New functionalities for the Tonatiuh ray-tracing software

Cite as: AIP Conference Proceedings **2033**, 210010 (2018); https://doi.org/10.1063/1.5067212
Published Online: 08 November 2018

João P. Cardoso, Amaia Mutuberria, Costas Marakkos, Peter Schoettl, Tiago Osório, and Iñigo Les

New Functionalities for the Tonatiuh Ray-tracing Software

João P. Cardoso[1,a)], Amaia Mutuberria[2,b)], Costas Marakkos[3,c)], Peter Schoettl[4,d)], Tiago Osório[5,e)] and Iñigo Les[2,f)]

[1]*Laboratório Nacional de Energia e Geologia, Estrada do Paço do Lumiar 22, 1649-038 Lisboa, Portugal.*

[2]*National Renewable Energy Center (CENER), Solar Thermal Energy Department, Ciudad de la Innovación 7, 31621 Sarriguren, Spain.*

[3]*The Cyprus Institute, 20 Konstantinou Kavafi Street, 2121, Aglantzia, Cyprus.*

[4]*Fraunhofer Institute for Solar Energy Systems (ISE), Heidenhofstrasse 2, D-79110 Freiburg, Germany.*

[5]*University of Évora - Renewable Energies Chair, Palácio Vimioso, Largo Marquês de Marialva, 7002-554 Évora, Portugal.*

[a)]Corresponding author: joao.cardoso@lneg.pt

[b)]amutuberria@cener.com

[c)]c.marakkos@cyi.ac.cy

[d)]peter.schoettl@ise.fraunhofer.de

[e)]tiagoosorio@uevora.pt

[f)]iles@cener.com

Abstract. Tonatiuh is an open source, freeware, Monte Carlo ray tracer suitable for CST applications, and is currently under further development to to increase and improve its functionalities. Work has recently been performed to implement the following functionalities: a flux distribution calculation utility; materials with incidence angle dependent optical properties; and the ability to import 3D geometries from CAD files. This paper provides a detailed account of these new functionalities, and the tests performed to establish their correct implementation in the new software version, Tonatiuh v 2.2.3.

INTRODUCTION

The development of Concentrating Solar Thermal (CST) technologies requires suitable tools to design, analyse and optimize optical systems. During the last decades, several software tools have been developed using convolution or ray tracing techniques [1, 2]. The detailed study of the solar flux in complex geometry systems, such as plants with cavity receivers or receivers with secondary concentrators, is mostly performed using software tools applying ray-tracing methods [2].

Tonatiuh is an open source, freeware, Monte Carlo ray tracer, able to use multi-threading computing, whose development is being performed by members of the CST Research and Technology Development (RTD) community, an effort which is currently led by CENER [3, 4]. The software is continuously being updated in order to increase and improve its functionalities. Under the European Union Seventh Framework Programme project STAGE-STE, a group of institutions, working in the development and study of point focusing technologies, reviewed currently available tools to design and optimize high concentration optical systems. The review helped to highlight a set of desirable functionalities for the new Tonatiuh version, from which three were added to the software: flux distribution calculation utility; materials with incidence angle dependent optical properties; the ability to import 3D geometries from CAD files. In the next sections the new utilities are described and scrutinized to confirm their correct implementation.

SolarPACES 2017
AIP Conf. Proc. 2033, 210010-1–210010-9; https://doi.org/10.1063/1.5067212
Published by AIP Publishing. 978-0-7354-1757-1/$30.00

FLUX DISTRIBUTION CALCULATION UTILITY

One of the limitations identified in previous versions of Tonatiuh was the absence of a post processing utility for analysing the various radiative flux parameters [2]. Instead, Tonatiuh users had to rely on external tools, such as Matlab®, Mathematica® or R, to post-process the results, which is a less user-friendly approach compared to having a built-in Tonatiuh utility.

The recently added Tonatiuh post-processing utility allows the user to readily visualize the radiative flux distribution on the modelled surfaces. Parameters such as incident solar power, minimum, maximum and average flux, uniformity and centroid location may now also be visualised in the Tonatiuh environment. This functionality applies to both flat and cylindrical surfaces.

The flux distribution utility divides the selected surface according to a bi-dimensional regular grid of equal area cells. The number of grid divisions in the width (I) and length (J) dimensions is user defined, allowing the definition of coarser or finer grids according to the user analysis requirements. For flat surfaces the grid is applied to the smallest rectangle enclosing the surface. Since in local coordinates a flat surface always lies in the $y = 0$ plane, the 3D impact position of photons hitting the 3D flat surface is transformed to the 2D rectangular grid following the transformation:

$$(u, v) = (x, z) \tag{1}$$

For cylindrical surfaces the grid is applied to the rectangle resulting from the unravelled surface of the cylinder starting from its generatrix. In local coordinates the cylinder lies in the XY plane with its axis parallel to the z axis. Thus a given photon hit point at the 2D rectangular grid is given by a suitable transformation of its 3D impact position (x, y, z):

$$(u, v) = (r\, arctan2(x, y), z) \tag{2}$$

where r is the cylinder radius and $Arctan2$ is the two argument arctangent function. It is then possible to count the number of photons intersecting each grid cell. With this knowledge one can compute the flux distribution and other related statistics.

The radiative flux incident on a given grid cell (i, j) is

$$\Phi_{i,j} = \frac{N_{i,j} P_{ph}}{A_c} \tag{3}$$

where $N_{i,j}$ is the number of photons intersecting the grid cell (i, j), P_{ph} is the power carried by each photon and A_c denotes the grid cell area. To display the flux map, a colour scale is established and a simple interpolation of the radiative flux between the grid cell centre and its neighbours is performed to smooth the plot.

The total incident power on the surface is given by

$$\dot{Q} = P_{ph} \sum_{i=1}^{I} \sum_{j=1}^{J} N_{i,j} \tag{4}$$

The minimum, maximum and average radiative flux in the surface is given respectively by

$$\Phi_{min} = \min_{i,j}(\Phi_{i,j}) \tag{5}$$

$$\Phi_{max} = \max_{i,j}(\Phi_{i,j}) \tag{6}$$

$$\bar{\Phi} = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} \Phi_{i,j}}{I\, J} \tag{7}$$

The coordinates for the maximum flux position are given as the centre coordinates of the grid cell where the maximum value was found. The centroid of the flux distribution map is given by

$$(u, v)|_{centroid} = \left(\frac{\sum_{i=1}^{I} \sum_{j=1}^{J} \Phi_{i,j} u_{i,j}}{\sum_{i=1}^{I} \sum_{j=1}^{J} \Phi_{i,j}}, \frac{\sum_{j=1}^{J} \sum_{i=1}^{I} \Phi_{i,j} v_{i,j}}{\sum_{j=1}^{J} \sum_{i=1}^{I} \Phi_{i,j}} \right) \tag{8}$$

A uniformity value is computed according to

$$uniformity = \frac{\sqrt{\frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J}(\Phi_{i,j} - \bar{\Phi})^2}}{\bar{\Phi}} \tag{9}$$

An error value is also computed by considering the difference between the maximum flux value and the one computed for a grid with one less division:

$$error = \frac{\left|\Phi_{max} - \Phi^*_{max}\right|}{\Phi_{max}} \tag{10}$$

where Φ^*_{max} is the maximum flux value obtained for a $I-1 \times J-1$ grid.

The graphic user interface (GUI) of the new functionality can be seen in figure 1.

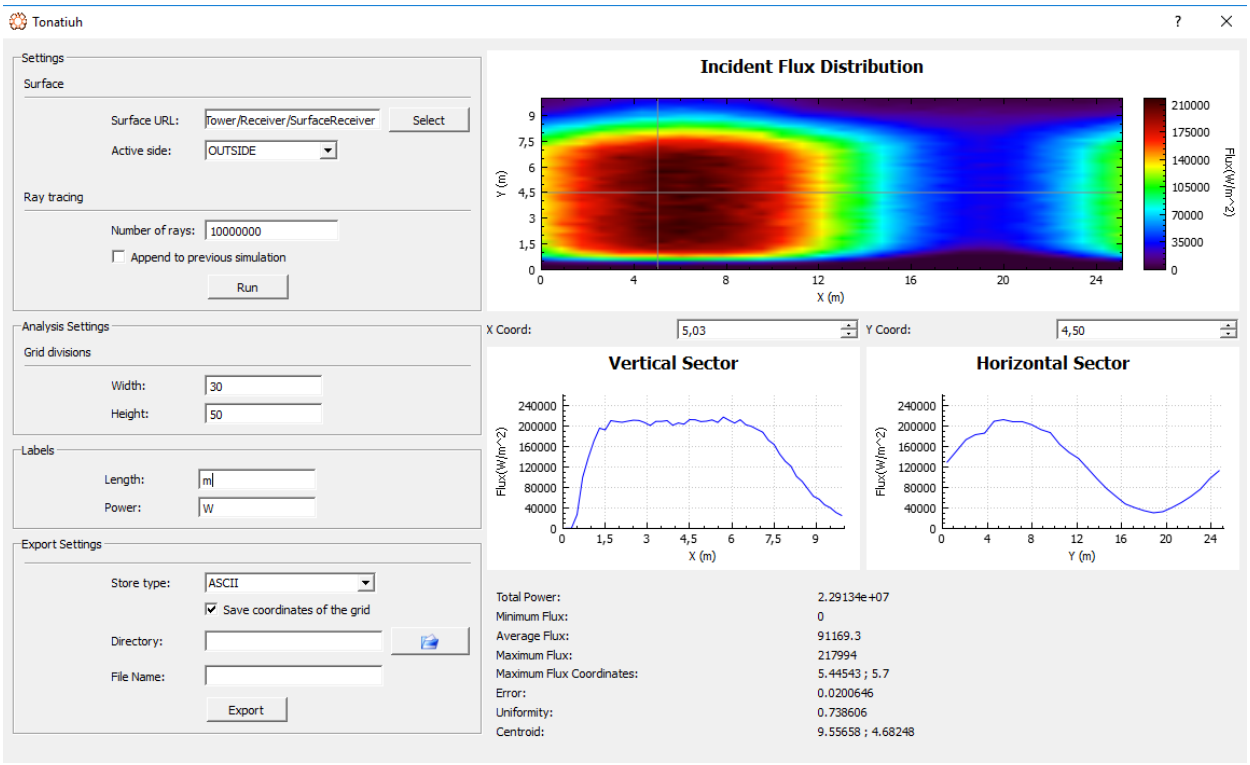

FIGURE 1: Graphic user interface of the flux distribution calculation utility.

Besides presenting the surface flux distribution map and the parameters previously defined, the new tool GUI allows the user to export the flux distribution map (as an image or a numerical table with the cell centre coordinates and respective radiative flux value) and to analyse the 1D flux distribution profile for user defined sections along the width or length direction.

Comparison with Standard Approach

The results obtained with the new functionality were compared with the results obtained when processing Tonatiuh photon maps with external tools in terms of the most representative parameters, namely total power and minimum, maximum and average flux. The comparison was performed for two different solar systems: a parabolic dish and a parabolic trough collector (figure 2). The tests were performed for a pillbox sunshape with θ_{max} = 0 and DNI = 1000 W m^{-2}, with 20 million rays cast for each test. Different grid divisions were tested since some of the parameters under analysis are sensible to the number of grid divisions. Table 1 presents the relative result obtained with the new utility in

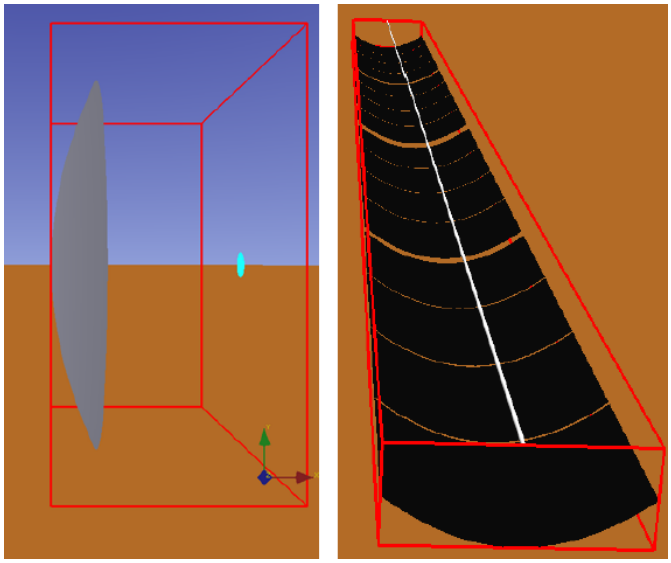

FIGURE 2: Tonatiuh GUI view of the systems used when testing the flux distribution utility: a) parabolic dish; b) parabolic trough.

relation to the values obtained with the standard approach (in this case processing the photon map with a Mathematica 10 script).

Table 1 shows the results obtained with the new flux distribution calculation utility are very close to the ones obtained using an external post-processing tool. Parameters whose computation does not depend on the grid (such as the total power) present equal values independently of the tool used to process the photon maps. However, the flux parameters whose computation depends on the grid (such as minimum, maximum and average flux) present relatively small deviations between the tools.

TABLE 1: Results obtained by the new functionality relative to the external processing of the photon map with a Mathematica 10 script for different grid divisions.

	Parabolic dish			Parabolic trough		
	20x20	40x40	80x80	20x20	50x50	100x100
Total power [%]	100.00	100.00	100.00	100.00	100.00	100.00
Minimum flux [%]	100.00	100.00	100.00	97.73	96.00	105.00
Average flux [%]	99.46	99.46	99.46	100.00	100.00	100.00
Maximum flux [%]	100.12	100.06	100.01	99.66	99.81	98.93

INCIDENCE ANGLE DEPENDENT MATERIALS

According to the electromagnetic theory, when an electromagnetic wave hits the interface between two homogeneous media it will be reflected and/or transmitted. Additionally, the transmitted wave will be attenuated due to absorption if it propagates through an absorbing medium. The fraction of the wave being reflected, transmitted and absorbed will depend not only on media properties, but also on the properties of the wave, namely on its wavelength, polarization and incidence angle.

In Tonatiuh surface properties are defined in terms of a material definition. Until now, Tonatiuh material plug-ins only considered properties at normal incidence. Thus, the amount of light being reflected or transmitted was independent of the incidence angle. Such approach demanded from the user the use of a suitable average value for the

entire range of possible incidence angles. Although such approach can be valid for some applications, it hinders the development of detailed simulations for systems were non-normal incidence is relevant.

New material types were implemented enabling the definition of angle dependent properties. The new material type does not account for polarization and wavelength dependencies, assuming that the user will provide the surface properties according to a suitable average in terms of both polarization and wavelength. The angle dependent material properties are introduced as a tuple (angle, value) by the user. The GUI of this new material type can be seen in figure 3.

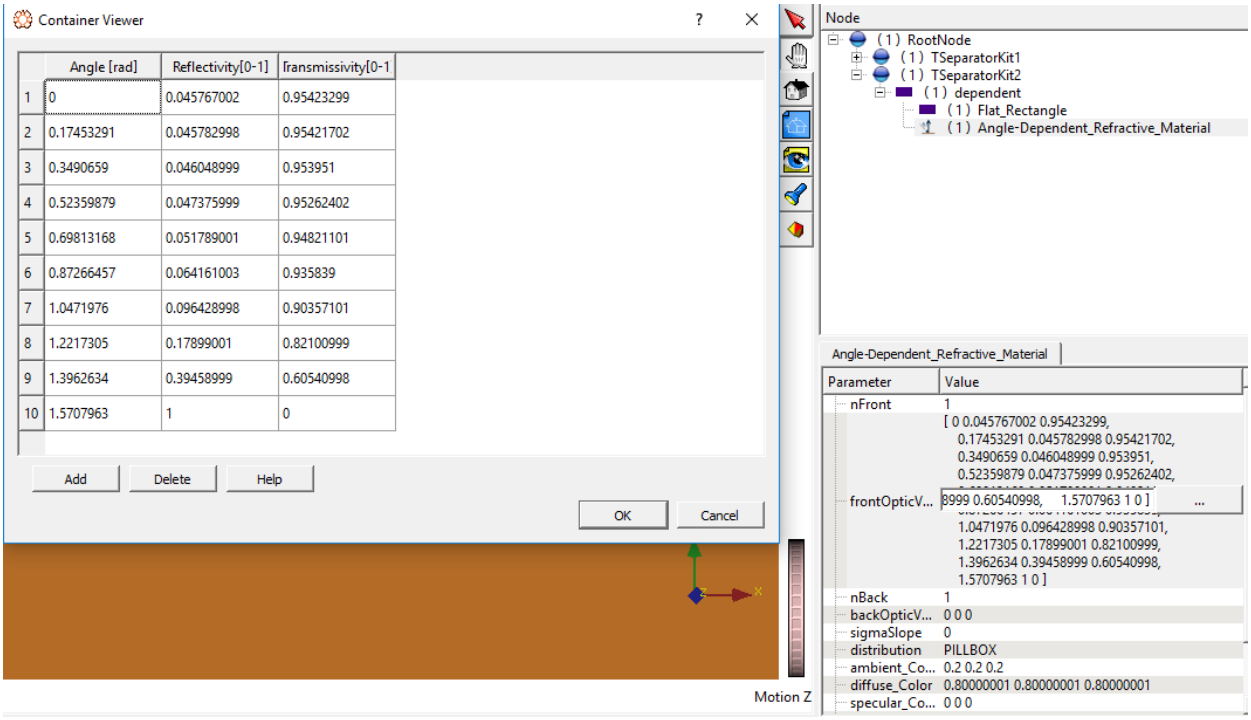

FIGURE 3: Graphic user interface of the new material type with angle dependent surface properties.

Since the material properties are described in a discretized way, it is necessary to interpolate in order to compute a given material property for incidence angles between the pairs of values defined by the user. These interpolations are currently performed using a linear interpolation scheme.

According to the new material types, the user is free to define the property values for any angle, thus non equidistant nodes can be defined. A pre-computation is performed in order to reduce the search to find the required pair of nodes to interpolate to an O(1)-operation, generating a new grid equal or smaller than the one introduced by the user. Since this computation is performed only once, at the simulation start, the global time spent in the interpolation step is reduced.

Two new material types were created, one considering an opaque reflector/absorber and another for refractive materials.

Validation

A set of simulations was performed to check the correctness of the results obtained with the new material types by checking if the material optical properties are being correctly computed by the software for different incidence angles. These simulations used two equal flat rectangular surfaces placed on top of each other with a slight distance between them. The top surface contained the new refractive material where a coherent beam of light is directed to at different angles. Since Tonatiuh provides the number of intersections occurring for each surface, it is possible to compute the number of rays transmitted and reflected by any given surface. In these simulations the transmittance is computed as the ratio between the number of rays incident in the lower rectangular surface and the number of rays reaching the upper surface. For non absorbing surfaces the reflectance is the symmetric value of the transmittance.

Figure 4 presents the results obtained for a set of simulations considering the reflectance and transmittance of quartz glass. The experimental curves for both transmittance and reflectance are displayed, as well as the values computed by Tonatiuh for a given set of incidence angles. It is possible to observe a good coincidence between the values computed by Tonatiuh and the experimental curves for the zones with small variations of the properties, which in this case correspond to low incidence angles. When the variations are larger the results obtained by the software present larger deviations from the experimental curves. This can be explained by the interpolation scheme between the user defined data points. Indeed, if one connects the user defined data points with straight lines (also displayed in the figure) it observes that the values computed by Tonatiuh fall within these lines, being computed as expected.

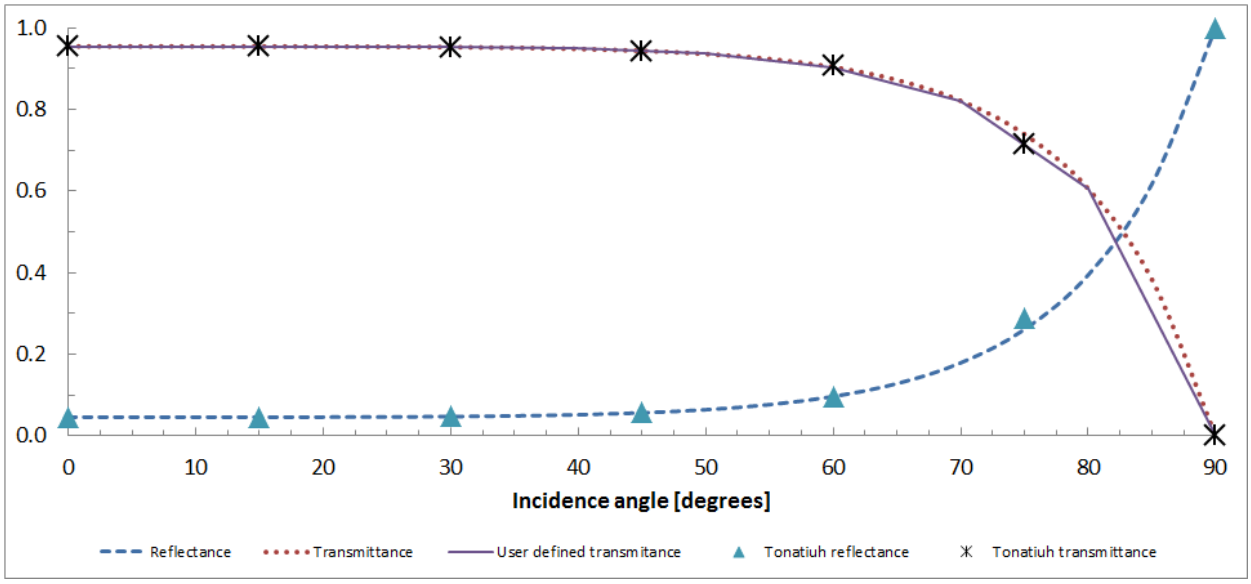

FIGURE 4: Reflectance and transmittance of quartz glass as defined by the user and computed by Tonatiuh for different incidence angles.

When performing simulations, and in order to ensure suitable results, the user should ensure that the grid (of tuples) describing the material properties should be fine enough to depict the incident angle dependency in an appropriate way, particularly in the presence of large and fast variations. Additionally, in future versions of the software, more complex interpolation schemes could be included (e.g. splines) to allow for more accurate representations of the original data.

COMPUTER AIDED DESIGN INPUT

So far, complex geometries could be defined by the Tonatiuh user through analytical equations (e.g.: Bezier surfaces), which can be a cumbersome task. However, in most cases, engineers and researchers use Computer Aided Design (CAD) tools to design CST components. Thus, the possibility to directly import surfaces designed using CAD software is a significant work flow simplification, improving the usability and user-friendliness of the ray-tracing software.

The great variety of CAD software currently available results in a large variety of CAD file formats. The majority of these are vendor dependent, complicating the exchange of CAD information between different CAD tools. However, there are vendor neutral CAD files (such as STL, IGES and STEP) which are usually supported by the majority of CAD software. The simplest of these formats is the STL (Stereolithography) format, which uses triangular tesselations to describe 3D surface geometries. Additionally, the possibility of exporting information in STL format is widespread amongst CAD software.

A new functionality (*CAD_shape*) enabling Tonatiuh to import 3D surfaces from CAD files was implemented, enabling the user to import ASCII or binary STL files. In this format, the 3D geometry is defined by means of a triangular tesselation, each facet consisting of three vertices and an outward pointing normal vector [5].

Performing ray-tracing with surfaces imported using the new CAD surface import functionality requires efficient algorithms to deal with the large number of triangular surfaces and the subsequent large number of ray-triangle intersection tests that must be performed. This then dictates the introduction of a new ray-triangle intersection algorithm.

Ray-triangle intersection is a paramount operation in modern ray-tracing rendering for computer graphics applications, resulting in a large body of literature devoted to this subject. The choice of a suitable intersection algorithm is dependent on the specific implementation (e.g.: using the Graphics Processing Unit (GPU) versus using the Central Processing Unit (CPU); use of back-facing culling; use of pre-computed data). According to [6], the Möller-Trumbore [7, 8] algorithm is generally considered to be the fastest for simple CPU implementations. However, the application of some acceleration techniques, such as back-face culling or the use of pre-computed data, can render the Jiménez algorithm [9] faster than the Möller-Trumbore for CPU implementations, although not significantly. Tests performed by the same authors show the Möller-Trumbore for GPU implementations to be the fastest algorithm in all tested cases [6]. Since refraction must be considered in many optical ray-tracing applications, back-face culling algorithms cannot be considered since back-face culling discards intersections of the rays with the inner face of the triangle which will occur during refraction events. Thus Tonatiuh's new ray-triangle intersection algorithm was implemented following the Möller-Trumbore algorithm.

The use of a suitable set of accelerating techniques to improve the speed performance of the ray-tracer is also recommendable, since the simple application of a ray-triangle intersection test implies testing all the mesh triangles at each step, which leads to high computation times when the number of triangles is large. A possible solution to avoid such extensive testing is to use spatial data structures to group the triangles within a given boundary, testing only the set of triangles inside the boundary pierced by the ray. Such approach significantly reduces the number of ray-triangle intersections at the cost of adding intersection tests for the traversal of the boundaries (which should result in less and cheaper tests). Several approaches have been proposed in the literature [10]. However, it is not clear which structure is the best since the results depend on several factors, including the geometry of the problem and the implementation [11, 12]. It was decided to implement a Bounding Volume Hierarchy, because it is a suitable method that appears to be the standard within the computer graphics community and is relatively simple to implement, having lower memory requirements by comparison with space-oriented partition schemes.

Comparison with Standard Approach

Ray-tracing simulations using receivers described by CAD surfaces were performed for two different CST technologies: a parabolic dish using a flat disc receiver and a central receiver system using an external receiver (figure 5). Within the same simulation standard surfaces, analytically described, were superimposed as virtual surfaces. For each case the photon intersections at each surface and the corresponding radiative flux were computed. Table 2 presents the relative difference of the total radiative flux on the receiver computed using surfaces described by analytical equations or by an STL surface.

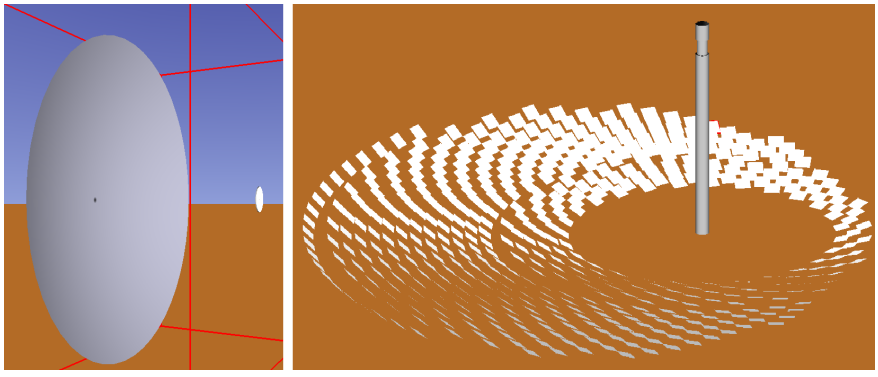

FIGURE 5: Tonatiuh GUI view of the systems used when testing the CAD import utility: a) parabolic dish; b) central receiver system.

For both cases the difference is very low. When considering a flat surface, such as the flat disc used in the simulation with the parabolic dish, there is no difference between using both surfaces. However, when using non-flat surfaces, such as the case of the simulation with the central tower receiver, there are slight differences between the

TABLE 2: Relative difference of the total radiative flux impinging on the system receiver between an STL surface and another described by analytical equations.

Cast rays [x10^6]	Parabolic dish	Central tower receiver
1	0.000%	0.076 %
20	0.000%	0.061 %
50	-	0.073 %

value computed using a surface imported with the new CAD utility (i.e. a surface described by a triangular tesselation) and the analytical surfaces previously implemented in Tonatiuh.

CONCLUSIONS

A set of limitations have been identified in the operation of the Tonatiuh software, namely the absence of a flux distribution calculation utility, materials with angle dependent optical properties and a CAD surface import utility.

A new Tonatiuh utility was developed and tested, enabling the direct computation and plotting of the flux distribution for planar and cylindrical surfaces, as well as additional information such as the total power incident on the surface, the maximum, minimum and average radiative flux on the surface, the surface position of the maximum radiative flux, the centroid of the flux distribution map and the uniformity and error values.

New material types were implemented and tested, allowing the simulation of concentrating solar thermal collectors composed of materials having angle dependent properties. The tests show they behave as intended, providing reliable results.

A CAD surface import utility was added, allowing Tonatiuh users to import CAD files of the STL format. This utility was tested, being concluded that it is correctly implemented, proving to give valid results for suitable STL meshes.

In the future additional activities should be planned to further test and enhance these new functionalities.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement n^o 609837, project STAGE-STE.

REFERENCES

1. P. Garcia, A. Ferriere, and J. Bezian, Sol. Energy **82**, 189–197 (2008).
2. N. C. Cruz, J. L. Redondo, M. Berenguel, J. D. Álvarez, and P. M. Ortigosa, Renew. Sust. Energ. Rev. **72**, 1001–1018 (2017).
3. M. J. Blanco, J. M. Amieva, and A. Mancillas, "The tonatiuh software development project: an open source approach to the simulation of solar concentrating systems," in *ASME 2005 International Mechanical Engineering Congress and Exposition* (American Society of Mechanical Engineers, Orlando, FL, USA, 2005), pp. 157–164.
4. M. J. B. Blanco, A. Mutuberria, A. Monreal, and R. Albert, "Results of the empirical validation of tonatiuh at mini-pegase cnrs-promes facility," in *Proceedings of SolarPACES 2011 Conference* (Granada, Spain, 2011).
5. A. Mutuberria, I. Les, P. Schöttl, J. Cardoso, C. Marakkos, A. Flores, and D. Fontani., "First version of simulation tool for design and optimization of high concentration optical systems," Tech. Rep. (Scientific and Technological Alliance for Guaranteeing the European Excellence in Concentrating Solar Thermal Energy - STAGE-STE project, 2015).
6. J. J. Jiménez, C. J. Ogáyar, J. M. Noguera, and F. Paulano, "Performance analysis for GPU-based ray-triangle algorithms," in *GRGRAPP 2014: 9th International Conference on Computer Graphics Theory and Applications* (Lisboa, Portugal, 2014).

7. T. Möller and B. Trumbore, J. Graph. Tools **2**, 21–28 (1997).
8. R. J. Segura and F. R. Feito, Journal WSCG **9**, 76–81 (2001).
9. J. J. Jiménez, R. J. Segura, and F. R. Feito, Comput. Geom. **43**, 474–492 (2010).
10. A. Y. Chang, "A survey of geometric data structures for ray tracing," Tech. Rep. TR-CIS-2001-06 (Department of Computer and Information Science, Polytechnic University, USA, 2001).
11. I. Wald, S. Boulos, and P. Shirley, ACM T. Graphic **26** (2007).
12. A. Áfra and L. Szirmay-Kalos, Comput. Graph. Forum **33**, 129–140 (2014).