies. Una de las ventajas es que reproduce la dirección de los rayos entre los diferentes elementos con una alta precisión tanto para sistemas pequeños o complejos con un coste computacional aceptable.

Los programas que utilizan algoritmos de Monte Carlo analizan la colisión de los rayos con los diferentes elementos. El problema es que este método no es muy eficiente porque al generar los rayos de forma aleatoria (de forma acotada), gran parte de ellos puede que no colisionen con ningún elemento lo que conlleva a que se pierdan en el espacio. El método utilizado no es el más eficiente de todos y por lo tanto esta técnica no se recomienda para la optimización de sistemas de concentración solar donde es necesario simular multitud de sistemas.

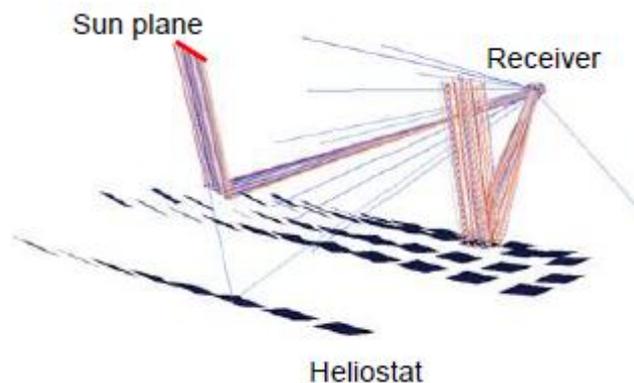


Figura 37: Representación del escenario con los rayos

La secuencia típica del trazador de rayos de Monte Carlo para un sistema sencillo sigue el siguiente algoritmo:

1. Introducir las superficies a estudio en el escenario.
2. Selecciona- aleatoriamente un rayo emitido por el sol.
3. Calcula la intersección del rayo con todas las superficies del escenario.
4. Encontrar la superficie más cercana, si no existe intersección el rayo se pierde en el escenario y se vuelve al punto 2.
5. Selecciona un número aleatorio  $R$  y lo compara con la reflectividad de la superficie  $0 < \rho < 1$ .  
Si  $R < \rho$  el rayo es absorbido, entonces vuelve al punto 2, sino al punto 6.
6. El rayo es reflejado. Selecciona un número aleatorio  $R$  y compara con la componente difusa  $d$  del punto.  
Si  $R > d$ , entonces se produce reflexión especular, sino reflexión difusa
7. Selecciona la nueva dirección para :  
Difusa → en cualquier dirección igualmente probable.  
Especular → alrededor de la dirección especular con el error predefinido.
8. Ir al punto 2 mientras no interseccione con ninguna otra superficie.

Existe gran cantidad de softwares que aplican conceptualmente este método de los cuales los más representativos se exponen a continuación:

Code	Comments
Tonatiuh™	Open-source and free. Monte Carlo ray tracer (MCRT) (CENER)
SOLFAST-4D™	MCRT that will evolve to free open source code SOLSTICE (PROMES)
SolTrace™	MCRT. Freeware (NREL)
STRAL™	MCRT. Commercial (DLR)
TracePro™	Commercial design and simulation software for optical and illumination designers (ray tracer)
Zemax™	

Figura 38: Software que utilizan el método de trazador de rayos

De todos ellos, el proyecto se desarrolla con el software de Tonatiuh implementado por el Centro Nacional de Energías Renovables (CENER). Entre los objetivos que inspiran el diseño de Tonatiuh destaca su vocación de convertirse en uno de los programas de simulación de sistemas solares de concentración de uso más extendido y consolidado [22]. Su flexibilidad y carácter general que le otorga la



capacidad para simular la gran mayoría si es que no la totalidad de los sistemas solares térmicos de concentración cuyo diseño tiene sentido plantear. Su facilidad de aprendizaje, uso, mantenimiento, mejora y extensión de su funcionalidad.

Se ha optado por configurar el desarrollo de Tonatiuh como un proyecto de software de código abierto y multiplataforma. Programarlo en C++ haciendo uso de las facilidades que dicho lenguaje proporciona para el desarrollo de software orientado a objetos. Utilizando en su desarrollo librerías y herramientas de código abierto estable, consolidadas y promovidas por empresas o consorcios de gran reputación. Implementando una gran parte de la funcionalidad del programa en base a plugs-in que pueden ser creados, modificados o extendidos por el usuario.

Por todas las facilidades que ello supone y el entorno amigable que proporciona Tonatiuh hacen que sea la herramienta más conveniente para poder implementar los nuevos modelos de sol.

## 5.2. Método de convolución

El método de convolución sigue una filosofía contrapuesta a los trazadores de rayos. De hecho los softwares que utilizan este método se les conoce como programas analíticos basados en la convolución de distribuciones gaussianas. En primer lugar se hace una explicación matemática de en qué consiste este método y luego como se ha derivado para su aplicación en este campo [19].

La distribución de probabilidad de la suma de dos o más variables aleatorias independientes se llama convolución de las distribuciones de las variables originales.

El método de convolución es entonces la suma de dos o más variables aleatorias para obtener una nueva variable aleatoria con la distribución de probabilidad deseada. La ecuación que la rige es la siguiente.

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(n)g(t - n)dn \quad \text{Ec. 14}$$

Se utilizan modelos analíticos para analizar el diseño del campo y la optimización en la densidad de flujo se aproxima a través de modelos simplificados de la convolución. Estos métodos de modelización incluyen HFLCAL, DELSOL y CRU-RECLL.

Code	Comments
DELSOL™	Power-tower system model and field/receiv optimization tool. Free (NREL)
CAMPO™	Layout and field optimization. (UNIZAR)
HFLCAL™	Layout and field optimization. Commercial (DLR)

Figura 39: Software que utilizan el método de convolución

Los softwares que utilizan este método- generan un- mapa de flujo a- partir de un rayo con la forma de una distribución de probabilidad definida por el perfil del sol y las desviaciones del rayo (errores de inclinación y seguimiento) [23]. El coste computacional es menor que los métodos basados en trazadores de rayos pero son menos precisos. El mapa de flujo resultante de todo el campo es generado por la superposición de las densidades de flujo de cada heliostato [24]. Por ello los métodos de aproximación se justificaron para grandes campos por el teoría del límite central, que indica que la superposición de un gran número de funciones de distribuciones aleatorias convergen hacia una distribución gaussiana.

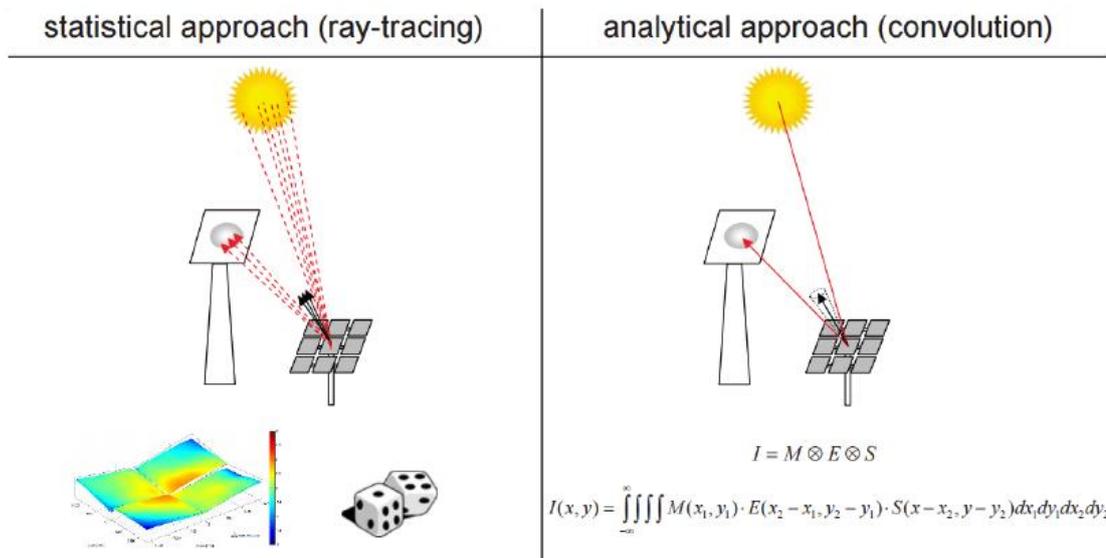


Figura 40: Comparativa Ray-tracing vs Convolution

## 6. Implementación en Tonatiuh

En este apartado se expone la manera de implementar los modelos que se van a introducir en Tonatiuh. Explicación de cómo se ha enfocado para implementarlos. NE el apartado de Anexos aparece el código en C++.

## 6.1. Gaussian sunshape

### 6.1.1. Descripción

En este proyecto, uno de los modelos que se va a implementar es un perfil de irradiancia que siga una distribución Gaussiana. Para ello se supone que el Sol es puntual y el perfil de irradiancia describe una función Gaussiana además de considerar que la distribución Gaussiana en 3 dimensiones se puede simplificar a una de una dimensión en función del ángulo  $\theta$  (desviación de los rayos respecto del vector solar) debido a la simetría impuesta al Sol. La función Gaussiana que describe el perfil de irradiancia ( $L$ ) queda de la siguiente manera:

$$L=(\theta, \varphi) \begin{cases} e^{-\frac{\theta^2}{2\sigma^2}} \\ \frac{1}{\sqrt{2\pi}\sigma} \end{cases} \forall \theta \leq \theta_{max} \quad \text{Ec. 15}$$

$$0 \forall \theta > \theta_{max}$$

Siendo  $\sigma$  la desviación estándar de la función de probabilidad Gaussiana utilizada para modelar la irradiancia del sol y  $\theta_{max}$  el valor máximo de la función Gaussiana en este caso  $\pi/2$ . Se considera el DNI (direct normal irradiance) de valor unidad ya que es solo un factor de escala.

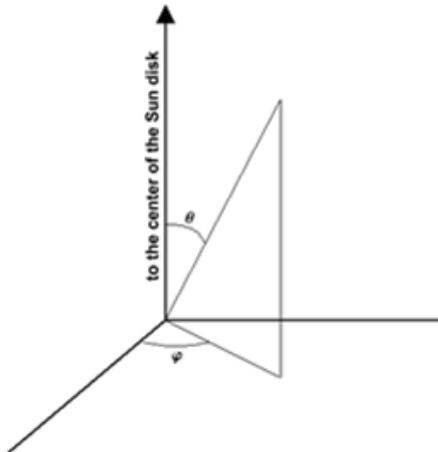


Figura 41: Sistema de coordenadas locales usado para la implementación del modelo Gaussiano

Por definición, la irradiancia total es:

$$I_b = \int_0^{2\pi} \int_0^{\theta_{max}} L \cos \theta \sin \theta d\varphi d\theta \quad \text{Ec. 16}$$

El modelo de sunshape ( $\phi(\theta) = e^{-\frac{\theta^2}{2\sigma^2}}$ ) es la función de probabilidad que describe la distribución angular de la intensidad incidente. Al ser la irradiancia simétrica la forma solar no depende de  $\varphi$  y además por definición  $\phi_{ab} = L(\theta) \cos \theta$  queda la ecuación de la siguiente manera:



$$I_b = 2\pi \int_0^{\theta_{max}} \phi_{ab} \sin \theta \, d\theta \quad \text{Ec. 17}$$

Por definición,  $\phi(\theta) = \frac{\phi_{ab}(\theta)}{\phi_{max}}$

$$1 = 2\pi \frac{\phi_{max}}{I_b} \int_0^{\Delta\theta} \phi(\theta) \sin \theta \, d\theta \quad \text{Ec. 18}$$

Finalmente queda:

$$2\pi \frac{\phi_{max}}{I_b} = \frac{1}{\int_0^{\Delta\theta} \phi(\theta) \sin \theta \, d\theta} = pMax \quad \text{Ec. 19}$$

### 6.1.2. Problemática

Para determinar la probabilidad de que una variable aleatoria caiga en una región específica del espacio de posibilidades estará dada por la integral de la densidad de esta variable entre uno y otro límite de dicha región. La función de densidad de probabilidad es positiva a lo largo de todo su dominio y su integral sobre todo el espacio es de valor unitario. Para este caso la función de densidad de probabilidad es:

$$PDF_{Gaussian}(\sigma, \theta) = pMax * \phi(\theta) * \sin \theta \quad \text{Ec. 20}$$

Lamentablemente, al calcular la pMax (Ec. 19) y utilizar una distribución tan característica como la Gaussiana sale como resultante la siguiente ecuación:

$$pMax = - \frac{2e^{\frac{\sigma^2}{2}} \sqrt{\frac{2}{\pi}}}{\sigma (\text{Erfi}[\frac{-i\theta + \sigma^2}{\sqrt{2}\sigma}] + \text{Erfi}[\frac{i\theta + \sigma^2}{\sqrt{2}\sigma}])} \quad \text{Ec. 21}$$

Dando lugar a números complejos y valores de Erfi (error imaginario de la función) y complicando los cálculos más adelante. Para ello se busca un cálculo de la integral más sencilla y lo más aproximada posible.

### 6.1.3. Desarrollo matemático

Aquí se explica el desarrollo matemático seguido para la posterior implementación en el código. Para que el cálculo sea lo más sencillo y aproximado para un correcto funcionamiento se ha determinado que el cálculo de la Ec. 19 se haga mediante la definición de la integral mediante las sumas de Riemann [25], como la suma de áreas muy pequeñas. Para ello el valor de  $\delta$  debe ser lo más pequeño para una buena aproximación, en este caso con un  $\delta=0.00001$  es más que suficiente dando la siguiente  $pMax_{aprox}$  aproximada:

$$pMax_{aprox} = \frac{1}{(9.98 * 10^{-7} * e^{\frac{-0.005}{\sigma^2}} + 9.98 * 10^{-7} * e^{\frac{-0.0049}{\sigma^2}} + \dots)} \quad \text{Ec. 22}$$

Aplicando la Ec. 20 pero con la nueva  $pMax_{aprox}$  la función de densidad de probabilidad resultante utilizando un valor de  $\sigma=2.485$  mrad queda de la siguiente manera:

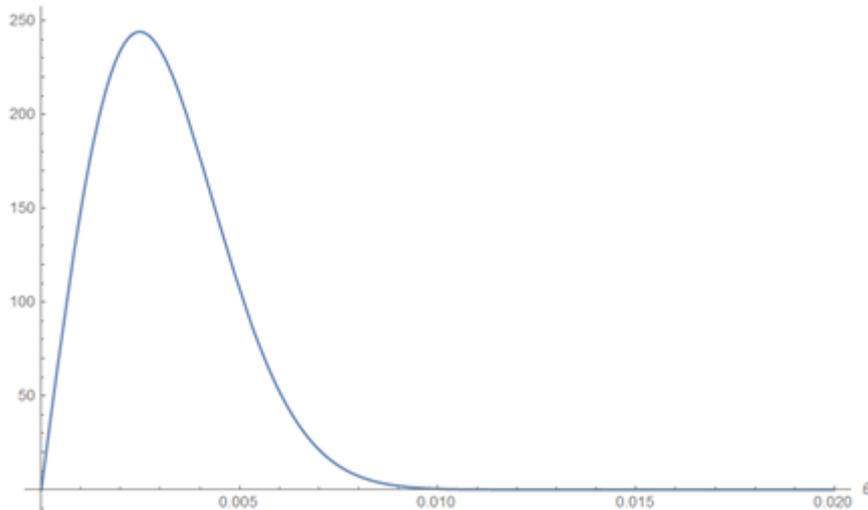


Figura 42: Función de densidad de probabilidad

Además la suposición de que el  $\theta_{max}=5\sigma$  es debido a que con estos valores la probabilidad de que todos los rayos estén dentro de la campana es de 0.999996, aproximadamente 1.

#### 6.1.4. Método de generación de números aleatorios

Como se explica en el apartado 5.1, Tonatiuh genera una serie de números aleatorios de tal manera que la sucesión de números aleatorios estén dentro de la distribución de densidad de probabilidad (PDF) calculado anteriormente. Para ello se utilizan el método de aceptación y rechazo. El algoritmo es el siguiente.

- 1) Se genera un número aleatorio “ $\theta$ ”  $\rightarrow \theta \in [0,a]$

- 2) Se genera un segundo número aleatorio independientemente del anterior “K”  $\rightarrow K \in [0,b]$

Para que la cantidad de números aceptados sea mayor y por lo tanto la eficiencia del método sea mejor se limita la región para que a la hora de generar números den valores dentro de este rango. Más adelante se explica el método para calcular el valor máximo de la función y así poder limitar la región y que la eficiencia del método sea mayor lo que disminuirá el coste computacional.

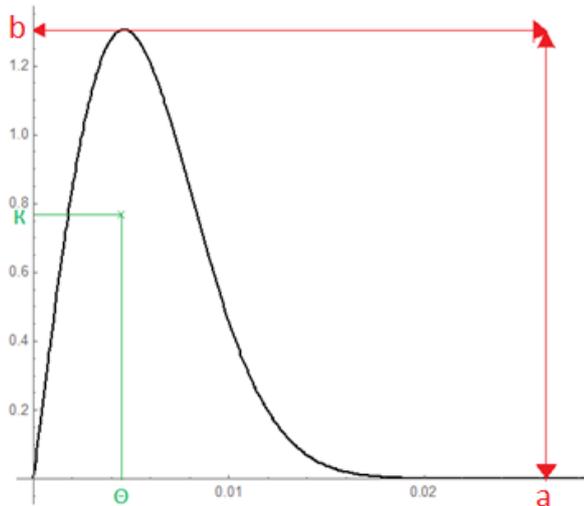


Figura 43: Procedimiento aceptación-rechazo

- 3) Entonces:
- Si  $K \leq f(\theta)$ , entonces  $X = \theta$
  - Si  $K > f(\theta)$ , entonces volver al paso 1).

De esta manera se van generando finitos números de los cuales se aceptan aquellos que queden dentro de la función de densidad de probabilidad.

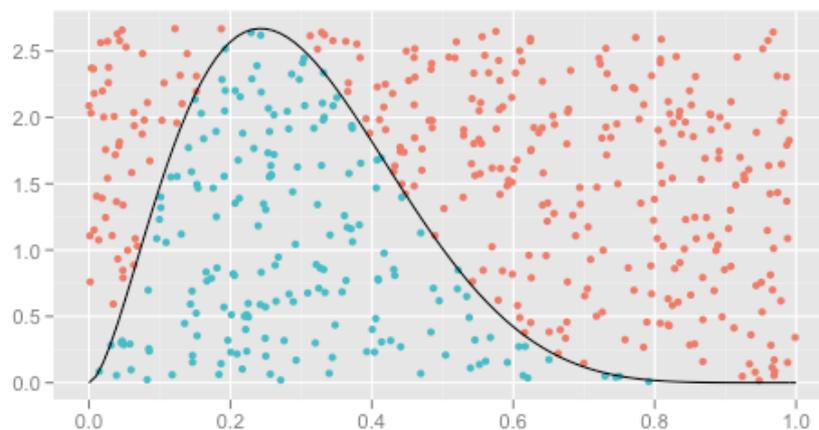


Figura 44: Método de aceptación y rechazo



Por otro lado es necesario buscar el valor de  $\theta$  donde PDFaprox es máximo para poder limitar la región en el método de aceptación y rechazo. Para ello es necesario derivar la función e igualar a 0. A partir de la Ec. 20 y despreciando el último término ya que no depende de  $\theta$ , se deriva.

$$\left[ e^{-\frac{\theta^2}{2\sigma^2}} * \sin[\theta] \right]' = \frac{e^{-\frac{\theta^2}{2\sigma^2}} (\sigma^2 \cos[\theta] - \theta \sin[\theta])}{\sigma^2} \quad \text{Ec. 23}$$

Como

$$e^{-\frac{\theta^2}{2\sigma^2}} > 0$$

La función que queda es:  $f(\sigma, \theta) = \sigma^2 \cos[\theta] - \theta \sin[\theta]$ . Con la función resultante es imposible calcular sus raíces mediante el cálculo convencional por lo que es necesario utilizar el método numérico, en este caso el elegido es el de Newton-Raphson. Para ello es inevitable volver a derivar para poder aplicar el método.

$$\frac{df(\sigma, \theta)}{d\theta} = -\theta \cos[\theta] - \sin[\theta] - \sigma^2 \sin[\theta] \quad \text{Ec. 24}$$

El método es el siguiente para el cálculo de la raíz para el cual la función de densidad de probabilidad es máxima.

$$\theta_{n+1} = \theta_n - \frac{f(\sigma, \theta)}{f'(\sigma, \theta)} \quad \text{Ec. 25}$$

Dando un valor inicial de  $\theta_0 = \sigma$  e iterando hasta que  $\Delta\theta$  sea un valor cercano a 0. Con este algoritmo se puede calcular cualquier raíz independientemente del valor de  $\sigma$  que se le introduzca.

Una vez deducido el valor de  $\theta_{max}$  se calcula el valor de la distribución Gaussiana a partir de la siguiente ecuación:

$$PDF(\theta_{max}, \sigma) = e^{-\frac{\theta_{max}^2}{2\sigma^2}} * \sin \theta_{max} * pmax \quad \text{Ec. 26}$$

### 6.1.5. Implementación y funcionamiento del código

Cuando el usuario quiere utilizar el modelo Gaussiano para cualquier tipo de estudio el programa entra en la clase definida como “*SunshapeGaussian*”. Los valores

de entrada que necesitara el programa serán el DNI y el valor de desviación estándar ( $\sigma$ ) para la distribución de la irradiancia.

A partir de aquí, y antes de producirse la generación de rayos, la clase calcula los parámetros explicados en el apartado 6.1.3 desde el valor de pMax, el punto máximo y su imagen para poder la limitar la región donde se van a generar los números aleatorios. A partir de ahí se generan los rayos que están dentro de la función de densidad de distribución y emitidos a los elementos que estén en el escenario en cuestión.

Se presenta un diagrama de bloques del funcionamiento del algoritmo para el cálculo de los valores de la forma solar para comprender mejor el funcionamiento interno del programa:

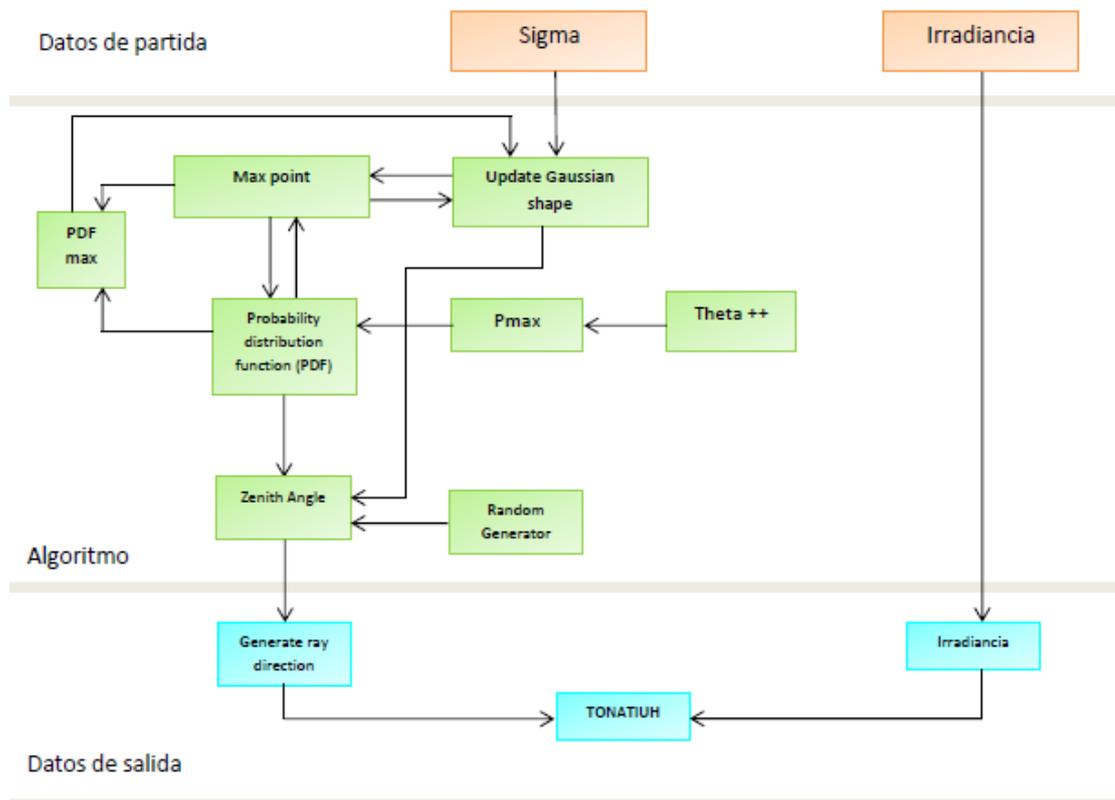


Figura 45: Diagrama de bloques del algoritmo Gaussian Sunshape

## 6.2. User defined sunshape

### 6.2.1. Descripción

El modelo que se describe a continuación no es un modelo en el que la forma solar siga una distribución concreta. La idea principal es facilitar al usuario una herramienta para que pueda hacer las simulaciones con un perfil de distribución de los rayos solares personalizada.



Normalmente los modelos expuestos anteriormente se han desarrollado a partir de recabar muchas información de diferentes lugares durante muchos años. La posibilidad de que el propio usuario pueda introducir los datos de su forma solar personalizada hace que el estudio sea más exacto ya que se parten con datos de entrada experimentales y no una aproximación con el error que con ello conlleva.

Por lo tanto los datos de partida serán un conjunto de puntos representados en un plano. Siendo estos puntos el ángulo respecto del centro del sol ( $\theta$ ) y el valor de la intensidad por unidad (I). El primer paso es representar la forma solar como un conjunto de funciones a trozos. Para simplificar el estudio cada par de puntos se unen con una recta. Cuanto mayor sea el número de puntos de partida, más exacto será la forma solar.

La forma solar será una combinación de rectas a trozos que describan el perfil de irradiancia y queda de la siguiente manera:

$$L=(\theta, \varphi) \begin{cases} m_i \theta_i + n_i & \forall \theta_i \leq \theta < \theta_{i+1} \\ 0 & \forall \theta > \theta_{max} \end{cases} \quad \text{Ec. 27}$$

Por definición, la irradiancia total es:

$$I_b = \int_0^{2\pi} \int_0^{\theta_{max}} L \cos \theta \sin \theta d\varphi d\theta \quad \text{Ec. 28}$$

El modelo de sunshape ( $\phi_i(\theta) = m_i \theta_i + n_i$ ) es la función de probabilidad que describe la distribución angular de la intensidad incidente. Al ser la irradiancia simétrica la forma solar no depende de  $\varphi$  y además por definición  $\phi_{ab} = L(\theta) \cos \theta$  queda la ecuación de la siguiente manera:

$$I_b = 2\pi \int_0^{\theta_{max}} \phi_{ab} \sin \theta d\theta \quad \text{Ec. 29}$$

Por definición,  $\phi(\theta) = \frac{\phi_{ab}(\theta)}{\phi_{max}}$

$$1 = 2\pi \frac{\phi_{max}}{I_b} \int_0^{\Delta\theta} \phi(\theta) \sin \theta d\theta \quad \text{Ec. 30}$$

Finalmente queda:

$$2\pi \frac{\phi_{max}}{I_b} = \frac{1}{\int_0^{\Delta\theta} \phi(\theta) \sin \theta d\theta} = pMax \quad \text{Ec. 31}$$



### 6.2.2. Desarrollo matemático

Para el cálculo de las rectas es necesario calcular la pendiente y la componente independiente.

$$y(\theta) = m\theta + n \quad \text{Ec. 32}$$

$$m_i = \frac{I_i - I_{i+1}}{\theta_i - \theta_{i+1}} \quad \text{Ec. 33}$$

$$n_i = I_i - m_i * \theta_i \quad \text{Ec. 34}$$

Por lo tanto la representación matemática de la función a trozos de la forma solar queda de la siguiente forma:

$$\phi_i(\theta) = \left\{ \begin{array}{ll} n_1 + m_1 \theta & \theta_1 \leq \theta < \theta_2 \\ n_2 + m_2 \theta & \theta_2 \leq \theta < \theta_3 \\ \vdots & \vdots \\ n_n + m_n \theta & \theta_n \leq \theta < \theta_{n+1} \end{array} \right\} \quad \text{Ec. 35}$$

Resolviendo de la Ec. 19 el valor de pMax queda de la siguiente manera:

$$pMax_i(\theta) = \left\{ \begin{array}{ll} C_1 - n_1 \cos \theta - m_1 \theta \cos \theta + m_1 \sin \theta & \theta_1 \leq \theta < \theta_2 \\ C_2 - n_2 \cos \theta - m_2 \theta \cos \theta + m_2 \sin \theta & \theta_2 \leq \theta < \theta_3 \\ \vdots & \vdots \\ C_n - n_n \cos \theta - m_n \theta \cos \theta + m_n \sin \theta & \theta_n \leq \theta < \theta_{n+1} \end{array} \right\} \quad \text{Ec. 36}$$

El valor de la constante “C” aparece al hacer la integral, podría tomar un valor cualquiera pero para la función sea continua se impone que la primera función empiece en 0 y que la imagen de la función al final del intervalo sea igual a la imagen de la siguiente función en ese punto. Es decir que  $f_1(0)=0$  y que  $f_n(\theta)=f_{n+1}(\theta)$ . Por lo tanto los valores de las constantes se hallan de la siguiente manera:

$$\left\{ \begin{array}{ll} C_1 = \theta_1 + (n_1 + m_1 \theta_1) * \cos \theta_1 - m_1 \sin \theta_1 & i = 1 \\ C_i = C_i + (-n_{i-1} + n_i - m_{i-1} \theta_1 + m_i \theta_1) * \cos \theta_1 + (m_{i-1} - m_i) \sin \theta_1 & i = 2, \dots, n \end{array} \right\} \quad \text{Ec. 37}$$

Una vez que se calculan los valores de las constantes “C” se puede calcular pMax con la ayuda de la Ec. 19:

$$pMax = \frac{1}{\sum [C_i - n_i \cos \theta - m_i \theta + \cos \theta + m_i \sin \theta]_{\theta_i}^{\theta_{i+1}}} \quad \text{Ec. 38}$$

Para este caso la función de densidad de probabilidad queda:

$$PDF_{Gaussian}(\sigma, \theta) = pMax * \phi(\theta) * \sin \theta \quad \text{Ec. 39}$$

### 6.2.3. Implementación

El primer paso es la lectura de los datos de partida. En este caso, la forma solar se introduce en una tabla donde se guardarán los ángulos y su intensidad, normalizada, correspondientemente. Por otro lado se introduce el valor de irradiancia para poder evaluar diferentes escenarios (día claro, nuboso, etc.)

Tras la lectura de los datos de partida, el programa calcula los parámetros necesarios. Al estar enfocado el programa en dividir la forma solar en funciones a trozos lo primero que hace es calcular los parámetros de las rectas (Ec. 33 y Ec. 34) y los guarda en vectores. Lo mismo se hace para calcular las constantes para poder calcular una función de densidad de probabilidad continua. Finalmente, como se ha explicado en el apartado anterior se calculan los parámetros de Pmax, theta máximo y el valor de la raíz donde la función de densidad de probabilidad se hace máximo.

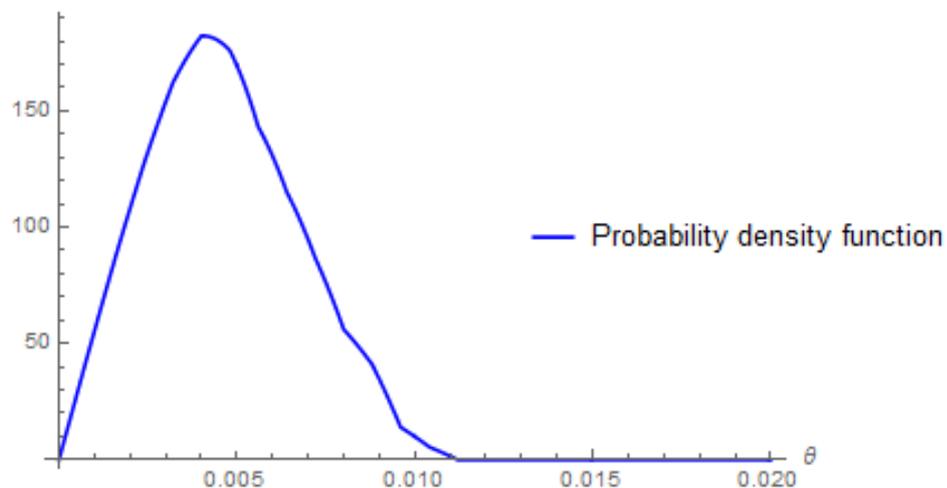


Figura 46: Función de densidad de probabilidad

A partir de aquí se utiliza el método de aceptación y rechazo que se ha explicado anteriormente en el apartado Método de generación de números aleatorios 6.1.4 y se van generando los diferentes valores de theta para los cuales su imagen están dentro de la función de densidad de probabilidad **¡Error! No se encuentra el origen de la referencia..** De esta manera es como se ejecuta el programa implementado. Como se ha hecho en el modelo anterior se expone un diagrama de bloques para una mejor comprensión del funcionamiento del programa:

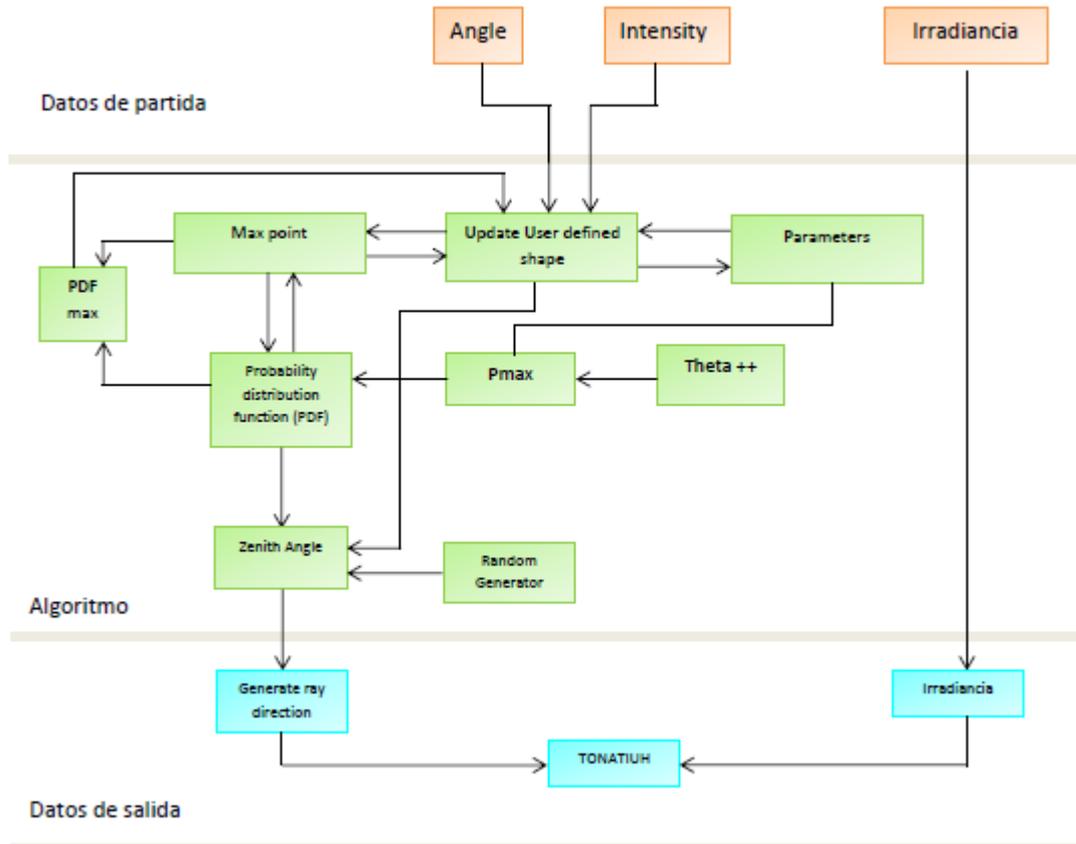


Figura 47: Diagrama de bloques del algoritmo User Defined Sunshape

## 7. Validación de los modelos

En este apartado se exponen las validaciones y verificaciones obtenidas tras las simulaciones con los modelos solares implementados, con el objetivo de comprobar que funcionan correctamente.

Para ello se introduce una superficie plana totalmente horizontal sobre la que se simulan 80 millones de rayos. Todos ellos saliendo desde un mismo punto y con una dirección perpendicular a dicha superficie. La distribución de la densidad de flujo que se obtiene en la superficie plana debería ser igual a la distribución de la irradiancia del

modelo. Lógicamente, cuanto mayor sea el número de rayos mayor será la exactitud de los resultados.

## 7.1. Modelo Gaussiano

### 7.1.1. Validación mediante Tonatiuh

Una vez implementado el código (ver ANEXOS) en el software y comprobado que funciona correctamente se han hecho diversas simulaciones para comprobar que los cálculos teóricos realizados en el apartado 6.1.3 son correctos. En una primera validación se ajusta el valor de  $\sigma=2.485$  mrad y se obtiene los datos tras la simulación.

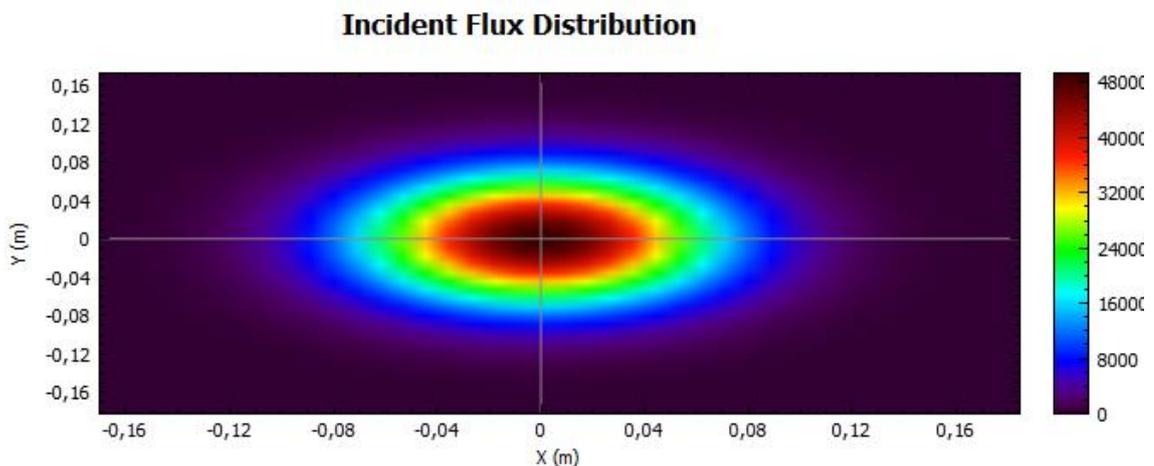


Figura 48: Distribución Gaussiana del flujo incidente en un heliostato

En la Figura 48 se observa la distribución del flujo incidido en la superficie planta. Cuanto más cerca del centro de la superficie mayor es el flujo y la imagen es simétrica ya que en si es una imagen del flujo emitido por el Sol. El punto donde el flujo es máximo es en el punto  $(x,y) \rightarrow (1.26 * E-16, -0.0024)$ , aproximadamente en el centro del heliostato. Para asegurarnos es necesario hacer un tratamiento de los datos obtenidos y para ello nos apoyamos con el software *Wolfram Mathematica*. Se busca ajustar los datos obtenidos a una función Gaussiana de la siguiente forma:

$$modelo(\sigma, \theta) = a * e^{-\frac{\theta^2}{2 * \sigma^2}} \quad \text{Ec. 40}$$

Los datos obtenidos con el programa están en coordenadas cartesianas por lo que es necesario hacer una transformación y pasarlas a coordenadas polares donde sólo depende del ángulo  $\theta$  y se hace el ajuste de los datos obtenidos con la función modelo.

En estudios anteriores, se utilizan rangos de  $\sigma$  comprendidos entre 2.24 y 2.73 mrad. En una primera comprobación se utiliza un valor medio de 2.485 mrad que se introduce en la simulación y se compara los datos obtenidos con el modelo.

	Estimate	Standard Error
<b>a</b>	0.998455	0.0000174832
<b><math>\sigma</math></b>	-0.00248794	$3.08045 \times 10^{-8}$

Se comprueba que el valor de es prácticamente igual al valor introducido en el programa.

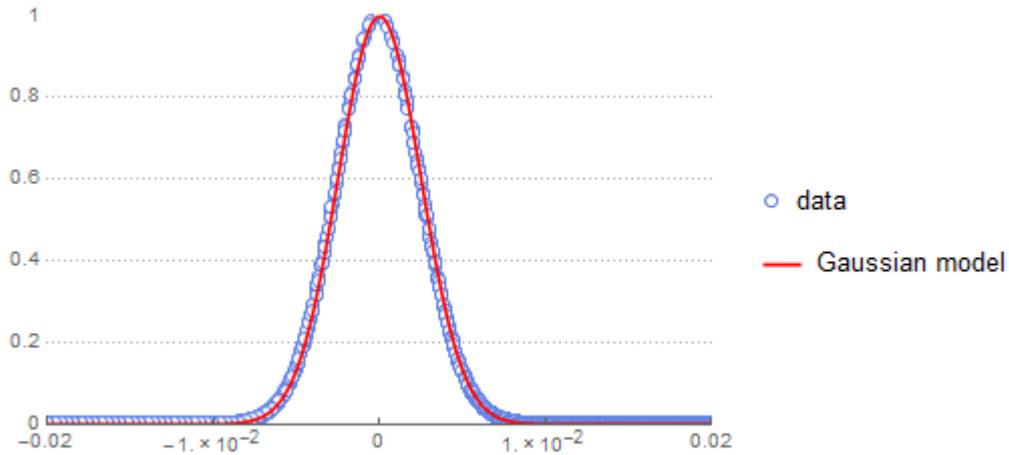


Figura 49: Comparación entre distribución Gaussiana y datos obtenidos

Otra forma de comprobar si el modelo está correctamente implementado es validarlo para un amplio rango de valores de  $\sigma$  y ver que los datos obtenidos tras las simulaciones siguen las funciones de probabilidades correspondientes. Para ello se compara la desviación típica introducida en Tonatiuh con la obtenida en el flujo incidente en el heliostato.

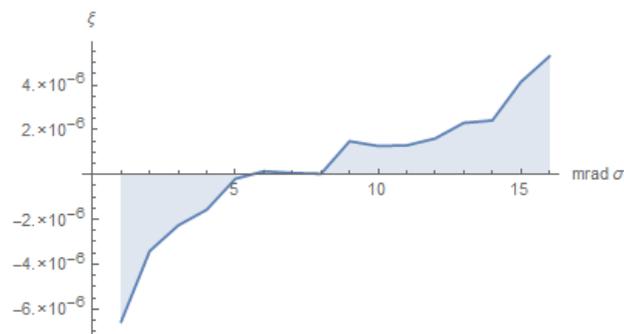


Figura 50: Error entre el  $\sigma$  teórico y el  $\sigma$  obtenido tras simulación

Se puede apreciar que el error obtenido tras las simulaciones en Tonatiuh son despreciables ya que los errores son del 0.0005 % en los casos más desfavorables. El rango de valores de  $\sigma$  es lo suficientemente amplio como para valorar todos los escenarios posibles.

### 7.1.2. Comparación con SolTrace

Para saber hasta qué punto los datos obtenidos son satisfactorios se comparan los resultados obtenidos en Tonatiuh con un programa desarrollado por el National Renewable Energy Laboratory (NREL) llamado SolTrace.

SolTrace es un software para modelizar sistemas CSP y analizar los efectos ópticos [11]. Aunque esta adecuado para aplicaciones solares también puede ser utilizado para cualquier sistema óptico. La geometría óptica se puede modelar como una serie de etapas compuesta de elementos ópticos con ciertos atributos como la forma, el contorno y la calidad óptica. Los resultados se pueden visualizar y guardar como diagramas de dispersión, mapas de flujo y gráficos de rendimiento. Al igual que Tonatiuh el código implementado utiliza el trazador de rayos con el método de Monte Carlo por lo que se puede hacer una comparación entre los dos softwares para comprobar el correcto funcionamiento.

Para comparar el modelo de la distribución del Sol es necesario que los datos de partida sean iguales en los dos Softwares. El sistema introducido para el estudio es lo más sencillo posible [19]. En este caso, el sistema está compuesto por el Sol, un heliostato y un receptor donde se analiza el flujo incidido por el heliostato. Las consideraciones de partida son las siguientes:

- El sol tiene una elevación de  $90^\circ$  y la DNI (direct normal irradiance) es de  $1000 \text{ W/m}^2$ .
- El heliostato está a una distancia de 100 metros respecto del origen
- El receptor está a 10 metros de altura respecto del origen.

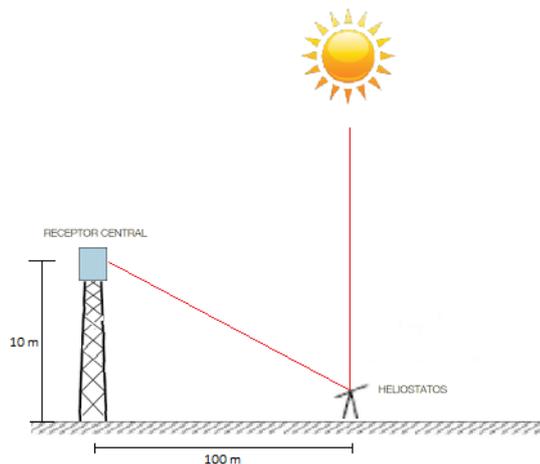


Figura 51: Disposición de los elementos en el sistema

El valor de sigma introducido en el modelo solar es de  $\sigma_{\text{sun}}=2.485 \text{ mrad}$  y un error en la superficie del heliostato de  $\sigma_{\text{slope}}=2 \text{ mrad}$  y  $\sigma_{\text{specular}}=0 \text{ mrad}$ . Son valores típicos en la vida real [26].



A partir de los datos introducidos se simulan tanto en Tonatiuh y en SolTrace obteniendo dos distribuciones Gaussianas similares.

En los resultados obtenidos y tras una valoración rápida la energía captada en el receptor mediante Tonatiuh es de 732.7 W mientras que haciendo la misma simulación en SolTrace la energía captada es de 741.2W. La diferencia de energía captada en el receptor es de 9 vatios entre los dos programas lo que supone un 1% más con SolTrace. Para guardar los datos es necesario que el número de “bins” sea el mismo en las dos plataformas. El perfil de la irradiancia solar sigue una distribución normal- y se aprecia que aproximadamente es parecido.

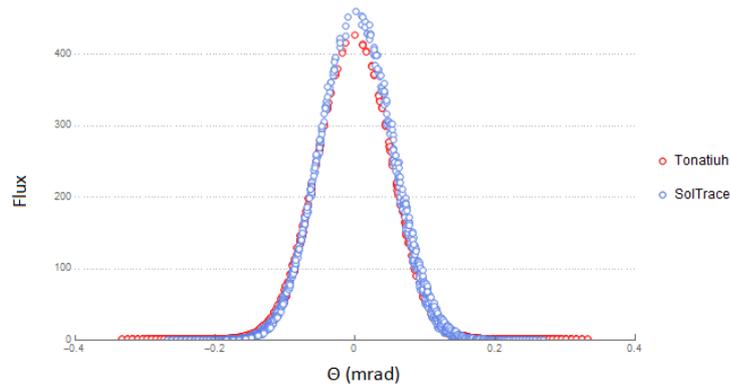


Figura 52: Distribución del flujo Tonatiuh vs SolTrace

Se puede ajustar los datos obtenidos mediante una regresión que sigan sendas distribuciones normales normalizadas respectivamente a partir de la Ec. 40.

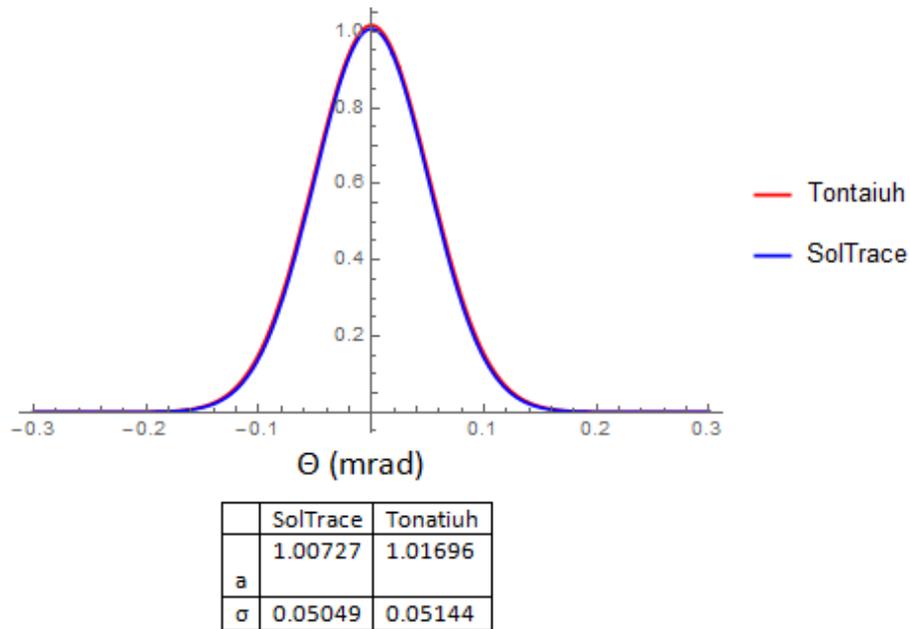


Figura 53: Ajuste de los datos obtenidos a sendas distribuciones normales

Aunque los valores de las funciones de distribución son parecidas es necesario estudiar las muestras de las poblaciones obtenidas para analizar de forma más detallada la distribución

Partimos de que tenemos dos muestras aleatorias de dos poblaciones independientes entre sí pero que deben seguir una distribución normal o al menos ser lo más parecido entre ellas.

Para comprobar que las dos distribuciones normales son similares se hace un test de hipótesis. En este caso tenemos muestras de dos poblaciones con medias diferentes y variancias conocidas ( $N(\mu_1, \sigma_x)$ ,  $N(\mu_2, \sigma_y)$ ) por lo que se aplica el Ztest [26]. Se impone que la hipótesis nula sea  $H_0: \mu_1 - \mu_2 = \Delta_0$  siendo  $\Delta_0$  un valor muy pequeño. A partir de los datos obtenidos tras la simulación se calcula los valores medios de cada población y la desviación estándar ya que siguen una distribución normal. Cuanto mayor sea el número de muestras mejor. En este caso se han cogido 3721 muestras de cada población dando los siguientes valores de partida:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i = 0.002358 \quad \rightarrow \quad \sigma_x = 0.0505 \quad \rightarrow \quad n_1 = 3721$$

$$\bar{Y} = \frac{1}{n} \sum_{j=1}^n y_j = -0.002027 \quad \rightarrow \quad \sigma_y = 0.0514 \quad \rightarrow \quad n_2 = 3721$$

$$H_0: \mu_1 - \mu_2 = \Delta_0 = 0.007$$



Una vez que se han calculado los datos de partida es necesario proponer una hipótesis lógica. La finalidad es asegurar que las dos poblaciones sigan distribuciones normales parecidas:

$$\text{Hipótesis nula} \quad \rightarrow \quad H_0: \mu_1 - \mu_2 = \Delta_0 = 0.007$$

$$\text{Hipótesis alternativa} \quad \rightarrow \quad H_1: \mu_1 - \mu_2 < \Delta_0 = 0.007$$

Se aplica el Test statistic (Ztest) con un nivel de significación del test,  $\alpha=0.05$  ( $Z_\alpha=1.96$ ):

$$Z_0 = \frac{\bar{x} - \bar{y} - \Delta_0}{\sqrt{\frac{\sigma_x^2}{n_1} + \frac{\sigma_y^2}{n_2}}} = - - 2.21 < -Z_\alpha = -1.96$$

Con los resultados obtenidos se descartar la hipótesis nula y se asegura la hipótesis alternativa. Es decir, se asegura que la diferencia de las medias entre las dos distribuciones es menor a 0.007 con una probabilidad al 95%.

Se llega a la conclusión de que el modelo implementado en Tonatiuh se ha hecho correctamente y no existe gran diferencia si se compara con el mismo modelo implementado en SolTrace.

## 7.2. Modelo User defined

### 7.2.1. Validación mediante Tonatiuh

Para la validación de este modelo es necesario introducir la forma solar deseada en una tabla. En esta validación se utiliza unos datos típicos adjuntos al software de SolTrace. Los datos son los siguientes:

$\Theta$ ( mrad)	Intensity	$\Theta$ ( mrad)	Intensity
0	1	6.4	0.321
0.8	0.9952	7.2	0.215
1.6	0.9795	8	0.125
2.4	0.9512	8.8	0.083
3.2	0.9036	9.6	0.026
4	0.8128	10.4	0.0095
4.8	0.6543	11.2	0
5.6	0.456	-	-

Tabla 1: Datos de partida

Tras la simulación se obtiene un diagrama de flujo ben el que se puede observar la distribución de flujo reflejado en la superficie descrita al principio del apartado.

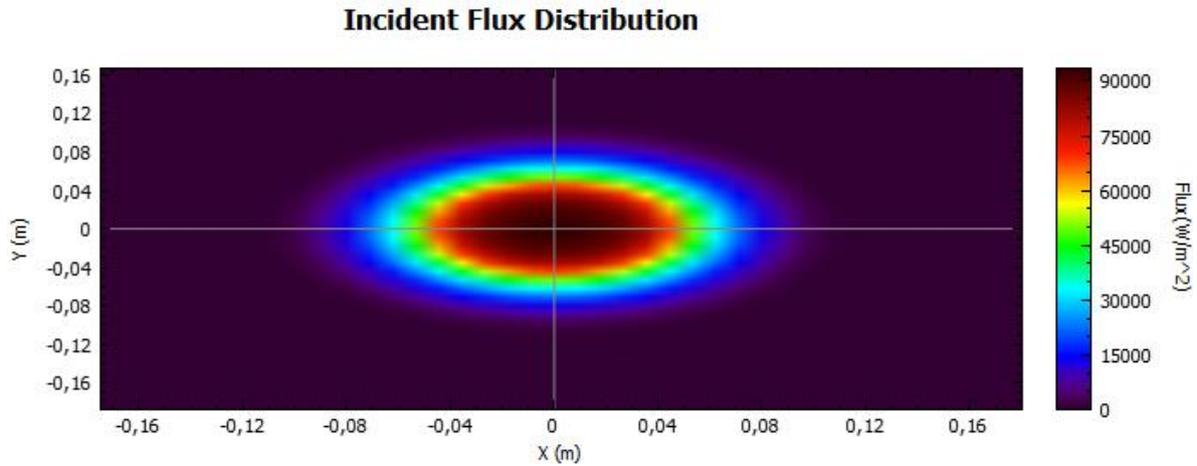


Figura 54: Distribución del flujo sobre un heliostato

Si se analizan los datos preliminares se asegura que la forma solar introducida está centrada en la superficie porque las coordenadas que son  $\{x,y\} \rightarrow \{-6 \cdot 10^{-8}, -0.003\}$ . Para hacer un análisis exhaustivo se guardan en un fichero los puntos de intersección en coordenadas cartesianas para su procesamiento empírico.

Lo primero es comparar visualmente la forma solar introducida por el usuario con los resultados obtenidos tras la simulación. Comparando los valores de intensidad en función del ángulo theta.

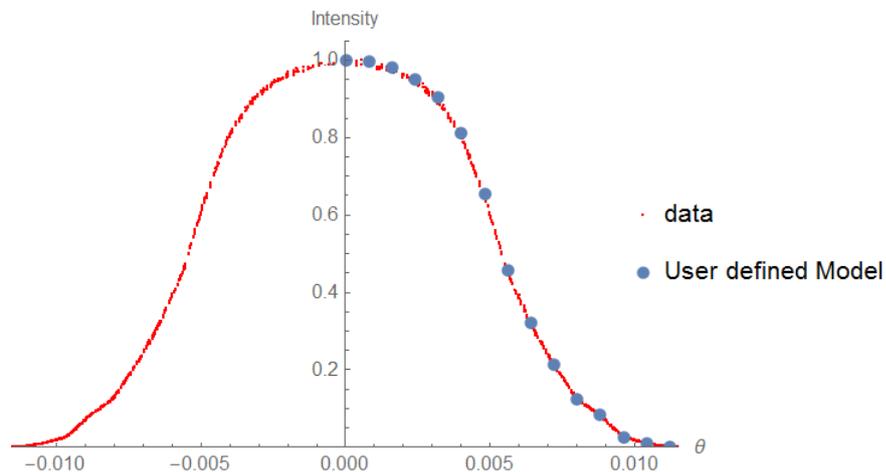


Figura 55. Comparativa datos teóricos y datos reales

Para un análisis más exhaustivo se calcula el error relativo de cada uno de los puntos respecto al valor teórico correspondiente para saber cómo de exacto es la forma solar obtenida respecto a la deseada. Es necesario saber en cada momento en qué tramo se está para que el cálculo se hace de la siguiente manera.

$$\text{Error absoluto} = |\text{Valor medido} - \text{valor real}| \quad \text{Ec. 41}$$

$$\text{Error relativo} = \frac{\text{Error absoluto}}{\text{Valor real}} * 100 \quad \text{Ec. 42}$$

Y los datos obtenidos son los siguientes:

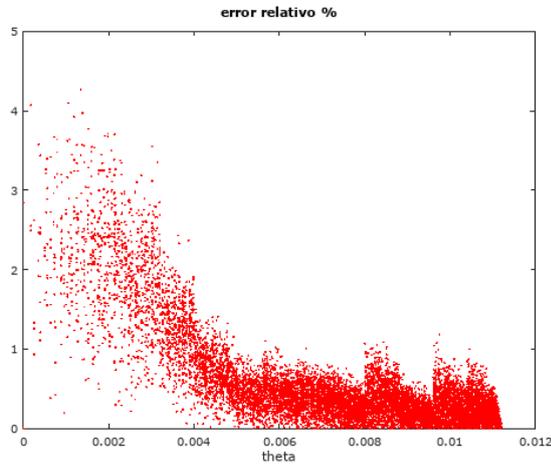


Figura 56: Error relativo en función de theta

Para valores pequeños de theta el error relativo es mayor, el motivo por el que puede deberse son diversos. Por un lado, al programar en c++ es probable que no se hagan los cálculos con todas las cifras significativas. Es decir, que exista un redondeo en los cálculos que se arrastren desde el principio hasta la obtención de los datos finales. Otra de las posibilidades es que sea necesario hacer las simulaciones con un número de rayos mucho mayor al realizado, algo que con la velocidad de procesamiento de los ordenadores convencionales no es suficiente. Por otro lado, cuando se simula con el software de Tonatiuh se simula para un orden de decenas de millones de rayos. Cuando se extraen los datos del programa no se extrae la información de cada uno de los rayos sino que se hace un análisis del flujo.

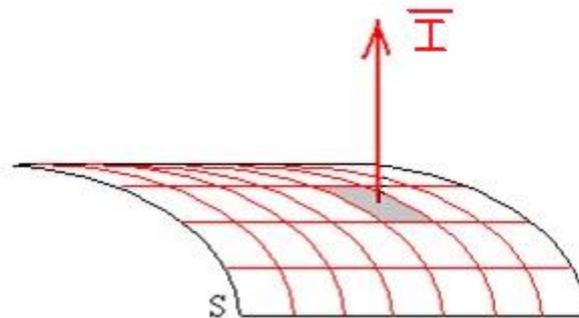


Figura 57: Partición de una malla

-Se obtiene los datos de cada uno de los puntos de una rejilla ficticia por lo que el valor obtenido de intensidad de flujo no tiene por qué coincidir exactamente con las coordenadas sino que será aproximado. Cuanto mayor sea el número de divisiones de la superficie mayor es la precisión pero también mayor el coste computacional del tratamiento de datos.- Con un error máximo del 4.26% para valores de intensidad cercanos a la unidad se considera óptimo.

Para asegurarse de que el programa funciona para una gran variedad de perfiles se simulan para diferentes formas solares inventados por el propio autor pero basándose en magnitudes empíricas. De esta forma se puede comprobar escenarios que tengan peculiaridades.

Uno de esos escenarios es suponer que la forma solar en vez de descender de forma progresiva conforme aumenta el ángulo theta lo haga de forma brusca:

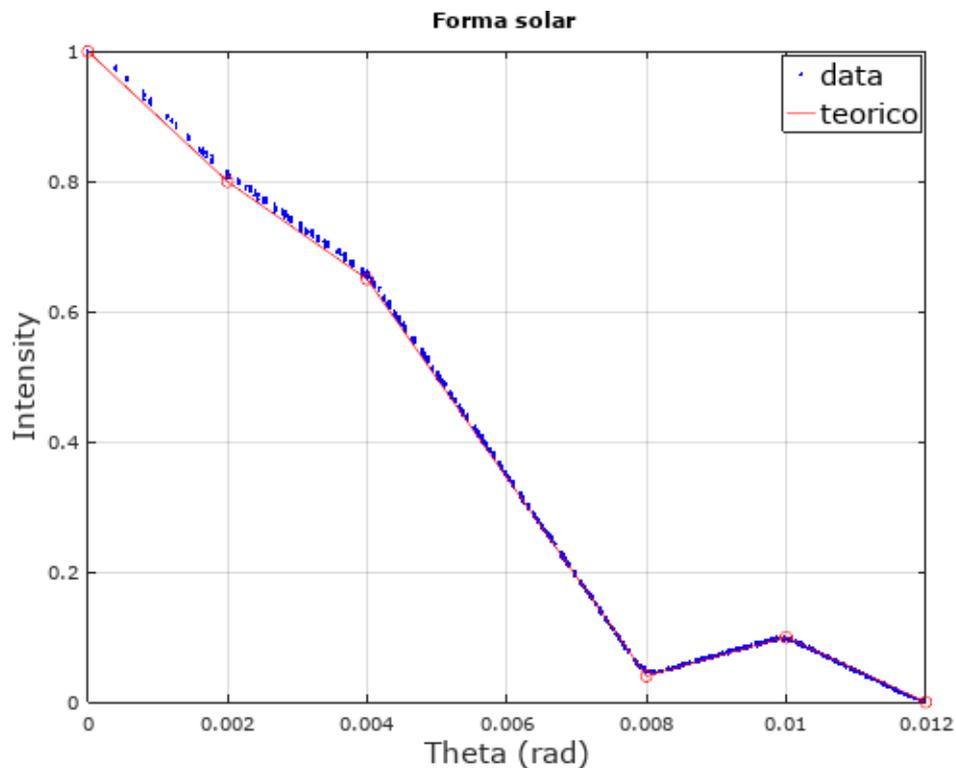


Figura 58: Escenario alternativo

A partir de un escenario posible como es el de la Figura 58- se puede analizar cómo evolucionan los errores respecto a la forma solar inicial.

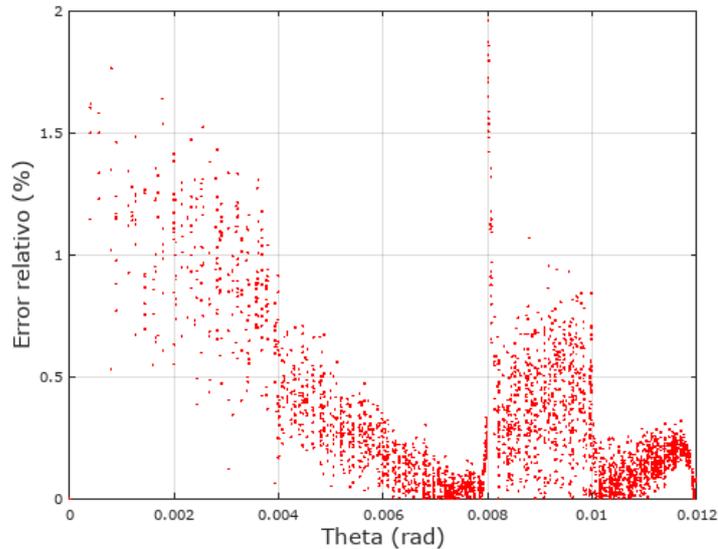


Figura 59:- Errores relativos correspondiente a un escenario relativo

En este caso se aprecia que cuanto menor es el valor de theta peor sigue el programa a la referencia como ocurría en el escenario anterior (ver Figura 56). La novedad que ocurre en este caso es que en los thetas donde hay un cambio de tendencia como por ejemplo en  $\theta = 0.008$  rad el modelo genera valores que difieren un poco del ideal. Aun así los valores que se obtienen no son altos por lo que se considera que el modelo es óptimo para la simulación de diferentes formas solares.

### 7.2.2. Comparación con SolTrace

En este caso también se va a comparar los resultados obtenidos con Tonatiuh y SolTrace. El escenario recreado es el mismo que en el apartado 7.1.2 pero en este caso la forma solar es diferente ya que no se compone de una distribución Gaussiana sino que se aplica la forma solar introducida en la Tabla 1. Para recordar un poco, los datos de partida son los siguientes:

- El sol tiene una elevación de  $90^\circ$  y la irradiancia es de  $1000 \text{ W/m}^2$ .
- El heliostato está a una distancia de 100 metros respecto del origen
- El receptor está a 10 metros de altura respecto del origen.
- El error en la superficie del heliostato de  $\sigma_{\text{slope}} = 2$  mrad

A partir de las simulaciones tanto en Tonatiuh como en SolTrace la radiación total que llega al receptor es de  $776.769 \text{ W}$  si se simula con Tonatiuh y de  $744.18 \text{ W}$  con SolTrace. La diferencia es de un 4%. Al simularse con un solo heliostato la diferencia puede ser considerable pero normalmente se trabaja con campos de heliostatos grandes (del orden de cientos de espejos) por lo que se minimiza el error.

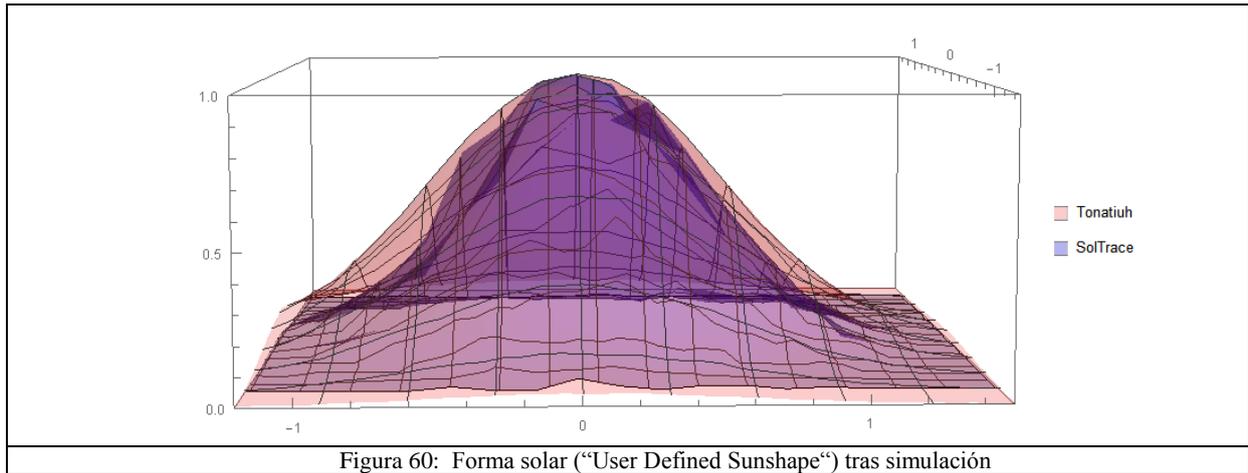


Figura 60: Forma solar (“User Defined Sunshape”) tras simulación

En la Figura 60 se aprecian las formas solares obtenidas mediante los dos programas. La distribución del flujo es mayor utilizando Tonatiuh además de ser más homogéneo que SolTrace. Por otra parte, el centroide de la forma solar con Tonatiuh está más centrado de las coordenadas (0,0) que sería en el caso ideal.

Para finalizar se muestra en la- Figura 61 una pequeña comparativa entre la forma solar obtenida tanto con Tonatiuh y con SolTrace:

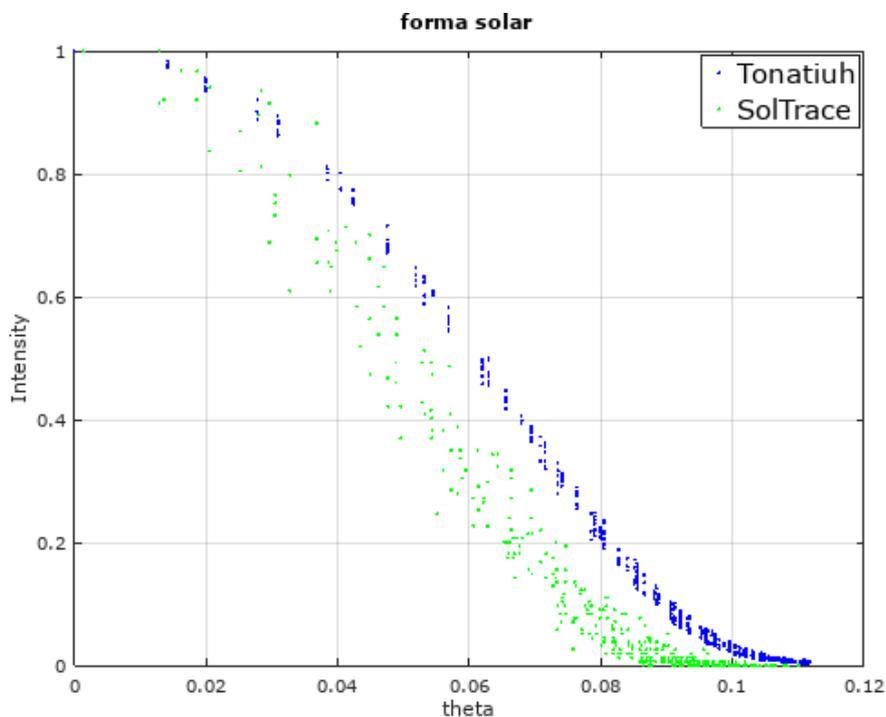


Figura 61: Comparativa misma forma solar y diferentes herramientas



Se observa que existe una pequeña diferencia entre las dos formas solares. Esto ocurre por cómo está implementado los códigos en ambos programas y también por el número de rayos generados en cada una de las simulaciones



## 8. Bibliografía

- [1] REN21, *Renewables 2016 Global Status Report*. 2016.
- [2] M. Mehos *et al.*, “Concentrating Solar Power Gen3 Demonstration Roadmap,” *Nrel/Tp-5500-67464*, no. January, pp. 1–140, 2017.
- [3] P. K. Chang, “No Title.”
- [4] L. Turchetti, “General introduction to CSP and CST technologies The dawn of concentrating solar thermal energy.”
- [5] N. B. Devi, “Chapter 1 Introduction,” 2013.
- [6] T. Nature and S. Radiation, “Solar Radiation,” *Sol. Eng. Therm. Process.*, pp. 3–42, 2013.
- [7] F. M. Montagnino, “Linear Fresnel Collectors Dish Collecors About us.”
- [8] J. a. Duffie, W. a. Beckman, and W. M. Worek, *Solar Engineering of Thermal Processes, 4nd ed.*, vol. 116. 2003.
- [9] S. Wilbert, R. Pitz-Paal, and E. Guillot, “Database with frequencies of circumsolar ratio (CSR) occurrence at different sites and correlations of CSR with DNI and DHI data,” pp. 1–31, 2013.
- [10] “SunshapeDistribution.” .
- [11] T. Wendelin, “SolTRACE: A New Optical Modeling Tool for Concentrating Solar Optics,” *Sol. Energy*, no. October, pp. 253–260, 2003.
- [12] D. Buie, A. G. Monger, and C. J. Dey, “Sunshape distributions for terrestrial solar simulations,” *Sol. Energy*, vol. 74, no. 2, pp. 113–122, 2003.
- [13] “Allen-AstrophysicalQuantities[2] Copy.pdf.” .
- [14] I. T. E. Limb-darkening, “1 Chapter 6 Limb Darkening 6.1,” no. 0, pp. 1–8, 2000.



- [15] S. Wilbert, “Determination of Circumsolar Radiation and its Effect on Concentrating Solar Power,” *Maschinenwesen*, vol. Doktor der, p. 177, 2014.
- [16] M. Izygon, C. Nilsson, N. Vu, and P. Armstrong, “GPU-based Monte Carlo Ray Tracing Simulation For Solar Power Plants (slides),” 2012.
- [17] P. Garcia, A. Ferriere, and J. J. Bezia, “Codes for solar flux calculation dedicated to central receiver system applications: A comparative review,” *Sol. Energy*, vol. 82, no. 3, pp. 189–197, 2008.
- [18] L. C. Barquín, “Análisis De Sensibilidad De Parámetros Radiantes En Un Captador De Concentración Cilindro-Parabólico,” pp. 1–61, 2015.
- [19] C. Caliot, “Optical modelling in CSP,” 2016.
- [20] E. V Votyakov, “Optics and Modelling in CST,” no. November, 2016.
- [21] C. He, J. Feng, and Y. Zhao, “Fast flux density distribution simulation of central receiver system on GPU,” *Sol. Energy*, vol. 144, pp. 424–435, 2017.
- [22] V. Romero, “Capítulo 3 . Códigos de simulación para el cálculo de distribución de flujo en receptor y factor de desbordamiento .,” pp. 55–76.
- [23] A. J. Gallego and E. F. Camacho, “On the optimization of flux distribution with flat receivers: A distributed approach,” *Sol. Energy*, vol. 160, no. February 2017, pp. 117–129, 2018.
- [24] S. M. Besarati, D. Yogi Goswami, and E. K. Stefanakos, “Optimal heliostat aiming strategy for uniform distribution of heat flux on the receiver of a solar power tower plant,” *Energy Convers. Manag.*, vol. 84, pp. 234–243, 2014.
- [25] A. L. Cauchy and J. Fou-, “8.1. Introducción.”
- [26] B. Cholesterol, H. Body, and A. Exercises, *Index of Applications in Examples and Exercises.* .



## ANEXOS

En el presente apartado se exponen los códigos en c++ que se han implementado en los dos modelos solares para su correcto funcionamiento:

La forma de presentarlos es la siguiente. Los códigos h. (header) son cabeceras y en él se definen todo tipo de variables, vectores, objetos, clases, etc., necesarios. En los códigos cpp. se implementan los algoritmos necesarios para el correcto funcionamiento. Existen más líneas de código pero no le doy tanta importancia, aunque esenciales para el correcto funcionamiento, por ser para la generación de tablas que luego adquirirán los datos introducidos por el usuario en el User defined Sunshape o crear los iconos de los modelos.



## Gaussian codes

- Gaussian h.

```

#ifndef SUNSHAPEGAUSSIAN_H_
#define SUNSHAPEGAUSSIAN_H_

#include "TSunShape.h"
#include "trt.h"
#include <Inventor/sensors/SoFieldSensor.h>

class SoSensor;

class SunshapeGaussian : public TSunShape
{
    SO_NODE_HEADER(SunshapeGaussian);

public:
    SunshapeGaussian( );
    static void initClass();
    SoNode* copy( SbBool copyConnections ) const;

    //Sunshape Interface
    void GenerateRayDirection( Vector3D& direction, RandomDeviate& rand) const;
    double GetIrradiance() const;
    double GetThetaMax() const;

    trt::TONATIUH_REAL irradiance;
    trt::TONATIUH_REAL sigma;

protected:
    static void updatesigma(void *data, SoSensor *);
    ~SunshapeGaussian();
private:

    SoFieldSensor*m_irradianceValuesSensor;
    SoFieldSensor*m_sigmaValuesSensor;

    double pdfTheta( double theta ) const;

    double zenithAngle( RandomDeviate& rand ) const;
    void updateState( double sigmaValue );
    double pMax();
    double findMax();
    double m_sigma;
    double m_pMax;
    double m_thetamax;
    double m_heightRectangle;
    static const double m_minsigmaValue;// = 0.001;
    static const double m_maxsigmaValue;// = 1.57;
};

#endif /* SUNSHAPEGAUSSIAN_H_ */

```

- Gaussian cpp.

```
#include <Inventor/sensors/SoFieldSensor.h>
```

```
#include "gc.h"
```

```
#include "SunshapeGaussian.h"
```

```
const double SunshapeGaussian::m_minsigmaValue = 0.000001;
```

```
const double SunshapeGaussian::m_maxsigmaValue = 0.2;
```

```
SO_NODE_SOURCE(SunshapeGaussian);
```

```
void SunshapeGaussian::initClass()
```

```
{  
  SO_NODE_INIT_CLASS(SunshapeGaussian, TSunShape, "TSunShape");  
}
```

```
SunshapeGaussian::SunshapeGaussian( )
```

```
{  
  SO_NODE_CONSTRUCTOR( SunshapeGaussian );  
  SO_NODE_ADD_FIELD( irradiance, ( 1000 ) );  
  SO_NODE_ADD_FIELD( sigma , ( 0.00465 ) );
```

```
  m_irradianceValuesSensor = new SoFieldSensor( updatesigma, this );  
  m_irradianceValuesSensor->setPriority( 1 );  
  m_irradianceValuesSensor->attach( &irradiance );
```

```
  m_sigmaValuesSensor = new SoFieldSensor( updatesigma, this );  
  m_sigmaValuesSensor->setPriority( 1 );  
  m_sigmaValuesSensor->attach( &sigma );
```

```
  double sigmaValue = sigma.getValue();  
  if( sigmaValue >= m_minsigmaValue && sigmaValue <= m_maxsigmaValue ) updateState( sigmaValue
```

```
}
```

```
SunshapeGaussian::~~SunshapeGaussian( )
```

```
{  
  delete m_irradianceValuesSensor;  
  delete m_sigmaValuesSensor;  
}
```

```
void SunshapeGaussian::updateState(double sigmaValue)
```

```
{  
  if( sigmaValue != m_sigma )  
  {  
    m_sigma=sigmaValue;  
    m_thetamax=5*m_sigma;  
    m_pMax = pMax();  
    m_heightRectangle = 1.001 * pdfTheta(findMax());  
  }  
}
```

```
double SunshapeGaussian::pMax()
```

```
{  
  double integral = 0.0;  
  
  double theta = 0;  
  double delta = 0.00001;  
  while( theta < m_thetamax )  
  {  
    integral += exp(-theta*theta/(2*m_sigma*m_sigma))*sin(theta)*delta ;  
    theta += delta;  
  }  
  return 1/integral;  
}
```

```

double SunshapeGaussian::findMax()
{
    //Método Newton-raphson  $x_{n+1}=n-f(x_n)/f'(x_n)$ 
    //Buscar el valor de theta para el pdfmax
    double xn1;
    double xn=m_sigma;
    double error=1e10;
    while ( error>0.0000001)
    {
        xn1=xn-(m_sigma*m_sigma*cos(xn)-xn*sin(xn))/(-xn*cos(xn)-sin(xn)-m_sigma*m_sigma*sin(xn));
        error=abs(xn1-xn);
        xn=xn1;
    }

    return xn1;
}

//Light Interface
void SunshapeGaussian::GenerateRayDirection( Vector3D& direction, RandomDeviate& rand ) const
{
    double phi = gc::TwoPi * rand.RandomDouble();
    double theta = zenithAngle( rand );
    double sinTheta = sin( theta );
double cosTheta = cos( theta );
    double cosPhi = cos( phi );
    double sinPhi = sin( phi );

    direction.x = sinTheta*sinPhi;
    direction.y = -cosTheta;
    direction.z = sinTheta*cosPhi;
}

double SunshapeGaussian::GetIrradiance( void ) const
{
    return irradiance.getValue();
}

double SunshapeGaussian::GetThetaMax() const
{
    return 0;
}

SoNode* SunshapeGaussian::copy( SbBool copyConnections ) const
{
    // Use the standard version of the copy method to create
    // a copy of this instance, including its field data
    SunshapeGaussian* newSunShape = dynamic_cast< SunshapeGaussian* >( SoNode::copy( copyConnections ) )

    newSunShape->irradiance = irradiance;
    newSunShape->sigma = sigma ;
    newSunShape->m_heightRectangle = m_heightRectangle;

    return newSunShape;
}

void SunshapeGaussian::updatesigma(void *data, SoSensor *)
{
    SunshapeGaussian* sunshape = ( SunshapeGaussian* ) data;
    double sigmaValue = sunshape->sigma.getValue();
    if( sigmaValue >= m_minsigmaValue && sigmaValue <= m_maxsigmaValue )
    sunshape->updateState( sigmaValue );
}

```

```

double SunshapeGaussian::zenithAngle( RandomDeviate& rand ) const
{
    double theta;
    double value;

    //Aceptation-rejection method
    do
    {
        theta = rand.RandomDouble() * m_thetamax; //rand.RandomDouble() da valores entre 0-1
        value=rand.RandomDouble()*m_heightRectangle;
    }
    while ( value > pdfTheta( theta ) );

    return theta;
}

double SunshapeGaussian::pdfTheta( double theta ) const
{
    return exp(-(theta*theta)/(2*m_sigma*m_sigma))*sin(theta)*m_pMax;
}

```



## User defined codes

- Userdefined.h

```
#ifndef SUNSHAPEUSERDEFINED_H_
#define SUNSHAPEUSERDEFINED_H_

#include "TSunShape.h"
#include "trt.h"
#include <vector>
#include <Inventor/sensors/SoFieldSensor.h>
#include "MFVecIntensity.h"

class SoSensor;

class SunshapeUserDefined : public TSunShape
{
    SO_NODE_HEADER(SunshapeUserDefined);

public:
    SunshapeUserDefined( );
    static void initClass();
    SoNode* copy( SbBool copyConnections ) const;
    MFVecIntensity intensityValues;

    //Sunshape Interface
    void GenerateRayDirection( Vector3D& direction, RandomDeviate& rand) const;
    double GetIrradiance() const;
    double GetThetaMax() const;

    trt::TONATIUH_REAL irradiance;
    trt::TONATIUH_REAL sigma;

protected:
    static void updatetheta(void *data, SoSensor *);
    ~SunshapeUserDefined();

    double OutputPropertyValue( std::vector< double > thetalist, std::vector< double > intensityList,
double incidenceAngle ) const;

    static void updateIntensity( void* data, SoSensor* );

private:
    SoFieldSensor*m_intensityValuesSensor;

    double pdfTheta( double theta ) const;
    double zenithAngle( RandomDeviate& rand ) const;
    void updateState( );
    double pMax();
    double findMax();
    void parameters( );
    double pdfmax();
};
```

```

std::vector< double > m_theta;
std::vector< double > m_intensity;
std::vector< double > m_m;
std::vector< double > m_n;
std::vector< double > m_c;

double m_pMax;
double m_thetamax;
double m_heightRectangle;

};

#endif /* SUNSHAPEUSERDEFINED_H_ */

```

- Userdefined.cpp

```

#include <Inventor/sensors/SoFieldSensor.h>
#include "gc.h"
#include "SunshapeUserDefined.h"
#include <algorithm>
#include <QMessageBox>
#include <QString>
#include <Inventor/lists/SoFieldList.h>
#include <Inventor/fields/SoFieldContainer.h>
#include "trt.h"
#include "RandomDeviante.h"
#include "Ray.h"
#include "tgf.h"
#include "Transform.h"
#include <QString>
#include <QFile>
#include <QTextStream>

SO_NODE_SOURCE(SunshapeUserDefined);

void SunshapeUserDefined::initClass()
{
    SO_NODE_INIT_CLASS(SunshapeUserDefined, TSunShape, "TSunShape");
}

SunshapeUserDefined::SunshapeUserDefined( )
{
    SO_NODE_CONSTRUCTOR( SunshapeUserDefined );
    SO_NODE_ADD_FIELD( irradiance, ( 1000 ) );
    SO_NODE_ADD_FIELD( intensityValues, ( 0.0f, 0.0f ) );
    intensityValues.SetNames( QObject::tr("Angle [rad]" ), QObject::tr( "Intensity[0-1]" ) );

    m_intensityValuesSensor = new SoFieldSensor( updateIntensity, this );
    m_intensityValuesSensor->setPriority( 1 );
    m_intensityValuesSensor->attach( &intensityValues );

    m_theta.push_back(0);
    m_intensity.push_back(1);
}

double SunshapeUserDefined::GetIrradiance() const
{
    return irradiance.getValue();
}

```

```

double SunshapeUserDefined::GetThetaMax() const
{
    return m_thetamax;
}

double SunshapeUserDefined::OutputPropertyValue( std::vector< double > thetalist, std::vector<
double > intensitylist, double incidenceAngle ) const
{
    // Size incidenceAnglesList vector
    int m = thetalist.size();

    //Classify element's incident Angle lowest to biggest
    for (int i = 0; i < (m - 1); i++)
    {
        for (int j = i + 1; j < m; j++)
        {
            if (thetalist[j] < thetalist[i])
            {
                double aux1 = thetalist[j];
                double aux2 = intensitylist[j];
                thetalist[j] = thetalist[i];
                intensitylist[j] = intensitylist[i];
                thetalist[i] = aux1;
                intensitylist[i] = aux2;
            }
        }
    }
}

void SunshapeUserDefined::updateIntensity( void* data, SoSensor* )
{
    SunshapeUserDefined* sun = static_cast<SunshapeUserDefined* >( data );

    std::vector< double > oldtheta2 = sun->m_theta;
    std::vector< double > oldintensity2 = sun->m_intensity;

    int numberOfValues = sun->intensityValues.getNum();

    sun->m_theta.clear();
    sun->m_intensity.clear();
    for( int i = 0; i < numberOfValues; i++ )
    {
        sun->m_theta.push_back( sun->intensityValues[i][0] );
        sun->m_intensity.push_back( sun->intensityValues[i][1] );
    }
    sun->updateState();
}

void SunshapeUserDefined::updateState()
{
    parameters( );
    m_thetamax=m_theta.back();//theta máxima
    m_pMax = pMax();
    m_heightRectangle = 1.001 * pdfmax();
}

void SunshapeUserDefined::parameters( )
{
    for (unsigned int i = 0; i < m_theta.size()-1;i++)
    {
        m_m.push_back((m_intensity[i]-m_intensity[i+1])/(m_theta[i]-m_theta[i+1]));
        m_n.push_back(m_intensity[i]-m_m[i]*m_theta[i]);
    }
}

```

```

double SunshapeUserDefined::pMax()
{
    double integral = 0.0;
    for (unsigned int i = 0; i < m_theta.size()-1; i++)
    {
        if( i<1 )
        {
            m_c.push_back( m_theta[i]+(m_n[i]+m_m[i]*m_theta[i])*cos(m_theta[i])-m_m[i]*sin(m_theta[i]) );
        }
        else
        {
            m_c.push_back( m_c[i-1]+(m_n[i]-m_n[i-1]-m_m[i-1]*m_theta[i]+m_m[i]*m_theta[i])*cos(m_theta[i])+(m_m[i-1]-m_m[i])*sin(m_theta[i]) );
        }
        integral += (m_c[i]-m_n[i]*cos(m_theta[i+1])-m_m[i]*m_theta[i+1]*cos(m_theta[i+1])+m_m[i]*sin(m_theta[i+1]))-(m_c[i]-m_n[i]*cos(m_theta[i])-m_m[i]*m_theta[i]*cos(m_theta[i])+m_m[i]*sin(m_theta[i])));
    }
    return 1/integral;
}

double SunshapeUserDefined::pdfmax()
{
    //Método Newton-raphson  xn+1=n-f(xn)/f'(xn)
    double fdensitymax=0;
    double xmax1=0;
    double xmax2=0;
    double fdensitymax1,fdensitymax2;
    double xmax;
    std::vector< std::pair< double, double > > localMax;
    for (unsigned int i =0; i < m_theta.size()-1; i++)
    {
        double xn;
        double xn1;
        double producto =
(m_pMax*((m_n[i]+m_m[i]*m_theta[i])*cos(m_theta[i])+m_m[i]*sin(m_theta[i]))*(m_pMax*((m_n[i]+m_m[i]*
m_theta[i+1])*cos(m_theta[i+1])+m_m[i]*sin(m_theta[i+1]))));
        if( producto < 0 )
        {
            xn = (m_theta[i]+m_theta[i+1])/2;
            double error = gc::Infinity;//1e10;//HUGE_VAL;
            while( error > 0.00000001)
            {
                double fderivada = m_pMax*((m_n[i]+m_m[i]*xn)*cos(xn)+m_m[i]*sin(xn));
                double fderivadaderivada = m_pMax*(2*m_m[i]*cos(xn)-(m_n[i]+m_m[i]*xn)*sin(xn));
                xn1 = xn-fderivada/fderivadaderivada;
                error=fabs(xn1-xn);
                xn=xn1;
            }
            localMax.push_back(std::make_pair( m_pMax*(m_n[i]+m_m[i]*xn)*sin(xn), xn));
        }
    }
    for(unsigned int i = 0; i < localMax.size(); i++ )
        std::sort( localMax.begin(), localMax.end());
    return localMax.back().first;
}

```

```
SunshapeUserDefined::~SunshapeUserDefined()
```

```
{  
  delete m_intensityValuesSensor;  
}
```

```
//Light Interface
```

```
void SunshapeUserDefined::GenerateRayDirection( Vector3D& direction, RandomDeviate& rand ) const
```

```
{  
  double phi = gc::TwoPi * rand.RandomDouble();  
  double theta = zenithAngle( rand );  
  double sinTheta = sin( theta );  
  double cosTheta = cos( theta );  
  double cosPhi = cos( phi );  
  double sinPhi = sin( phi );  
  
  direction.x = sinTheta*sinPhi;  
  direction.y = -cosTheta;  
  direction.z = sinTheta*cosPhi;  
}
```

```
SoNode* SunshapeUserDefined::copy( SbBool copyConnections ) const
```

```
{  
  // Use the standard version of the copy method to create  
  // a copy of this instance, including its field data  
  SunshapeUserDefined* newSunShape = dynamic_cast< SunshapeUserDefined* >( SoNode::copy(  
copyConnections));
```

```
  newSunShape->m_theta = m_theta;  
  newSunShape->m_intensity = m_intensity;  
  newSunShape->m_m = m_m;  
  newSunShape->m_n = m_n;  
  newSunShape->m_c = m_c;  
  newSunShape->m_pMax = m_pMax;  
  newSunShape->m_thetamax = m_thetamax;  
  newSunShape->m_heightRectangle = m_heightRectangle;
```

```
  return newSunShape;
```

```
}  
double SunshapeUserDefined::zenithAngle( RandomDeviate& rand ) const
```

```
{  
  double theta;  
  double value;  
  
  do  
  {  
    theta = rand.RandomDouble() * m_thetamax; //rand.RandomDouble() da valores entre 0-1  
    value=rand.RandomDouble()*m_heightRectangle;  
  } while ( value > pdfTheta(theta) );  
}
```

```
return theta;
```

```
}
```

```
double SunshapeUserDefined::pdfTheta( double theta ) const
```

```
{  
  double prob = 0;  
  for (unsigned int i =0; i < m_theta.size()-1; i++)  
  {  
    if ((theta >= m_theta[i]) && (theta< m_theta[i+1]))  
    {  
      prob = m_pMax*(m_n[i]+m_m[i]*theta)*sin(theta);  
      break;  
    }  
  }  
  return prob;  
}
```