Verification of Optical Modelling of Sunshape and Surface Slope Error for Concentrating Solar Power Systems

Ye Wang, Daniel Potter, Charles-Alexis Asselineau, Clotilde Corsi, Michael Wagner, Cyril Caliot, Benjamin Piaud, Manuel Blanco, Jin-Soo Kim, John Pye

Verification of Optical Modelling of Sunshape and Surface Slope Error for Concentrating Solar Power Systems

Ye Wang[1], Daniel Potter[2], Charles-Alexis Asselineau[1], Clotilde Corsi[2], Michael Wagner[3], Cyril Caliot[4], Benjamin Piaud[5], Manuel Blanco[6], Jin-Soo Kim[2] and John Pye[1*]

[1]*Research School of Electrical, Energy and Materials Engineering, Australian National University, Canberra, Australia*
[2]*CSIRO Energy, Newcastle, Australia*
[3]*National Renewable Energy Laboratory, Golden, CO, USA*
[4]*Processes, Materials and Solar Energy Laboratory, PROMES, CNRS, Font-Romeu-Odeillo, France*
[5]*Méso-Star, Toulouse, France*
[6]*Energy, Environment and Water Research Center (EEWRC), The Cyprus Institute, Nicosia, Cyprus*

Abstract

Sunshape and reflector surface slope error distributions are significant elements in modelling the optical behaviour of a concentrating solar power system. Different optical modelling tools implement these elements with various approaches. Discrepancies can easily accumulate in simulations of a large optical system as a result of incorrect implementations. This study reviews and verifies the implementations of these two factors in six tools that are widely used for optical modelling in solar energy research: Tonatiuh, SolTrace, Tracer, Solstice, Heliosim and SolarPILOT. The review incorporates three rounds of tests. Firstly, basic tests examine each factor carefully in simplified on-axis reflector–target configurations (round ‘A’). Secondly, off-axis effects are introduced (round ‘B’). Thirdly, full heliostat field simulations are verified (round ‘C’). All of the test cases are simulated with each modelling tool, and results are compared. Discrepancies were observed due to approximations inherent in the cone optics (convolution) methods, incorrect implementation the of pillbox slope errors, different approaches to setting the circumsolar ratio for the Buie sunshape, and different approaches to the calculation of blocking and shading losses in some tools. All issues are discussed fully, and solutions to most issues were implemented within the scope of the present study. Some remaining issues are noted. The study highlights the importance of careful implementation of these aspects of optical modelling and contributes to an improvement in the quality of several widely-used tools.

Keywords

Optical modelling; verification; sunshape; surface slope error; Monte Carlo ray tracing; cone optics.

1. Introduction

The reflector/concentrator, together with the receiver, constitute the optical system at the front end of a concentrating solar power (CSP) system, which accounts for around 30–50% of the total capital cost (Buck, 2012). Designing a highly efficient optical system and operating it in a safe manner are crucial in CSP applications, whether it be in parabolic dishes, trough systems, or central tower

* Corresponding author: john.pye@anu.edu.au.

systems. Optical modelling is commonly used to assist these activities. There are two categories of optical modelling methods: (1) Monte Carlo ray tracing (MCRT) (e.g. MIRVAL (Leary and Hankins, 1979), Tonatiuh (Blanco et al., 2009), SolTrace (Wendelin, 2003), Tracer (Wang et al., 2016), Solstice (Caliot et al., 2015) and Heliosim (Potter et al., 2017)), and (2) cone optics convolution-based method, such as UHC/RCELL (Lipps and Vant-Hull, 1978), DELSOL (Dellin and Fish, 1979), HELIOS (Vittitoe and Biggs, 1981), HFLCAL (Schwarzbözl et al., 2009) and SolarPILOT (Wagner and Wendelin, 2018). MCRT modelling tools can be further categorised into those that irradiate rays from a plane above reflectors (e.g. Tracer, Tonatiuh and SolTrace), and those that irradiate rays directly from the reflector surfaces (e.g. Solstice and Heliosim). Several review papers (Garcia et al., 2008; Ho, 2008; Li et al., 2016; Levêque et al., 2017) have thoroughly summarised the features of the techniques.

The sun is not a point source but appears as a 'disk' when viewed from the earth. Realistic concentrator surfaces deviate from the design shape due to material stress, gravity and wind effects, or manufacturing errors (Rabl, 1985). The impacts of sunshape and surface slope errors are critical at the design stage as they directly affect the incoming and reflected solar radiation and contribute to the image spread on the receiver.

Various implementation methods for these two factors can be found in different optical modelling tools. Besides the existing optical modelling tools listed above, some research groups develop their own optical modelling codes so as to have freedom and a controllable platform for the optical analyses of solar concentrators, receiver designs and for integration with other system modelling tools. Specific evaluation and verification of the implementation of the physical relations in simulation codes are not commonly found in the literature. Even though some validations against experimental data or comparisons against other optical modelling tools were published (Blanco et al., 2009, 2010, 2011; Yellowhair et al., 2014), the data is not available in a form that enables the validation of more recent tools or methods comprehensively. Besides, experimental validations are expensive and difficult as it is very challenging to control, isolate or measure many real-world factors influencing the results, e.g. canting, tracking, surface slope error, weather conditions and measurement errors.

In this paper, a thorough comparison of the results obtained from simulations of heliostat field optics with six well-known optical modelling tools is presented, with the emphasis on checking the implementation and accuracy of sunshape and slope error simulations. The main features of the tests and the selective results are discussed in this paper. The detailed parameters and results are available in the supplementary material for readers interested in further verification. The data and models can also be accessed via the Github repository[1] maintained by the Australian National University (ANU) Solar Thermal Group (STG). This study has contributed to an improvement in the quality of these six tools. We hope it would also ensure better agreement and build confidence amongst CSP researchers on accurate modelling of the optical behaviour of solar concentrators.

2. Tools and Method

2.1 Tools

1 Data access: https://github.com/anustg/optics-verification

The six optical modelling tools selected for this study, Tonatiuh, Tracer, Solstice, Heliosim, SolTrace and SolarPILOT, are widely used in the solar research community. They are briefly reviewed in this section. These tools use a variety of methods for optical modelling. All of them except Heliosim are open source codes. For a list of a wider range of tools, beyond those considered in this study, see Li et al., 2016.

2.1.1 Tonatiuh

Tonatiuh is a C++ multi-threading open source Monte Carlo ray tracer package for optical modelling of all types of solar concentrators both reflective and refractive, jointly developed by the National Renewable Energy Centre of Spain (CENER) and the University of Texas at Brownsville (UTB) with the support of the National Renewable Energy Laboratory (NREL) (Blanco et al., 2009). A preliminary comparison against SolTrace was conducted in 2009 via three simulation scenarios (a parabolic dish, a parabolic trough and a solar furnace system with pillbox sunshape). The maximum difference was under 3% (Blanco et al., 2009). It was also validated with another two experimental data sets. The first was using the data gathered at CIEMAT's Plataforma Solar de Almería (PSA), although it was difficult to draw definite conclusions about the accuracy of flux estimation due to the lack of sunshape's circumsolar ratio and surface reflectivity of the secondary concentrator (Blanco et al., 2010). The second was validated at the Mini-Pegase CNRS-PROMES facility, and a high level of similarity between the measured flux distributions and those calculated by Tonatiuh was observed (Blanco et al., 2011). Having been under development since 2004, Tonatiuh has a vast number of features to facilitate the modelling of any kind of solar concentrators, such as wizards that make it possible to very easily define solar tower systems with thousands of heliostats. Its plugin-based architecture also makes it easy to expand the program to incorporate new types of surfaces, materials or solar radiation models. It also incorporates scripting capabilities that make it easy to automate the running the program to achieve a large variety of purposes.

2.1.2 Tracer

Tracer is an open source package implemented in Python, and utilising efficient numerical routines from SciPy (Jones, 2001) and NumPy (Oliphant, 2006), with 3D rending provided by Coin3D[2] and Pivy[3] in Python as well. Originally created by Yosef Meller at Tel Aviv University (TAU) in Israel[4], Tracer was further developed by the Solar Thermal Group (STG) at the Australian National University (ANU)[5] for modelling CSP systems. New additions include parallel processing capability for faster tracing, simulations of slope errors and sunshape distributions. A TowerScene module has been recently developed, which includes flexibility for different heliostat tracking mechanisms (azimuth-elevation, pitch-roll), different aiming strategies (single-point aiming or multiple fixed points aiming), heliostat slope error distributions (normal and pillbox) and field layout options etc. Preliminary verification of Tracer was presented by Wang et al. (2016) via reproducing the same test case published by Yellowhair et al. (2014). The result from Tracer agreed well with that from DELSOL, HELIOS, SolTrace and Tonatiuh. It has been used in the receiver design activities for the Big Dish system in the USASEC project at ANU (Asselineau et al., 2015). The highly efficient

2 Coin3D: https://bitbucket.org/Coin3D/coin/wiki/Home
3 Pivy: https://bitbucker.org/Coin3D/pivy
4 Tracer (Y. Meller): https://github.com/yosefm/tracer
5 Tracer (ANU): https://github.com/anustg/Tracer

receiver ‘SG4’ at ANU was evaluated using Tracer and was experimentally confirmed through on-sun tests (Pye et al., 2017).

2.1.3 Solstice

The SOLSTICE (SOLar Simulation Tool In ConcEntrating optics) is a free open source software released under the GPLv3+ license and jointly developed by PROMES-CNRS[6] and Méso-Star[7]. The integral formulation Monte Carlo (IFMC) algorithm (Delatorre and Baud, 2014) was implemented to solve the concentrated solar flux on a receiver, which was experimentally validated by Caliot et al. (2015). The convergence rate is faster than the collision-based algorithm (e.g. as used in SolTrace and Tonatiuh) as Solstice applies energy-partitioning method. It is also efficient in large field simulations due to its sampling rays of the first intersection on the primary reflector surface. It uses the Embree library from Intel® and is fully parallelisable on shared-memory architecture. The program is intended to be executed as a command-line tool enabling the user to couple the ray tracing simulation with other programs such as in optimisation loops and fluid mechanics or thermal software. Solstice considers input files containing the solar facility description, the geometry elements, the stereolithography (STL) files and spectral data (solar radiative intensity, refractive index, extinction coefficient and reflectivity), to compute the flux maps on receivers (with the associated statistical standard deviation) that can be visualised with the solar facility geometry using ParaView[8]. In addition, a map of local normal deviations could be attached to the reflector geometry to account for measured or simulated waviness of the reflectors due to the manufacturing process and the installation on the support structure.

2.1.4 Heliosim

Heliosim is an integrated central receiver CSP simulation and optimisation software developed by the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia. It is a closed commercial source package. The motivation for the initial development of the ray tracing model used by Heliosim in 2007 was for supporting experiments (e.g. receiver design and providing inputs for CSIRO’s heliostat control system software) performed using the two central receiver CSP facilities at the CSIRO Energy Centre in Newcastle, Australia (e.g. Kim et al., 2013). As part of the Australian Renewable Energy Agency (ARENA) sponsored “Optimisation of central receivers for advanced power cycles” project, receiver thermal modelling, pipe stress modelling, heliostat field optimisation, and a graphical user interface were added to the Heliosim software (Potter et al., 2017). The core physical modelling is implemented as a C++ library that is exposed as a plugin for Workspace (Watkins et al., 2017), a scientific workflow framework developed by CSIRO. The desired behaviour of the software is encapsulated as a Workspace workflow, which is compiled into a standalone application with a graphical user interface created using the Qt framework.

For simulating heliostat optics, Heliosim currently (version 5.4.0) implements a MCRT method where rays are cast from the primary reflector surfaces (i.e. the heliostat mirror facets). The incident direction, mirror intercept location and mirror surface normal direction for each ray are calculated via Monte Carlo sampling of cumulative distribution functions (CDF) using the function inversion method. As described in §4.6, Heliosim previously (version 4.0.3 and below) implemented a

6 Solstice PROMES: https://www.labex-solstice.fr/solstice-software/
7 Solstice meso-star: https://www.meso-star.com/projects/solstice/solstice.html
8 Paraview: https://www.paraview.org/

deterministic ray tracing model with an approximate treatment of off-axis incident rays, however errors identified as a result in the participation in this validation study motivated the implementation of the current MCRT model. The traversal of rays through the scene is handled by the GPU-accelerated NVIDIA® OptiX™ Ray Tracing Engine, where each object in the scene is represented by a surface mesh comprising of triangular facets, allowing complex receiver geometries and realistic shading scenes such as buildings and terrain to be considered. The ray tracing calculation can be run in parallel on distributed memory computer clusters, allowing in excess of 100 million rays to be traced each second. The computational efficiency of this ray tracing model has been used in objective functions for optimising heliostat field layout and receiver geometry (e.g. Potter et al., 2015).

2.1.5 SolTrace

SolTrace is generalised Monte Carlo ray tracing open source software that can model a wide variety of optical systems, surface geometries, and surface interactions by casting sun rays randomly over a region of interest, rather than by generating rays which originate from points on the heliostat surface. As such, it is not designed for efficient operation with power tower systems, but provides functionality that SolarPILOT (Wagner and Wendelin, 2018) utilises for that application, managing automated geometry definition, heliostat tracking, sun positions, and definition of error and sunshape distributions. The features are presented in detail by Wendelin and Dobos (2013). It was validated through comparison with measurements taken at the NREL High Flux Solar Furnace (Wendelin and Dobos, 2013). SolTrace can be used either as a module within via SolarPILOT or as stand-alone software that provides significantly greater flexibility and ray data analysis functionality.

2.1.6 SolarPILOT

SolarPILOT is an open source software that utilises analytical methods for optical modelling (Wagner and Wendelin, 2018). Instead of ray tracing, it uses the Hermite polynomial expansion technique to approximate reflected sun images as Gaussian distributions. It is a similar technique as applied in DELSOL that has been demonstrated by Walzel et al. (1977), Dellin (1979), and Kistler (1986). SolarPILOT implemented a number of improvements, including dynamic heliostat grouping, efficient annual performance prediction, and field layout generations that were reviewed by Wagner and Wendelin (2018). The analytical method is significantly more computationally efficient than ray tracing methods, but has limitations in precise modelling of all optical conditions, e.g. (1) non-Gaussian distributions, (2) multiple reflections in cavity-type receivers. It embeds the core tracing functions of SolTrace through an application programming interface (API) to assist these limitations.

2.1.7 Tools Overview

The simulation codes evaluated in this study can be regrouped in several categories. SolTrace and Tonatiuh both implement a purely stochastic ray tracing method and use the collision-based algorithm. Rays are associated with random variates in the range of 0 to 1. If the random variate is higher than the reflectivity of the intercepted surface, the ray is fully absorbed; otherwise, the ray is fully reflected. This method of handling events is also commonly labelled “Russian roulette”.

Tracer, Solstice and Heliosim apply the energy partitioning approach for ray tracing. In this approach, a certain amount of energy is carried by each ray and is reduced at each intersection point. Compared to the collision-based method, it requires recording the energy in each intersection, which implies relatively more computational effort per ray, but results in much faster convergence, which means significantly fewer rays for the whole simulation, especially in modelling of multiple reflection effects.

Rather than casting all sun rays from a plane located above the entire heliostat field, Solstice and Heliosim cast rays from the primary reflection surfaces (i.e. heliostat facets). It reduces the large wastage of rays hitting the ground, between the heliostats for example, and avoids the computationally expensive calculation of ray intercept locations on the heliostat mirror facets, but adds the need for a separate shading calculations.

The detailed information regarding version, link and contact authors of the tools that are verified in this study is listed in Table 1.

Table 1. Tools considered in this study

Name	Version	Open Source (URL)	Contact Author	Host Institution
Tonatiuh	2.2.3	http://iat-cener.github.io/tonatiuh/	Manuel Blanco	The Cyprus Institute
Tracer	1.0.0	https://github.com/anustg/Tracer	John Pye	ANU
Solstice	0.8.1	https://www.meso-star.com/projects/solstice/solstice.html	Cyril Caliot	PROMES-CNRS
Heliosim	5.4.0	Closed source commercial product (trial version available on request)	Daniel Potter	CSIRO
SolarPILOT	1.2.1	https://github.com/NREL/solarpilot	Michael Wagner	NREL
SolTrace	3.0.0	https://github.com/NREL/soltrace	Michael Wagner (Tim Wendelin - original)	NREL

2.2 Sunshape and Implementations in MCRT

The sun is not a point source but appears as a ‘disk’ when viewed from the earth. The sunshape describes the distribution of solar radiation in the solar disk, i.e. the normalised radiance profile ($\hat{L}(\theta)$) of the solar radiation, which is a distribution of the rate of energy per unit solid angle in a specified direction and per unit of projected surface area normal to the specified direction (Blanc et al., 2014).

The most realistic model of sunshape includes the limb-darkened solar disk with circumsolar radiation. The Buie sunshape is one such example, as shown in Eq. (1), where θ is the radial angular displacement, and each of κ and γ is a function of the circumsolar ratio (CSR or χ) (Buie et al., 2003). In optical simulations, θ_{disk} is usually taken as 4.65 mrad to represent the annual averaged angular width of the solar disk, and θ_{aureole} is 43.6 mrad to represent the angular extent of the aureole. It should be noted that the unit of θ is presented in milliradians by Buie et al. (2003). It is converted to radians here in Eq. (1) to comply with standard units.

$$\hat{L}_{\text{Buie}}(\theta)=\begin{cases}\dfrac{\cos(326\,\theta)}{\cos(308\,\theta)}, & 0\leqslant\theta\leqslant\theta_{disk}\\ e^{\kappa}(10^{3}\theta)^{\gamma}, & \theta_{disk}<\theta\leqslant\theta_{aureole}\end{cases} \tag{1}$$

The pillbox (Eq. (2)) and Gaussian (Eq. (3)) distributions are also widely applied models for simulating sunshapes in CSP due to their simplicity.

$$\hat{L}(\theta)_{Pillbox}=\begin{cases}L_{constant}, & 0\leqslant\theta\leqslant\theta_{disk}\\ 0, & \theta>\theta_{disk}\end{cases} \tag{2}$$

$$\hat{L}(\theta)_{Gaussian}=\frac{1}{\sqrt{2\pi}\,\sigma}e^{-\frac{\theta^{2}}{2\sigma^{2}}},\theta\geqslant 0 \tag{3}$$

In optical modelling, generally, a control surface normal to the main direction of the sun rays is considered to cast rays towards the solar concentrator. The normalised differential radiant power in a given direction towards the solar concentrator, that specified by (θ, φ), can be computed by the following expression:

$$d\,\hat{q}=\hat{L}(\theta)\sin\theta\,d\,\theta\,d\,\varphi\cos\theta\,dA\,. \tag{4}$$

The term of $\sin\theta d\theta d\varphi$ represents the solid angle subtended by the solar concentrator in dθ and dφ. The $\cos\theta\,dA$ is the term that converts the segment area to the one that is perpendicular to the direction of (θ, φ), also known as Lambert's Cosine Law. The total hemispherical radiant power from the infinitesimal area dA is obtained by integrating the previous expression for all possible values of θ and φ:

$$\hat{q}=\int_{0}^{2\pi}\int_{0}^{\pi/2}\hat{L}(\theta)\sin\theta\cos\theta\,d\,\theta\,d\,\varphi\,dA\,. \tag{5}$$

The ratio between the two previous expressions (4) and (5) is the probability of a ray leaving the surface in the direction specified by (θ, φ):

$$P(\theta,\varphi)_{\text{sunshape}}=\frac{\hat{L}(\theta)\sin\theta\cos\theta\,d\,\theta\,d\,\varphi}{\int_{0}^{2\pi}\int_{0}^{\pi/2}\hat{L}(\theta)\sin\theta\cos\theta\,d\,\theta\,d\,\varphi}\,. \tag{6}$$

The cumulative distribution function (CDF) is:

$$F(\theta,\varphi)_{\text{sunshape}}=\frac{\int_{0}^{\varphi}\int_{0}^{\theta}\hat{L}(t)\sin(t)\cos(t)\,dtdu}{\int_{0}^{2\pi}\int_{0}^{\pi/2}\hat{L}(\theta)\sin\theta\cos\theta\,d\,\theta\,d\,\varphi}\,. \tag{7}$$

The azimuthal CDF is:

$$F(\varphi)_{\text{sunshape}}=\frac{\varphi}{2\pi}\,. \tag{8}$$

The CDF of the radial angular displacement (θ) is:

$$F(\theta)_{\text{sunshape}} = \frac{\int_0^{\theta} \hat{L}(t)\sin(t)\cos(t)\,dt}{\int_0^{\pi/2} \hat{L}(\theta)\sin\theta\cos\theta\,d\theta}. \quad (9)$$

In MCRT, ray-sampling expressions are obtained by inversion of the CDF expressions (Arvo et al., 2003). The azimuthal angle sampling expression is relatively trivial as the ray directions are assumed azimuthally symmetric:

$$\varphi = 2\pi \cdot R_{\varphi},\ R_{\varphi} \in (0,1), \quad (10)$$

where R_{φ} is a set of uniform random variates. The zenithal angle sampling expression is more complex and depends on the sunshape expression considered, as presented in the following sections.

2.2.1 Sampling θ for the Pillbox Sunshape

The angular displacement θ in the pillbox sunshape can be sampled directly since (9) can be performed analytically:

$$\theta = \sin^{-1}(\sin\theta_{disk} \cdot \sqrt{R_{\theta}}),\ R_{\theta} \in (0,1), \quad (11)$$

where R_{θ} is a set of uniform random variates. This approach is used in Tonatiuh and Solstice.

It should be noted that θ is typically very small (e.g. θ <100 mrad), such that $\cos\theta \approx 1$ and $\sin\theta \approx \theta$. It is reasonable to simplify (9) to (12).

$$F(\theta) = \frac{\int_0^{\theta} \hat{L}(t)\,t\,dt}{\int_0^{\pi/2} \hat{L}(\theta)\theta\,d\theta} \quad (12)$$

The simplified sampling expression for the angular displacement in the pillbox distribution is :

$$\theta = \theta_{disk} \cdot \sqrt{R_{\theta}}. \quad (13)$$

This approach is tested in Tracer, and the results are verified against other methods. Details are shown in the next section.

2.2.2 Sampling θ for the Gaussian Sunshape

The angular displacement θ in Gaussian sunshape cannot be analytically sampled by (9), but is applicable by its simplification (12). Taking (3) into (12), it is found that

$$\theta = \sigma\sqrt{-2\ln[1 - F(\theta)(1 - e^{\frac{-\pi^2}{8\sigma^2}})]}. \quad (14)$$

The term $e^{\frac{-\pi^2}{8\sigma^2}}$ is less than 10^{-11} even the standard deviation σ reaches 50 mrad, which can be approximately treated as 0. Thus the sampling expression of θ can be simplified as:

$$\theta = \sigma\sqrt{-2\ln(R_{\theta})}. \quad (15)$$

This is implemented and tested in Tracer, and the results are identical with other codes.

Alternatively, the projected length of the ray direction vector in the xz and yz plane (*r, s*) can be sampled assuming that they both follow Gaussian distributions with a standard deviation σ. This

simplification, introduced by Biggs and Vittitoe (1979), relies on the small angular deviation assumption. The directional vectors can be sampled directly as (ε_x, ε_y, 1), where ε_x and ε_y are random variates in following a normal distribution with a standard deviation of σ. This approach is also verified via Tracer.

2.2.3 Sampling θ for the Buie Sunshape

For the Buie sunshape, neither the cumulative distribution function (9) nor its simplified form (12) can be fully integrated analytically. The solar disk region of the sunshape requires numerical treatment. Two main methods are employed for the numerical integration: (1) approximating the sunshape as a piecewise linear function that can be integrated (as applied in Tracer and Heliosim) or (2) using a 'Rejection' sampling method (Arvo et al., 2003) as implemented in Tonatiuh and Solstice.

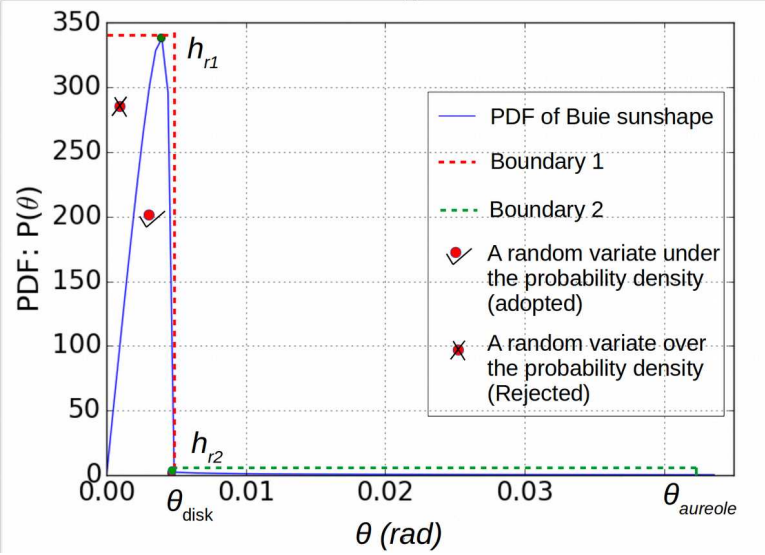


Fig. 1. Illustration of a two-region rejection sampling of the angle θ for the Buie sunshape. h_{r1} is the height of the upper boundary of the probability density function in the solar disk region; h_{r2} is that in the circumsolar region.

The rejection sampling method is a stochastic trial-and-error method that works by declaring uniformly random θ angles and associate them with a uniform random variate weight. This weight is compared with the probability density function (PDF) of the sunshape. Angles are accepted as valid if their weights are lower than the corresponding probability; otherwise they are discarded. It is possible to introduce sampling regions to improve the performance of this method, as illustrated in Fig. 1 with two regions.

2.3 Surface Slope Error and Implementations in MCRT

Macroscopic deformations and microscopic roughness alter the optical behaviour of surfaces. Within the geometrical optics limits, these irregularities cause a local deviation of the surface slope, which in turn influences the direction of the reflection. The exact geometry of real surfaces is complex. Modelling these irregularities is done by artificially changing the local surface normal directions by using statistical distributions. The degree of the irregularities is often called surface slope error (or simply slope error) and is quantified by the one-sided deviation from the ideal direction of the normal vector using root-mean-square (RMS) angular width. In addition to surface slope errors, the optical error of a concentrator system also includes contributions from specular errors (e.g. soiling scattering of radiation), tracking errors and the displacement of positions (Rabl,

1985). Each error type can be specified by its RMS angular width. Whilst in this paper only slope error models are verified, the verification also applies to other error mechanisms that use similar statistical distributions.

Statistically, the surface slope error distribution $\hat{P}(\theta)$ can be approximated as a pillbox or Gaussian distribution (Rabl, 1985). This distribution gives deviation of surface normals in any specified solid angle increment (Biggs and Vittitoe, 1979). It should be emphasised that the surface slope error distribution is not describing the displacement of normal vectors in any range but describing the displacement of normal vectors per solid angle increment. Therefore, the probability of an actual normal vector at a point on the concentrator surface points to a given direction that is specified by (θ, φ) is:

$$P(\theta,\varphi)_{\text{slope}} = \frac{\hat{p}(\theta)\sin\theta\, d\theta\, d\varphi}{\int_0^{2\pi}\int_0^{\pi/2} \hat{p}(\theta)\sin\theta\, d\theta\, d\varphi}\,. \tag{16}$$

Similarly to the sunshape, the cumulative distribution function (CDF) is:

$$F(\theta,\varphi)_{\text{slope}} = \frac{\int_0^{\varphi}\int_0^{\theta} \hat{p}(t)\sin(t)\, dt du}{\int_0^{2\pi}\int_0^{\pi/2} \hat{p}(\theta)\sin\theta\, d\theta\, d\varphi}\,. \tag{17}$$

The azimuthal CDF is:

$$F(\varphi)_{\text{slope}} = \frac{\varphi}{2\pi}\,. \tag{18}$$

The CDF of the angular width (θ) is:

$$F(\theta)_{\text{slope}} = \frac{\int_0^{\theta} \hat{p}(t)\sin(t)\, dt}{\int_0^{\pi/2} \hat{p}(\theta)\sin\theta\, d\theta}\,. \tag{19}$$

The expressions for slope error modelling are identical to the sunshape modelling expressions and the same approximations (small angles and direct planar projection) are also therefore available. In the pillbox slope error, θ can be sampled as Eq. (13). In the normal slope error, θ can be sampled as Eq. (14). The alternative method (direct planar projection) that was introduced in section 2.2.2 for Gaussian sunshape is also applicable to the normal slope error. In Tonatiuh, Gaussian sampling is implemented using the Box-Muller transformation (Box and Muller, 1958).

2.4 Convolution Method in Brief

An alternative method to MCRT uses the convolution of analytical distributions representing the mirror shape, optical errors and sunshape, for faster optical modelling. In general, the accuracy of convolution methods is lower than MCRT, however the increased computational performance allows the calculation of annual performance and optimisation of heliostat field layouts for large-scale central tower systems to be performed using standard desktop computers. The convolution method is also commonly referred to as the ‘cone optics’ method.

In general, three steps are involved in the analytical approach to determine the flux distribution reflected by each heliostat: (1) obtaining the principal image of a heliostat (*M)*, (2) convolving the principal image with the distributions of the sunshape (*S)* and the optical errors (*G*) to obtain the reflected image and (3) mapping the reflected image onto a receiver (Walzel et al., 1977; Grigoriev and Corsi, 2017). The principal image of a heliostat is a virtual image formed by projection of the effective area of the heliostat facet (i.e. after shaded and blocked regions are removed) onto the image plane. The image plane is located at the centre of the target and normal to the line between the centre of the target and the centre of the heliostat. At all locations on the image plane, the principal image can be 'blurred' to account for the sunshape and optical error distributions, to give the resulting flux distribution on the image plane *F* (Lipps, 1976). Mathematically, *F* is calculated as the convolution integral combining *M*, *S* and *G*:

$$F = M * S * G \, . \tag{1}$$

Solving the convolution integral can be non-trivial due to arbitrary distributions of *M*, *S* and *G*. Collado et al. (1986) reviewed the numerical treatments of *M*, *S* and *G* in different codes developed in 1980s. One widely used numerical method is expanding the components of *F* into sums of orthogonal polynomials. The orthogonal polynomials with respect to the Gaussian distribution, i.e. the Hermite polynomials, are generally adopted since they provide good representation of typical flux patterns using a small number of terms, and provide uniform convergence on the entire image plane (Walzel et al., 1977). The Hermite polynomial approach was reviewed by Wagner and Wendelin (2018), and applied in SolarPILOT, the details of which have been introduced in §2.1.6. The performance of SolarPILOT is compared with MCRT methods in this study and discussed in the following sections. Other tools implementing the convolution method via Hermite polynomials include DELSOL (Dellin and Fish, 1979) and UHC/RCELL (Lipps and Vant-Hull, 1978).

There are further simplified treatments of the convolution integral in previous work (Garcia et al., 2008; Collado et al., 1986). Three examples are UNIZAR (Collado et al., 1986; Collado, 2010), HFLCAL (Schwarzbözl et al., 2009), and a unified algorithm developed by Grigoriev and Corsi (2017). Both the UNIZAR model and the HFLCAL model approximate *S* and *G* as a radially symmetric Gaussian. The principal image *M* is assumed to be rectangular in the UNIZAR model, whereas it is assumed to be a point in the HFLCAL model. Off-axis aberrations (i.e. astigmatism errors) are considered in the HFLCAL model by enlarging the standard deviation of the optical errors, but are excluded in the UNIZAR model. Both the UNIZAR and HFLCAL models showed acceptable accuracy of less than ±9.3% absolute difference in intercept fraction for all individual heliostats, when compared to experimental measurements (Collado, 2010). Meanwhile, the algorithm of Grigoriev and Corsi (2017) improves the representation of *M* using a decomposition of the shape into a set of right triangles, allowing arbitrary heliostat geometries to be considered, and is furthermore valid for any arbitrary sunshape having radial symmetry. The convolution of the sunshape and optical error distributions are also pre-computed, and implemented using high-speed graphics processing code. These codes were excluded from the present study.

3. Models and Results

The verification is done thoroughly by three rounds of tests, with gradually increasing complexity, from single heliostat to full field simulations. Descriptions of each case and selective results are

presented and discussed in the following sections. Details are available in the supplementary material for readers who are interested in using these tests to verify their code. Data files can be accessed via the ANU STG Github repository[1], include case descriptions, parameter list of each case, the heliostat field coordinates, result data files, and Python scripts of applying Tracer for running each of the test and data post processing.

3.1 Round A

3.1.1 Model

A solar source, one paraboloid mirror and a flat target in an axially aligned configuration (as shown in Figure 2) are used for the first-round test in order to individually allow sunshape and slope error to play leading roles in the results. Collimated rays are simulated in the cases where surface slope errors are examined; while zero slope error is applied when sunshape is being checked. The ray source is arranged under the target to avoid shading effects. Under such arrangements, a theoretical flux distribution on the receiver target for each case is readily obtained correspondingly (see §3.1.2).

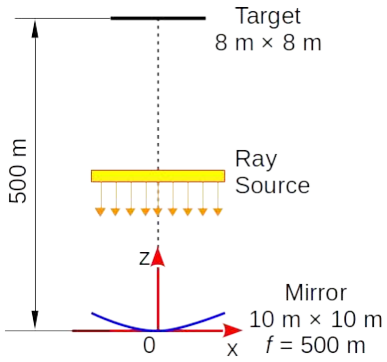


Fig. 2. Test model of Round A

Preliminary results of this test round were presented at the SolarPACES conference in 2017 (Wang et al., 2017). The sizes of the mirror and the target from that study have been enlarged in the present work to be matched to the case of a large scale heliostat field. In addition, two combination cases of sunshape and slope error are simulated. Those are pillbox sunshape with normal slope error (A3.1) and Buie sunshape with normal slope error (A3.2). A summary of the test cases is listed in Table 2.

A rectangular 100 × 100 mesh is overlaid on the target for binning of the output flux map for each case. Each term of energy is analysed according to the balance:

$$\dot{Q}_{\mathrm{irr}} = \dot{Q}_{\mathrm{abs}} + \dot{Q}_{\mathrm{refl}} + \dot{Q}_{\mathrm{spil}} \,, \tag{20}$$

where $\dot{Q}_{\mathrm{irr}}$ is the total energy reflected by the mirror, $\dot{Q}_{\mathrm{abs}}$ is the energy absorbed at the target, $\dot{Q}_{\mathrm{refl}}$ is the energy reflected by the target, and $\dot{Q}_{\mathrm{spil}}$ is the energy spillage from the target. This energy balance is also valid for Round B.

Table 2. Test cases of Round A

		Slope error distributions of the mirror surface		
		No error	**Pillbox**	**Normal**
Sunshape distributions	**Collimated rays**	-	(A 1.1) σ_{slope} =1, 2, 3 mrad	(A 1.2) σ_{slope} =1, 2, 3 mrad
	Pillbox	(A 2.1) θ_{sun} = 4 mrad	-	(A 3.1) θ_{sun} =4.65 mrad, σ_{slope} =2 mrad
	Gaussian	(A 2.2) θ_{sun} = 4 mrad	-	-
	Buie	(A 2.3) CSR = 0.01, 0.02, 0.03	-	(A 3.2) CSR = 0.02, σ_{slope} =2 mrad

3.1.2 Theoretical Radiance Distribution on the Target

375 The axially aligned arrangement makes it possible to calculate the theoretical radiance distributions on the target for cases A1 and A2, in which sunshape and slope error are examined separately. As the distance between the mirror and the target is far (50 times the side-length of the mirror), the radiance distribution will be very close to the corresponding statistical distribution of the slope error or the sunshape, as explained below.

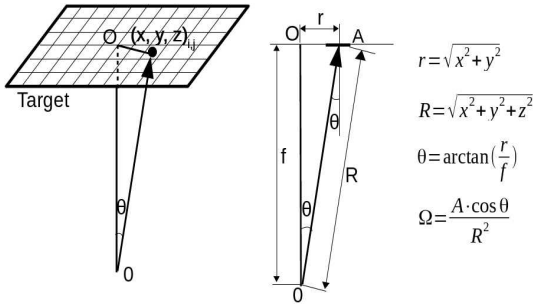


Fig. 3. The angular displacement θ and the solid angle Ω that subtended by the area of the mesh element i,j at point 0.

380 The absorbed flux at each mesh element on the target ($q_{i,j}$) is obtained in each simulation. The radiance $L_{i,j}$ can be calculated by (21), where Ω is the solid angle that subtended by the area of the element (i,j) at point 0, as shown in Figure 3.

$$L_{i,j}=\frac{q_{i,j}}{\Omega} \tag{21}$$

By separating the angular displacement θ into small segments in a radial direction and binning the radiance $L_{i,j}$, the function $L(\theta)$ can be established.

385 In order to compare the results with theoretical statistical distributions, the radiance distribution needs to be normalised:

$$\hat{L}_s(\theta)=\frac{L(\theta)}{\int_0^{\theta_{rim}} L(\theta)\,d\theta}\cdot\int_0^{\theta_{rim}} D(\theta)\,d\theta\,, \tag{22}$$

where $\hat{L}_s(\theta)$ is the normalised radiance from simulations, and $D(\theta)$ is the statistical distribution of $\hat{L}(\theta)$ or $P(\theta)$ presented in §2. It should be noted that for slope errors, the deviation of the

statistical distribution (e.g. the sigma in a normal distribution) should be doubled due to Snell's law of reflection.

3.1.3 Selected Results and Discussion

3.1.3.1 Slope Error

For pillbox distributions, the results from all tools except SolarPILOT and Tonatiuh agree well (see Figure 4 and 5 (a)). SolarPILOT has no option of pillbox slope error for surface optics and could not be included in this test. A discrepancy from Tonatiuh can be seen clearly in Figure 4.

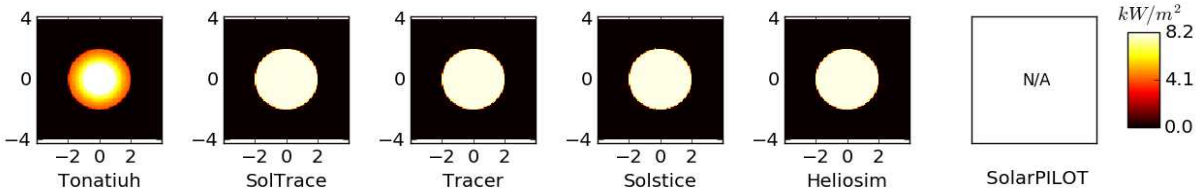
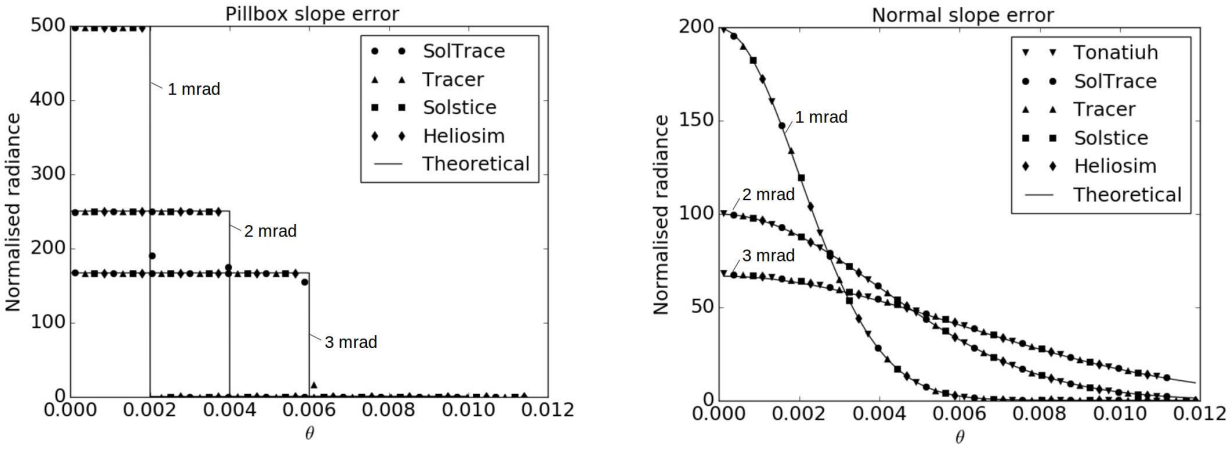


Fig. 4. Flux map of the 2 mrad pillbox slope error (Case A 1.1.2), the discrepancy of Tonatiuh shows the improper implementation of pillbox slope error. Flux map axis labels are in metres.

The sampling of the angular displacement θ is uniformly distributed in 0 and θ_s in the 2.2.3 version of Tonatiuh that participated in this study, rather than the one that described in Section §2.3. This implementation can be checked in the Tonatiuh package: "MaterialStandardSpecular" class, "OutputRay" method. This error equates to ignoring the non-linear increase of the solid angle as the θ increases. This oversight is not found in the pillbox sunshape which was implemented properly. It is expected that this error will be corrected in a future release.



(a) (b)

Fig. 5. The normalised radiance distributions from each simulation compare to the theoretical statistical distributions for (a) the pillbox slope error test case and (b) the normal slope error test case.

It should be noted that the pillbox distribution of slope error is only interesting from a validation point of view. In applications, normal distributions are more common in modelling optical errors for a large array of collectors. Besides, based on the central limit theorem of statistics, when many statistically independent distributions are integrated (e.g. from multiple heliostats/facets), the result soon approaches a Gaussian distribution.

For the results of normal distribution slope errors (case A1.2), previously SolTrace and SolarPILOT showed discrepancies relative to other results. The reasons were found along with the progress of this study and are described in more detail in §4.2 and §4.3. The discrepancies were allowed to be

amended. Good agreement is now shown among all the six examined tools, as can be seen in Figure 6 and 7.

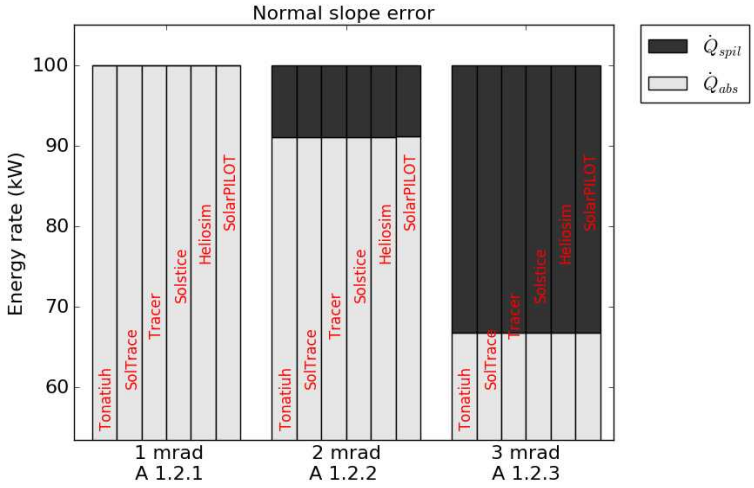


Fig. 6. Energy bar chart of the normal slope error case (Case A 1.2)

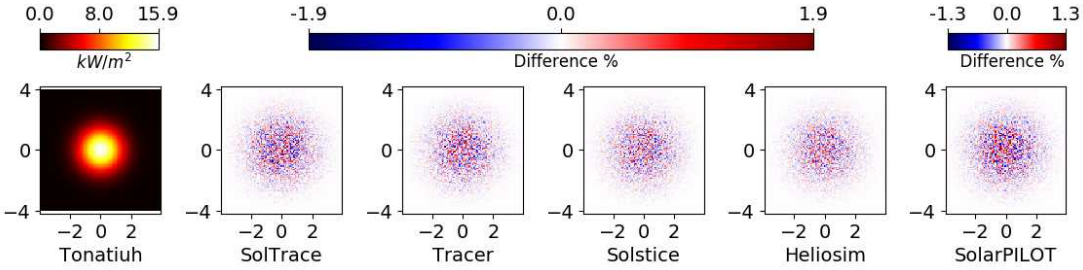


Fig. 7. Flux map of the 1 mrad normal slope error (Case A 1.2.1) from Tonatiuh and the differences of flux map from other tools compared to that from Tonatiuh. The difference in percentage is defined as the difference of flux value over the maximum flux value of the reference case. Flux map axis labels are in metres.

3.1.3.2 Sunshape

Good agreements are observed in the modelling of the pillbox and Gaussian sunshape distributions among tools except with SolarPILOT in the pillbox case.

SolarPILOT performs well in the Gaussian distribution cases, but is limited in representations of other distributions, e.g. Buie sunshape or pillbox distributions. SolarPILOT approximates the statistical distributions as polynomials by Hermite series expansion, and convolves them together for predicting the reflected image. As a simplification, the first seven terms in the polynomial expansion were applied to perform the convolution. This works well in predicting absorbed energy and local flux density distributions on the target in most realistic situations where error sources are compounded on each other and inclining to a Gaussian distribution, but in the specific case in this test round where a non-Gaussian sunshape distribution coupled with an optically perfect reflector, the polynomial expansion does not represent the specified sun shape. SolarPILOT is powerful in quickly estimating the optical performance for a large heliostat field. It just takes seconds or fractions of a second for designing a commercial scale heliostat field layout or suggesting aiming strategies, but has limitations in accuracy in certain circumstances.

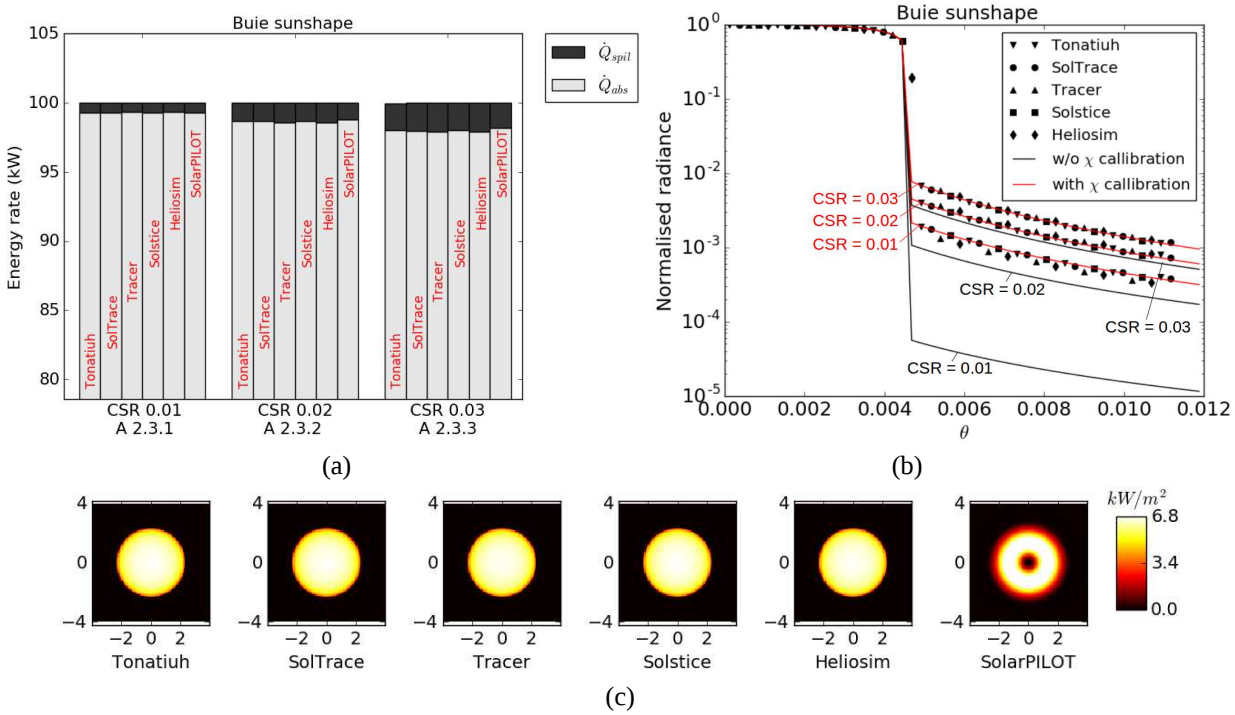


Fig. 8. Results of Buie sunshape test with (a) the total absorbed energy, (b) the normalised radiance distribution on the target and (c) the flux maps for a CSR of 0.02. Flux map axis labels are in metres.

Apart from SolarPILOT, some small differences among other tools appear in the simulations of Buie sunshape, as shown in Figure 8. These discrepancies are caused by an issue in Buie's original correlation (Buie et al., 2003) that was identified independently by the researchers working on the development of Tonatiuh and Tracer. The parameter χ is the circumsolar ratio (CSR) that defines the Buie sunshape profile. The issue is that the circumsolar ratio that is then calculated from the defined Buie profile is not equal to the assigned χ value. To address this issue, polynomial calibration correlations are applied in both codes respectively to make CSR = χ. The source code can be referred to as the demonstration in the Tracer repository[9]. The polynomial calibration equation from Tonatiuh was implemented in Solstice, and recently added in SolTrace and SolarPILOT, and that from Tracer has since been implemented in Heliosim. The slight differences shown in the energy bar chart (Figure 8(a)) are caused by this issue. The developers are collaborating and preparing another thorough study to reveal this issue.

3.1.3.3 Combination of Sunshape and Slope Error

When the sunshape distribution is simulated together with a slope error, the individual impact of each distribution becomes less significant and is inclined to a Gaussian distribution. Figure 9 shows the flux map of Buie sunshape (CSR 0.02) with normal slope error from Tonatiuh, and the differences compared to other optical modelling tools. Even though the flux distribution varies significantly when modelling a Buie sunshape with zero slope error in SolarPILOT, it is not significant when combined with a physically realistic slope error. However, further investigations on the energy balance of SolarPILOT are still required, as discrepancies can be observed in case A 3.2 in Fig. 10.

9 Calibration of CSR in Buie sunshape (source code): https://github.com/anustg/Tracer/blob/master/tracer/sources.py

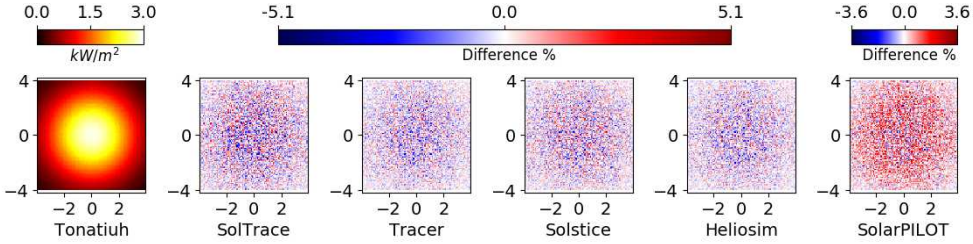


Fig. 9. Flux map of case A 3.2 Buie sunshape (CSR 0.02) with 2 mrad normal slope error from Tonatiuh, and the differences of flux map from other tools compared to that from Tonatiuh. The difference in percentage is defined as the difference of flux value over the maximum flux value of the reference case. Flux map axis labels are in metres.

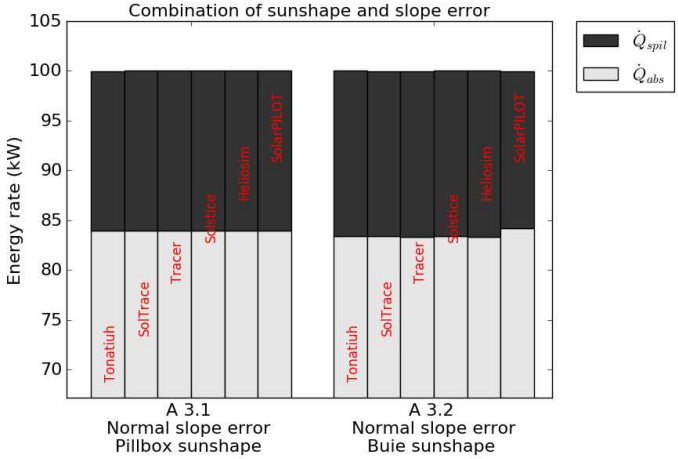


Fig. 10. Energy bar chart of Case A 3

3.2 Round B

A PS10-like (Osuna et al., 2006) radially-staggered heliostat field (Figure 11) was generated by SolarPILOT to verify the optical modelling in a more realistic scenario for the Round B and Round C tests. The total designed power is 30 MW_{th} and located in Barstow, California. The simulated heliostat field is constituted by 522 heliostats, each of them a 10 m by 10 m single-facet mirror, ideally focused without canting. The tower height is 62 m. The receiver is a 6 m height (in the vertical direction) and 8 m width billboard, with the centre located at (0, 0, 62 m). The coordinate system follows the right-hand rule and the positive y points to the North direction. The coordinates of the heliostat field can be found in the website repository[1].

Four representative positions are selected for individual tests in Round B and the full field simulations are performed in Round C. Sunshape and slope errors effects are now considered simultaneously. Different sun positions (morning and noon) are simulated. The tracking mechanism is azimuth-elevation with pivoting axes centred at the middle of each heliostat. Whilst shading and blocking are considered for the full field simulations in Round C, they are not considered in Round B (i.e. the individual heliostats are considered in isolation from the effects of other neighbouring heliostats).

3.2.1 Model

The four representative heliostats in Round B tests are shown in Fig. 11.

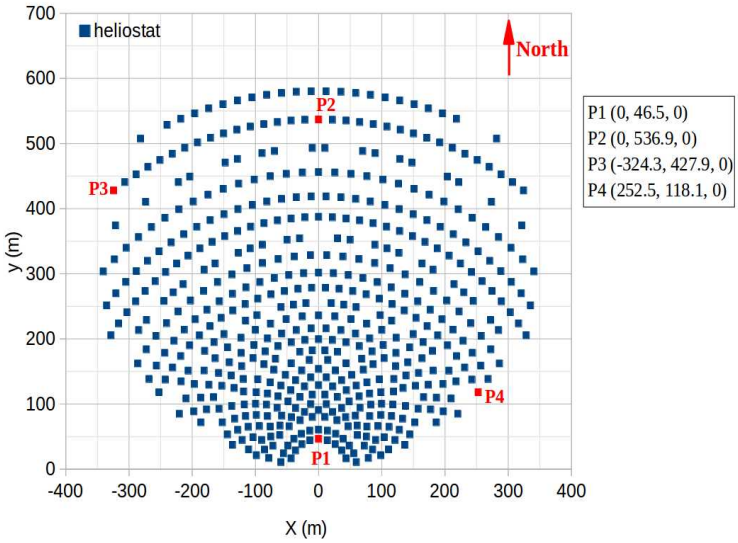


Fig. 11. A PS10-like radially-staggered heliostat field used in round B and C, as created by SolarPILOT. The layout indicates 522 heliostats, a 10 × 10m single facet mirror, ideally focused with no canting, 62 m tower height, 30 MW_{th} with 6 m height and 8 m width billboard receiver. The site location is Barstow, California USA. Heliostats labelled P1 to P4 are the four points that selected for tests in Round B. The coordinates can be found in the website repository[1].

Two sun positions on the summer solstice (20th June) at Barstow, California US (W116º56', N34º53') are simulated: (1) solar noon: azimuth 180º and zenith 12º; (2) morning (two hours after sunrise): azimuth 76º and zenith 68º. The azimuth is the angle from North increasing towards to East (E of N) and zenith is the angle between the solar vector and the vertical axis.

Two combinations of sunshape and slope error are simulated: case B1 is the pillbox sunshape (4.65 mrad) with normal slope error (2 mrad); case B2 is the Buie sunshape (CSR 0.02) with normal slope error (2 mrad).

3.2.2 Selected Results and Discussion

Discrepancies were identified through the exercises and were revised to improve the quality of the tools. The improvements are summarised in §4. In most of the cases, now very small deviations can be observed among Tonatiuh, SolTrace, Tracer, Solstice and Heliosim, and larger deviations are seen in most cases for SolarPILOT. Figure 12 shows an example of the results. The maximum local flux differences are within 2.4 % from SolTrace, Tracer, Solstice and Heliosim compared to Tonatiuh. The reason for discrepancies for SolarPILOT were explained in §3.1.3.2.

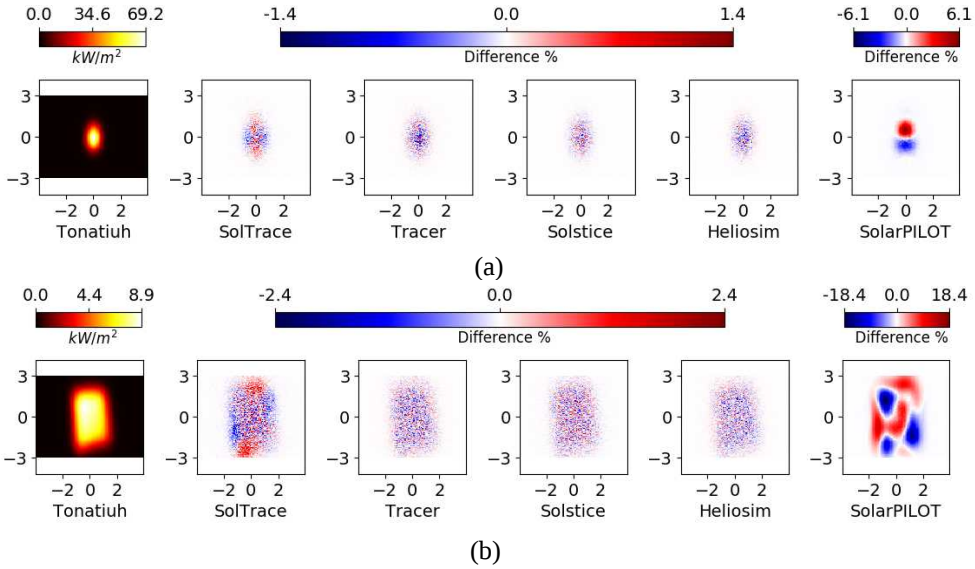


Fig. 12. P1 position results using the Buie Sunshape and Gaussian slope error at (a) Solar Noon and (b) morning sun postions. Flux map axis labels are in metres.

3.3 Round C

3.3.1 Model

The full field, presented in Figure 11, is simulated in this test round. The results are compared for two sun positions, and two types of combination of sunshape and slope error, which are the same as in Round B. Figure 13 shows an example of the 3D visualisation of the morning case (from Tracer).

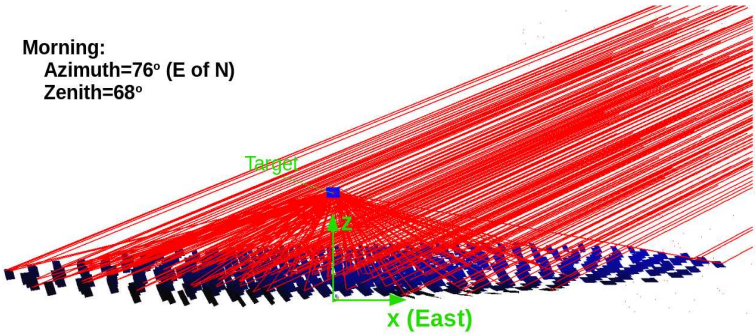


Fig. 13. Sun position in the morning (2 h after sunrise) (from Tracer)

In addition, the reflectivity of heliostats 0.95 and absorptivity of the receiver 0.9 are used in this test round, whereas they were set to unity in the previous rounds. The atmospheric attenuation is not considered in this study.

The breakdown of energy is recorded for each simulation, namely:

- $\dot{Q}_{\text{all}}$, the rate of the maximum radiative energy on the heliostats from the sun, which is equal to the total aperture area of heliostats multiplied by the direct normal irradiation (DNI);

$$\dot{Q}_{\text{all}} = A_{\text{heliostats}} \cdot \text{DNI} \tag{26}$$

- $\dot{Q}_{\cos}$, the rate of energy losses due to the cosine effect;
- $\dot{Q}_{\text{shad}}$, the rate of energy losses due to shading;
- $\dot{Q}_{\text{hstat,abs}}$, the rate of energy losses due to heliostat absorption;
- $\dot{Q}_{\text{block}}$, the rate of energy losses due to blocking;
- $\dot{Q}_{\text{spil}}$, the rate of energy reflected from the heliostats but misses the target, i.e. spilled;

- $\dot{Q}_{\text{refl}}$, the rate of energy reflected by the target;
- $\dot{Q}_{\text{abs}}$, the rate of energy absorbed by the target.

The energy balance is:

$$\dot{Q}_{\text{all}} = \dot{Q}_{\text{cos}} + \dot{Q}_{\text{shad}} + \dot{Q}_{\text{hstat,abs}} + \dot{Q}_{\text{block}} + \dot{Q}_{\text{spil}} + \dot{Q}_{\text{refl}} + \dot{Q}_{\text{abs}} \ . \tag{27}$$

3.3.2 Selected Results and Discussion

Figure 14 shows the results of (a) the breakdown of energy distribution of the full field case obtained by each optical modelling tool and (b) the difference in each energy term compared to that of Tonatiuh. The difference in percentage is defined as the difference value over the corresponding energy term of Tonatiuh. In general, the solar noon cases show less difference than the morning cases.

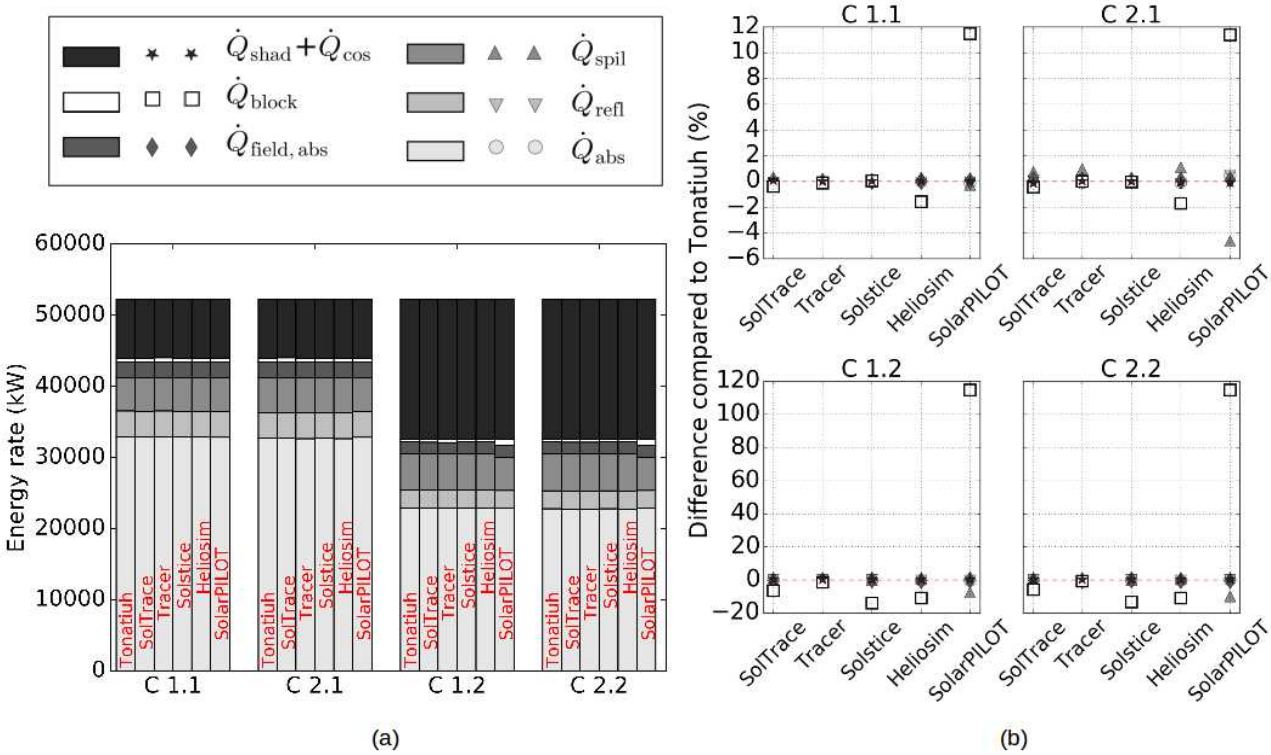


Fig. 14. Results of Round C: (a) Energy bar chart and (b) relative difference compared to Tonatiuh. C1.1: solar noon, pillbox sunshape; C2.1 Solar noon, Buie sunshape; C1.2: morning, pillbox sunshape; C2.2: morning, Buie sunshape

Tracer presents good agreements with Tonatiuh in each energy term in all the cases. The difference in the spillage shown in Fig. 14(b), case C2.1 is due to different CSR calibrations as discussed in §3.1.3.2.

SolTrace initially underestimated the spillage by 3–4% and the blockage by 6–9% in all the cases, but after implementing the corrections described in §4.2, the results agree.

Solstice now underestimates blocking by about 13% for morning cases (C1.2 and C2.2) whereas no significant discrepancy was found for noon cases. Heliosim performs well in the noon cases as all the differences are less than 2%, but it does not perform well on blockage in the morning cases as the differences reach ~10%. In these two tools that both have a discrepancy in the calculation of blockage, rays are sampled directly from the primary reflector surface, instead of that from a sky that covers the whole field. As reviewed in §2.1.7, this method reduces the large wastage of rays

hitting the ground, but requires a separates shading calculations. In the test results, the quantities of the shading losses from these two tools agreed well with the results from the others. The locations and directions of the sampled rays in cases that shading effects are involved require further investigation.

The blockage in the morning case from SolarPILOT is over twice that of Tonatiuh. The discrepancy is likely due to the geometric calculations used in SolarPILOT that are designed to be conservative in estimating the losses.

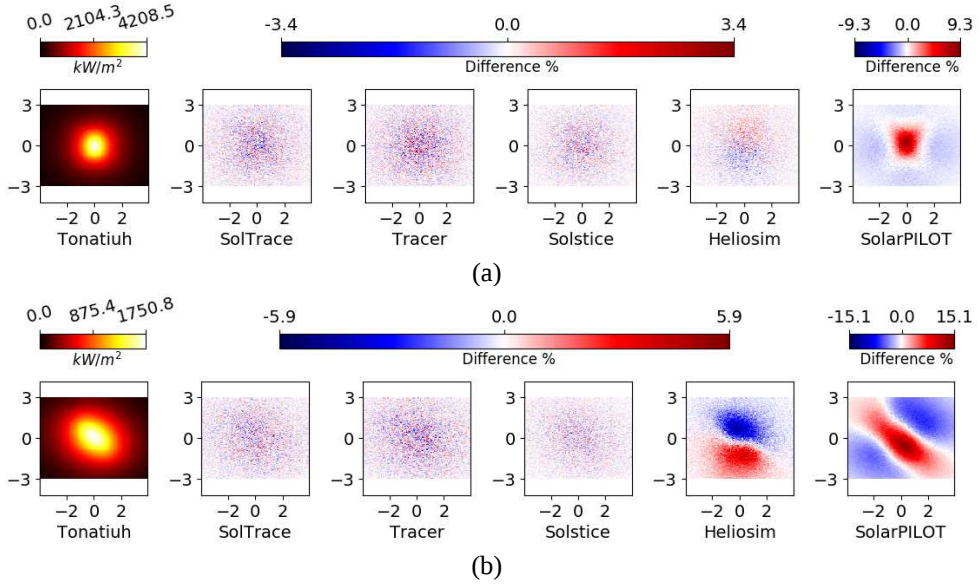


Fig. 15. Pillbox sunshape in Round C at (a) solar noon and (b) morning sun positions. Flux map axis labels are in metres.

Figure 15 shows the flux distributions from each tool compared to Tonatiuh in the cases of the pillbox sunshape at both the solar noon and morning sun positions[10]. The flux distributions from SolTrace, Tracer and Solstice agree well with Tonatiuh. Heliosim performs well in the solar noon position with the maximum local flux difference just 1.8%, but higher differences (5.9%) in the morning case. However, in the individual heliostat tests in Round B, the pattern of flux distribution from Heliosim is matching well with Tonatiuh. It is suspected that the issue is coming from shading or blockage which were not present in Round B. The flux distribution from SolarPILOT shows the most significant differences. The maximum local flux difference is around 9.3% in the noon case, and around 15.1% in the morning case. Such differences can be acceptable in some situations considering the fast computational speed that it provides. Further research is required to reveal the reason for these differences. Tracking, which is also an important factor that influences the optical simulation results, has not been compared in this study. Verifications on optical modelling of tracking mechanisms and tracking errors will be studied in future work.

In terms of the overall efficiency, defined as $\dot{Q}_{\text{abs}}$ over $\dot{Q}_{\text{all}}$, the maximum difference compared to Tonatiuh is 0.47%, which comes from SolarPILOT in case C2.1. The differences from the rest of the tools are all within 0.2%.

4. Summary: Value of This Study to the Six Tools Evaluated

10 The original flux maps and results obtained by all the tools are available in the supplementary material.

In the three rounds of tests, six optical modelling tools are reviewed. Through these exercises, discrepancies were identified and allowed to be revised to improve the quality of the tools.

4.1 Tonatiuh

In Tonatiuh (version 2.2.3), the incorrect implementation of pillbox slope error was identified and discussed in §1.3.3.1. This issue will be corrected in the next released version. Tonatiuh is a widely used optical modelling tool for CSP that has been validated through experimental tests, and it was used as the benchmark in this study to compare the other tools. Even though discrepancies were observed, the results of Tonatiuh agreed well with the majority of the tools in each case, indicating its quality of optical modelling.

4.2 SolTrace

SolTrace predates Tonatiuh and is widely used for CSP simulations by researchers. Several changes and bug fixes were identified and implemented as a result of this study. Firstly, it was determined that the Buie sunshape equation was not accurately predicting power in the circumsolar region in most cases. The Buie sunshape correction equation that is used in Tonatiuh was adopted in generating the results for this paper. Secondly, it was observed that SolTrace consistently slightly underestimated the variance in normal distribution population samples. In effect, this led to increased peak flux density near the centre of the image and decrease predicted spillage losses near the periphery of the image on the order of 1–2%. The cause of this error was found to be an approximation in the original Gaussian distribution model, which is shown in the pseudocode in Fig.16, where ‘sigma’ is the standard deviation of the normal distribution, ‘random()’ is a function used to generate a uniform random number between zero and one (inclusive), and the angle of displacement from the mean of the distribution is calculated as ‘theta’.

```
delta = 3*sigma;
do
{
	do
	{
		theta_x = delta*(2-random());
		theta_t = exp(theta_x^2 / (2*delta^2))^-1;
	} while (random() > theta_t);
	do
	{
		theta_y = delta*(2-random());
		theta_t = exp(theta_x^2 / (2*delta^2))^-1;
	} while(random() > theta_t);
	theta = theta_x^2 + theta_y^2;
} while(theta > delta^2);
theta = sqrt(theta_x^2 + theta_y^2);
```

Fig. 16. The original implementation of Gaussian distribution in SolTrace that underestimated the variance

Ultimately, the distribution was replaced with the built-in normal distribution generator in the C++ standard library, and this resolved the discrepancies with other models. While a specific problem with this algorithm was not resolved, it was observed that the scaling of sigma with respect to delta affected the distribution in much the same way as using the built-in generator.

4.3 SolarPILOT

The performance predictions using SolarPILOT's analytical model were largely consistent with the ray tracing results, although some differences were observed. The exercises showed that it represents flux distributions well for most realistic scenarios where error sources are compounded on each other, but has limitations in the Hermite polynomial approach for accurate representations of certain non-Gaussian distributions under highly controlled circumstances. This was discussed in §3.1.3.2.

As a result of this exercise, several issues were identified and corrected. Firstly, a field-wide efficiency calculation issue was identified and corrected. Previously, SolarPILOT reported the total cosine, shading, blocking, etc., losses as the mean loss across all heliostats for each specific loss mechanism. In fact, this approach ignores the relatively different energy impact of each loss mechanism depending on what has happened "upstream". If there is a large amount of shading loss, for example, then the subsequent losses have a less impact on lost power. The reported field-wide efficiency values are now weighted based on their energy contribution, and the product of all field-wide losses now equals the reported total efficiency. Additional details can be seen in the documentation of the problem on the SolarPILOT GitHub page[11]. Secondly, as with the SolTrace model for Buie sunshape, SolarPILOT's model was updated to include the correction calculation that ensures that the energy in the circumsolar region equals the specified fraction. Furthermore, the analysis showed that the Buie sunshape could be better represented in SolarPILOT with a truncation of the intensity function at an angle of 20 mrad from the centre of the sun. Including small values of non-zero intensity above this angle resulted in the excessive weighting of the circumsolar region by the Hermite polynomial fitting algorithm, so this empirical conclusion has improved the ability of SolarPILOT to model Buie sunshapes in most real-world cases, though the issues with zero-error heliostats remain. Thirdly, several improvements were made to the SolarPILOT interface, to the scripting language, and to the parametric simulation capabilities as a result of bugs and lacking features were noted during the exercise.

4.4 Tracer

Tracer, as an open source optical modelling tool in CSP, also benefited from this study. Tracer implementations of slope errors and sunshapes were validated by comparison with the state-of-art research tools. The highly readable Python language of Tracer provides a manageable platform for implementations of new algorithms for testing. While Tracer is accurate, its longer computational times and limitations in dealing with large heliostat fields were also noted. Efforts are being made to improve this ability.

4.5 Solstice

The present study provided a valuable comparison of different methods and software from which Solstice benefited through the numerical validation. Especially, this study leads to improvements in the post processing programs to better compute the breakdown of optical losses.

4.6 Heliosim

Heliosim initially had significant discrepancies with the reference solutions from Tonatiuh, especially for the off-axis test cases (rounds B and C). The deterministic heliostat ray casting model

11 Documentation of issues on SolarPILOT Github: https://github.com/NREL/SolarPILOT/issues/22

originally implemented in Heliosim (v4.0.3 and below) was identified as a source of error, and was replaced with a Monte Carlo model (v5.4.0). The original deterministic model cast a precalculated bundle of rays from a regular grid of points on the heliostat mirror surface. The ray bundle was calculated by the numerical convolution of the sunshape and slope error distributions for the on-axis reflection case, discretising the resultant intensity distribution with a regular grid of points in the azimuthal and zenith axes, and applying rotational transforms to account for off-axis effects. This approach was found to be both mathematically incorrect and computationally inefficient. The mathematical inaccuracy stemmed from the assumption that the intensity distribution for an off-axis reflected beam can be calculated by applying rotational transforms to the on-axis distribution, which is not possible for non-zero slope error. The computational inefficiency was due to the use of a deterministic model with regular discretisations of space and direction, which required large numbers of rays to provide a converged solution. Both of these problems were overcome with the implementation of a Monte Carlo model, where the incident direction, reflection point and surface normal for each ray cast from heliostat mirror surfaces is determined by sampling from the appropriate CDF. The Monte Carlo approach was found to require fewer rays to reach a converged solution, as the statistical sampling implicitly ensures more rays are cast in directions with higher energy density, whereas the deterministic approach casts an equal number of rays (but with different energies) in all directions.

Following the implementation of the Monte-Carlo ray casting model, slight discrepancies remained for the round B cases. This was found to be due to Heliosim automatically applying physically realistic geometric offsets between the heliostat actuation axes and the mirror surface, whereas the test case assumed that the origin points of the actuation axes and mirror surface coincide. An option was therefore added to the Heliosim software that allowed the offset distances to be specified by the user as some fraction of the heliostat characteristic length (i.e. mirror surface diagonal).

Despite good agreement then being found between Heliosim and Tonatiuh for rounds A and B, slight disagreement remained for round C. A critical difference between round B and C is the inclusion of shading and blocking effects, and therefore the possibility of error due to the treatment of shading and blocking in Heliosim was investigated. Previously Heliosim implemented approximations when simulating shading and blocking, where ideal sunshape (i.e. collimated) and perfect mirror (i.e. no slope error) models were assumed and the energy of each reflected ray was reduced by a heliostat-averaged shading and blocking factor. To check if these approximations were the source of error for round C, an option was added to the software to allow shading and blocking to be simulated without these approximations (i.e. sunshape and slope error models are considered, and each reflected ray has its own binary shading and blocking factors). The maximum error in overall efficiency due to the shading and blocking approximations was found to be less than 0.2% for round C, and the flux map discrepancies with Tonatiuh were not resolved. This discrepancy is to be investigated as future work by the Heliosim developers.

5. Conclusions

The sunshape and surface slope error models in six optical simulation tools are reviewed in three rounds of test cases. In the first test round, on-axis reflector–target configurations are applied and the sunshape and slope error are examined separately so that the radiance distribution can be obtained theoretically. Most of the tools showed good agreement with each other, except that (1)

Tonatiuh incorrectly implemented the pillbox slope error distribution, which is anticipated to be amended quickly in a later release; (2) SolarPILOT has limitations in simulations of a solo non-Gaussian distribution accurately (e.g. pillbox or Buie sunshape) due to simplification made in the Hermite polynomial expansion method, although these sunshape models can be used accurately for most real-world conditions where errors are compounded that lead to a Gaussian distribution; (3) In addition to the issue with SolarPILOT, slight differences are observed in the Buie sunshape results in the rest of the tools. It is caused by an issue of Buie's correlation (Buie et al., 2003) that was identified by different groups of researchers and solved independently. More thorough verification on this issue has been being performed and will be presented in future work.

The combinations of surface slope error and sunshape distributions for both individual heliostat and full field simulations are also compared under two sun positions (morning and solar noon). Good agreement was observed between Tonatiuh and Tracer. Instead of sampling sun rays from a sky that covers the whole heliostat field (e.g. in Tonatiuh, SolTrace and Tracer), Heliosim and Solstice both sample rays of the first intersection on the primary reflector surface, in such a way that accelerates the speed of simulation by avoiding generating wasted rays that hit the ground and avoiding the calculation of ray intercept locations on the heliostat mirror facets. They perform well for solar noon sun positions, however, discrepancies can be observed in predicting blockage and shading in morning sun positions. The cone-optics method (SolarPILOT) had the lowest accuracy due to its theoretical simplicity but has the merits of fast simulations (in seconds or fractions of a second). In the full field simulations, the flux distributions of the noon and morning cases obtained by SolarPILOT can differ by up to 9% and 15% respectively compared to those obtained by Tonatiuh.

The exercises of the three rounds of tests brought benefits to all the six optical modelling tools that were reviewed in this study. Through this exercise, discrepancies were identified and allowed to be revised to improve the quality of the tools. These improvements and some remaining issues were summarised in §4. It is our hope that this study will ensure better agreement and build confidence amongst CSP research on accurate modelling of the optical behaviour of solar concentrators. The details of each test case, parameter details and results data files are available online[1] for readers who are interested in repeating or extending these tests or applying them to other tools.

The aspects that are interesting but not covered in this study are listed below for further investigations: (1) verification on tracking mechanisms and tracking errors; (2) simulation in secondary concentrators (e.g. CPC); (3) accurate and faster blockage and shading simulation method; (4) CSR calibration in the Buie sunshape model.

Acknowledgements

This work is a collaboration between the Australian National University (ANU), CSIRO, NREL, PROMES-CNRS, Méso-Star Company and the Cyprus Institute. The authors gratefully acknowledge the support of (1) the European Union's Horizon 2020 research and innovation programme within the context of the Cyprus Institute's CySTEM ERA Chair project, under grant agreement No. 667942, (2) the French ‘‘Investments for the future” program managed by the National Agency for Research (ANR) No. ANR-10-LABX-22-01-SOLSTICE and (3) the Australian Renewable Energy Agency, 2014/RND010.

References

Arvo, J., Dutre, P., Keller, A., Jensen, H. W., Owen, A., 2003. Monte Carlo Ray Tracing. Siggraph.

Asselineau, C.-A., Zapata, J. and Pye, J., 2015. Integration of Monte-Carlo ray tracing with a stochastic optimisation method: application to the design of solar receiver geometry. Opt. Express.

Biggs, F., Vittitoe, C. N., 1979. The helios model for the optical behavior of reflecting solar concentrators. Tech. Rep. SAND76 - 0347, Sandia National Laboratory, Albuquerque, NM.

Blanc, P., Espinar, B., Geuder, N., Gueymard, C., Meyer, R., Pitz-Paal, R., Reinhardt, B., Renné, D., Sengupta, M., Wald, L., Wilbert, S., 2014. Direct normal irradiance related definitions and applications: The circumsolar issue. Solar Energy 110, 561–577.

Blanco, M., Mutuberria, A., Garcia, P., Gastesi, R., Martin, V., 2009. Preliminary validation of Tonatiuh. In the SolarPACES conference, Berlin, Germany.

Blanco, M., Mutuberria, A., Martinez, D., 2010. Experimental validation of Tonatiuh using the Plataforma Solar De Almería secondary concentrator test campaign data. In the SolarPACES conference, Perpignan, France.

Blanco, M., Mutuberria, A., Monreal, A., Albert, R., 2011. Results of the empirical validation of Tonatiuh at Mini-Pegase CNRS-PROMES facility. In the SolarPACES conference, Granada, Spain.

Box, G. E. P. and Muller, M. E, 1958. A Note on the Generation of Random Normal Deviates. Ann. Math. Stat. 29, 610–611.

Buck, R., 2012. Heliostat field layout using non-restricted optimisation. In the SolarPACES conference, Marrakech, Morocco.

Buie, D., Monger, A. G., Dey, C. J., 2003. Sunshape distributions for terrestrial solar simulations. Solar Energy 74, 113–122.

Caliot, C., Benoit, H., Guillot, E., Sans, J., Ferriere, A., Flamant, G., Coustet, C., Piaud, B. , 2015. Validation of a Monte Carlo Integral Formulation Applied to Solar Facility Simulations and Use of Sensitivities. Journal of Solar Energy Engineering, Vol. 137.

Collado, F., 2010. One-point fitting of the flux density produced by a heliostat. Solar Energy, Vol 84, 673–684.

Collado, F., Gomez, A., Turegano, A., 1986. An analytic function for the flux density due to sunlight reflected from a heliostat. Solar Energy, Vol. 37, pp. 215–234.

Delatorre, J. and Baud, G., 2014. Monte Carlo advances and concentrated solar applications. Solar Energy 203, 653–681.

Dellin, T. A. and Fish, M. J., 1979. User’s manual for DELSOL: A computer code for calculating the optical performance, field layout and optimal system design for solar central receiver plants. Tech. Rep. SAND79–8215, Sandia National Laboratory, Albuquerque, NM.

Dellin, T.A., 1979. An improved Hermite expansion calculation of the flux distribution from heliostats. Tech. Rep. SAND79–8619, Sandia National Laboratory, Albuquerque, NM.

Garcia, P., Ferriere, A., Benzian, J.J., 2008. Codes for solar flux calculation dedicated to central receiver system applications: A comparative review. Solar Energy 82(3), 189–197.

Grigoriev, V. and Corsi, C., 2017. Unified algorithm of cone optics to compute solar flux on central receiver. AIP Conference Proceedings, vol. 1850, p. 030021.

Ho, C.K., 2008. Software and codes for analysis of concentrating solar power technologies. Tech. Rep. SAND2008–8053, Sandia National Laboratory, Albuquerque, NM.

Jones E, Oliphant E, Peterson P, et al., 2001. SciPy: Open Source Scientific Tools for Python. http://www.scipy.org/.

Kim, J.-S., Burton, A., McGregor, J., Stein, W., Nakatani, H., 2013. Design and test of a 600kWt receiver for solar air turbine systems. In the SolarPACES conference, Las Vegas, USA.

Kistler, B.L., 1986. A user's manual for DELSOL3: A computer code for calculating the optical performance and optimal system design for solar thermal central receiver plants. Tech. Rep. SAND86–8018, Sandia National Laboratory, Albuquerque, NM.

Leary, P.L. and Hankins, J.D., 1979. User's Guide for MIRVAL-A Computer Coe for Modeling the Optical Behavior of Reflecting Solar Concentrators. Tech. Rep. SAND77–8280, Sandia National Laboratory, Livermore, CA.

Levêque, G., Bader, R., Lipiński, W., Haussener, S., 2017. High-flux optical systems for solar thermochemistry. Solar Energy 156, 133 - 148.

Li, L., Coventry, J., Bader, R., Pye, J., Lipiński, W., 2016. Optics of solar central receiver systems: a review. Opt. Express 24, A985 –1007.

Lipps, F., 1976. Four different views of the heliostat flux density integral. Solar Energy, Vol. 18, pp. 555–560.

Oliphant, T., 2006. A guide to NumPy. USA: Trelgol Publishing.

Osuna, R., Olavarria, R., Morillo, R., Sanchez, M., Cantero, F. et al., 2006. PS10, Construction of a 11 MW solar thermal tower plant in Seville, Spain. In the SolarPACES conference, Seville, Spain.

Potter, D., Burton, A., Kim, J.S., 2015. Optimised Design of a 1 MWt Liquid Sodium Central Receiver System. In Proceedings of 2015 Asia-Pacific Solar Research Conference.

Potter, D., Kim, J.-S., Khassapov, A., Pascual, R., Hetherton, L., Zhang, Z., 2017. Heliosim: An Integrated Model for the Optimisation and Simulation of Central Receiver CSP Facilities. In the SolarPACES conference, Santiago, Chile.

Pye, J., Coventry, J., Venn, F., Zapata, J., Abbasi, E., Asselineau, C.-A., Burgess, G., Hughes, G., Logie, W., 2017. Experimental testing of a high-flux cavity receiver. AIP Conference Proceedings.

Rabl, A., 1985. Active solar collectors and their applications. Oxford University Press, pp. 141.

Schwarzbözl, P., Pitz-Paal, R. and Schmitz, M. , 2009. Visual HFLCAL- A software tool for layout and optimisationof heliostat field. in Proceedings of the 15th SolarPACES Int. Symposium on Concentrating Solar Power andChemical Energy, Berlin, Germany, 2009.

Vittitoe, C.N. and F. Biggs, 1981. User's Guide to HELIOS. Tech. Rep. SAND81–1180, Sandia National Laboratory, Albuquerque, NM.

Wagner, M.J., Wendelin, T., 2018. SolarPILOT : A Power Tower Solar Field Layout and Characterization Tool. Solar Energy 171, 185–196 .

Walzel, M.D., Lipps, F.W., Vant-Hull, 1977. A solar flux density calculation for a solar tower concentrator using a two-dimensional Hermite function expansion. Solar Energy 19, 239 - 256.

Wang, Y., Asselineau, C.A., Coventry, J., Pye, J., 2016. Optical performance of bladed receivers for CSP systems. Proceedings of the ASME Power and Energy Conference.

Wang, Y., Potter, D., Asselineau, C.A., Corsi, C., Wagner, M., Blanco, M., Kim, J.S., Pye., J., 2017. Comparison of Optical Modelling Tools for Sunshape andSurface Slope Error. In the SolarPACES conference, Santiago, Chile.

Watkins, D., Thomas, D., Hetherton, L., Bolger, M., and Cleary, P., 2017. Workspace - a Scientific Workflow System for enabling Research Impact. In MODSIM2017, 22nd International Congress on Modelling and Simulation.

Wendelin, T., 2003. SolTRACE: A New Optical Modeling Tool for Concentrating Solar Optics. ASME 2003 Solar Energy Conference, Kohala Coast, HI.

Wendelin, T., Dobos, A., 2013. SolTrace: A ray-tracing code for complex solar optical systems. Tech. Rep. NREL/TP-5500–59163, National Renewable Energy Laboratory, Golden, CO.

Lipps, F. and Vant-Hull, L., 1978. A cellwise method for the optimization of large central receiver systems. Solar Energy, 20-6, 505–516.

Yellowhair, J., Christian, J. M., Ho, C. K., 2014. Evaluation of solar optical modeling tools for modeling complex receiver geometries. In Proceedings of the ASME 2014 8th International Conference on Energy Sustainability, Boston, Massachusetts, July.