Computationally Fast Analytical Ray-Tracer for Central Receiver Systems

Pascal Richter[1, a)], Florian Hövelmann[1]

[1]*RWTH Aachen University, Department of Computer Science, Ahornstr. 55, 52074 Aachen, Germany.*

[a)]Corresponding author: pascal.richter@rwth-aachen.de

Abstract. An accurate and computationally fast ray-tracer is the key part for the simulation and optimization of the optical irradiation of a heliostat field layout in a solar central receiver system. Within this work we present an analytical ray-tracer which is fast in runtime while obtaining highly accurate results. The runtime improvement is achieved by a faster integration method that does not require a discretization of the receiver. This allows for discretizing the heliostat surface into smaller cells each having a representative flux function to better account for a variety of optical errors. Our new ray-tracer is implemented on the same C++ platform as other existing ray-tracers, such that reasonable cross validation with direct run-time comparisons are possible. Within a case study we demonstrate that the new convolution method decreases the run-time by a factor of 20 compared to HFLCal, and a factor of three compared a bidirectional Monte-Carlo ray-tracer, while achieving a stable accuracy of 99.98 %.

INTRODUCTION

There exist several different ray tracing techniques to compute the solar irradiation at the receiver surface, which all have their benefits and drawbacks in accuracy and runtime. The ray tracing techniques can be classified into two groups: First, the Monte Carlo ray-tracers as Tonatiuh [1], SolTrace [2] and STRAL [3] where the randomized rays are used to compute the solar power at the receiver. The second group are the analytical ray-tracers as HFLCaL [4] and UNIZAR [5] which represent the reflected solar flux of a heliostat on a plane orthogonal to the reflection direction, called image plane, by a two-dimensional function, as a circular Gaussian distribution or an error function. The incoming solar power at the receiver is then calculated by integrating over an area representing the receiver. This is commonly done numerically using some kind of quadrature rule, thus discretizing the receiver into smaller pieces. Monte-Carlo ray-tracers tend to be more accurate than analytical ray-tracers as they replicate real photon interactions. But due to the large number of rays required to achieve accurate and stable results, they have a higher runtime than analytical ray-tracers which commonly use one flux function for a whole heliostat facet.

The overall principle of both groups of ray-tracers is similar. For the sake of simplicity, we will concentrate on the bidirectional ray-tracers where the rays are generated on the heliostat's surface instead of on a plane above it. So, the heliostat's surface gets discretized into several cells with an area of A_{cell}. Here, the analytical ray-tracers are using the whole heliostat facet as a cell. A ray is placed in the center of each cell and reflects the direct normal irradiation I_{DNI} from the sun towards the receiver, while taking a variety of effects into account, see Fig 1. The shape of the sun results in an uncertainty of the solar ray $\vec{\tau}_{solar}$. Since the heliostat is not a perfect mirror and might deviate to its intended alignment, its normal also fluctuates. These errors are referred to as the slope and tracking error, respectively. Altogether they lead to a deviation of the perfect reflected ray. The portion of rays potentially hitting the receiver without considering other losses determines the intercept efficiency. Besides that, the heliostat can also be shaded by other heliostats or the reflected ray can get blocked such that the ray does not reach the receiver. Moreover, the reflective area of the heliostat is reduced by the cosine efficiency η_{cos}, its mirror only reflects a certain amount of solar power specified by the reflective efficiency η_{ref}, and power loss as the ray travels through the air given by the atmospheric attenuation η_{aa}. In respect to the optical losses the representative ray power P_{ray} of the current heliostat cell is given by $P_{ray} = I_{DNI} \cdot A_{cell} \cdot \eta_{cos} \cdot \eta_{ref} \cdot \eta_{aa}$.

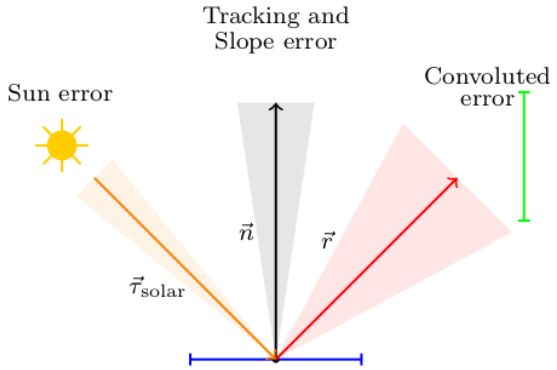

FIGURE 1. Illustration of the ray disturbance and its causes. The heliostat is shown in blue and the receiver in green.

Besides the described heliostat blocking and shading, the tower is also casting a shadow and certain receiver types can block incoming rays. Blocking and shading calculations are almost equally for each ray tracer. For the Monte Carlo methods, a perturbed version of the representative ray is evaluated whereas analytical ray-tracers use the ideal reflected ray. The ray is then traced, i.e., it is checked for intersections with potential blocking heliostats. Moreover, to encounter shading effects, intersections of the incoming sun ray with potential shading heliostats as well as the tower are evaluated.

To accelerate this process, for each heliostat a set of potentially shading or blocking heliostats is precomputed by using a simplified representation of the heliostats and rays. With this preselection, a single ray just needs to evaluate against this list of objects instead of testing all heliostats in the field.

An accurate computation of the shading and blocking effects is one of the main reasons for a large number of needed rays for the Monte Carlo ray-tracer. On the other hand, a problem with existing analytical ray-traces is, that so far just one ray (or flux) represents a whole facet of a heliostat. This causes inaccuracies as the ideal reflected ray in the facet center either is blocked/shaded, or not. Thus, a partial blocking or shading of a facet is often not considered.

As a last step for all non-blocked/shaded rays, the ray-tracer is calculating interceptions with the receiver. The Monte Carlo ray tracing methods are utilizing the straightforward approach of computing whether the perturbed ray is intersecting the receiver or not. Since this is a common task in the field of computer graphics, it is accelerated a lot by existing methods such as using axis-aligned bounding boxes for the receiver. When enough rays are generated the ray disturbance is accurately represented. Furthermore, as ray intersections can easily be calculated for different geometry, Monte Carlo ray tracing methods are suitable for complex cases where analytical techniques are not applicable anymore. However, considering the computational effort involved in simulating a vast number of rays, these methods have a comparatively long processing time. Analytical ray-tracers, on the other hand, calculate the intercept efficiency by integrating the flux function over an area that represents the receiver. This can also be seen as computing the probability of the ideal reflected ray being perturbed in such a way that it intersects the receiver. Multiplying the probability with the ray power then gives the incoming power at the receiver. Therefore, flux functions consist of a two-dimensional probability density function that is scaled by a factor representing the total power reflected by the heliostat. Since these functions are often defined on a plane orthogonal to the ideal reflected ray, the receiver needs to be mapped onto the plane. To integrate over the resulting area most existing ray-tracers are using some kind of quadrature rule meaning that the receiver surface is discretized into smaller pieces. However, these types of integration methods are computational expensive which is one reason why analytical ray-tracer use one ray per facet. In the following we present our convolution ray-tracer which addresses these issues.

CONVOLUTION RAY-TRACER

Our new Gaussian convolution method aims to deliver a higher precision for the computation of the received solar power on the receiver. This is achieved by rasterizing the heliostat's surface in a number of cells and using an

accelerated method to evaluate each cell. Similar to HFLCal [9] and based on the idea of Rabl [8], we model the sun error with a Gaussian distribution. However, in contrast to HFLCal our ray-tracer can model different tracking and slope errors in vertical and horizontal direction. With the Central Limit Theorem these Gaussian distributions are convoluted to two Gaussian distribution with standard deviations σ_{hor} and σ_{vert} for the horizontal and vertical direction, respectively. The horizontal and vertical directions are defined relative to the cell and its normal and the convoluted deviation directions are obtained using a rotation [7]. As two separate Gaussian distributions are used to describe horizontal and vertical deviation of the perfect reflected ray, our two-dimensional probability density function is given by

$$f(x,y) = \frac{1}{2\pi\sigma_{hor}\sigma_{vert}} \cdot exp\left(\frac{-1}{2}\left(\frac{x^2}{\sigma_{hor}^2} + \frac{y^2}{\sigma_{vert}^2}\right)\right) \tag{1}$$

Besides a better representation of photon interactions, the fine discretization also improves accounting for blocking and shading effects. After accounting for blocking and shading effects, the power at the receiver from each heliostat cell is then calculated in two steps. First the area corresponding to the receiver on the image plane where the bivariate Gaussian distribution is defined is computed, see Fig. 2.

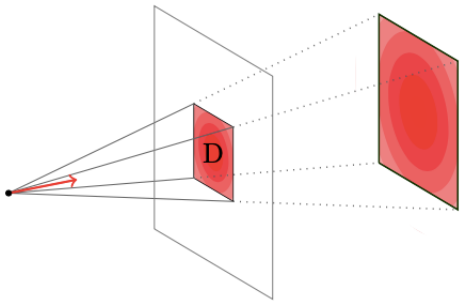

FIGURE 2. Sketch for the perspective projection of the receiver area onto the plane where the bivariate Gaussian distribution is defined.

Thus, we need to compute the integral over each cell area *D*,

$$P_{\text{int}}(D) = \iint_D f(x,y)\ dA \tag{2}$$

The area is obtained by a perspective projection of the receiver onto the image plane. Projecting a corner of the receiver requires only one matrix multiplication and gives the projected point in local coordinates of the image plane. A perspective projection results in an exact representation of the receiver for the case of a perturbed version of the ideal reflected ray, because if such a ray intersects the area it will also hit the receiver. To also account for tower blocking in the case of a cavity receiver, the polygonal receiver represenation gets cut by computing its intersections with two lines representing the blocking boundaries of the receiver [7]. Now, on the resulting polygon we need to evaluate the integral of the bivariate Gaussian distribution. Instead of using a quadrature rule to solve the integral as it is commonly done our ray-tracer directly integrates over the polygon in an efficient and accurate manner with a method developed by Di Donato et al. [6] which got extended in [10] to handle arbitrary polygons. Thus, the algorithm evaluates the integral over the complementary region $\acute{D}$, i.e., over the outer region of the polygon. As the bivariate Gaussian distribution *f(x,y)* is a probability density function it holds that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\ dxdy = 1 \Rightarrow P_{\text{int}}(D) = 1 - P_{\text{int}}(\overline{D}). \tag{3}$$

To compute $P_{\text{int}}(\acute{D})$ the algorithm divides the region $\acute{D}$ into angular regions A_i as illustrated in Fig. 3 and relies on a fast method to evaluate the integral over those regions. For each vertex V_i of the Polygon a corresponding

angular region A_i is introduced. The angular regions A_i are defined as the semi-infinite part bounded by two intersecting lines [6] as shown in Fig. 3. Before calculating the integral over these regions, the integrand of $P_{\text{int}}(D)$ is reduced by substitution to

$$P_{\text{int}}(D) = P'_{\text{int}}(D') = \frac{1}{2\pi} \iint_{D'} \exp\left(-\frac{u^2+v^2}{2}\right) dudv,\ u = \frac{x}{\sigma_x},\ v = \frac{y}{\sigma_y}. \tag{4}$$

Note that the polygon D′ also has to be given in coordinates of u and v and thus is obtained by using the same substitution. The corresponding coordinate transformation matrix is added to the transformation matrix used to project the receiver such that the polygon is directly given in coordinated of u and v . With this substitution a circular symmetric bivariate Gaussian distribution is reached. This property is required in order to calculate the integral over an angular region A_i, such that

$$P'_{\text{int}}(\overline{D}') = \sum_{i=1}^{n} P'(A'_i) = \sum_{i=1}^{n} \frac{1}{2\pi} \iint_{A'_i} \exp\left(-\frac{u^2+v^2}{2}\right) dudv. \tag{5}$$

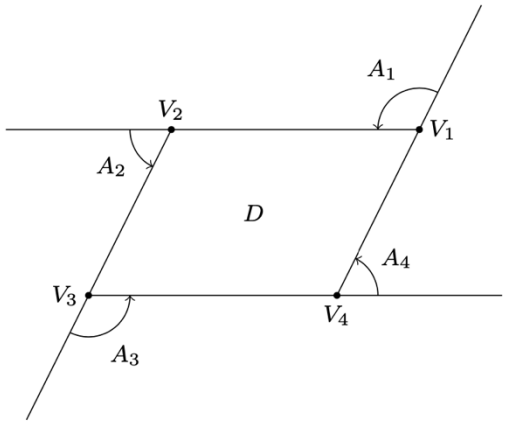

FIGURE 3. Angular regions of a polygon with four vertices [7].

Due to the circular symmetry, a rotation of the axes can be performed without changing the result. Therefore, the axes can be rotated such that the line L of an angular region as shown in Fig. 4 coincide with the positive u-Axis [6]. The line L is defined by the Vertex V and the origin. Its corresponding angular region is described using the distance from the origin to the vertex V and the angles θ_1 and θ_2. From now on the axes are assumed to be rotated but are still referred to as u and v.

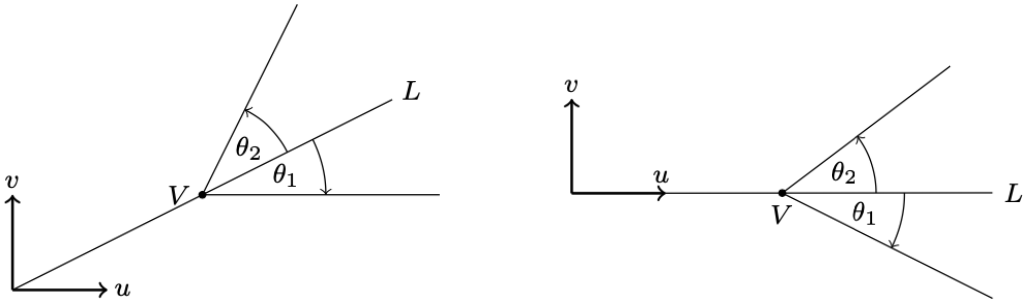

FIGURE 4. General angular region before (left) and after the rotation (right) [7].

Using polar coordinates centered at the vertex V with the distance R from the origin, we get

$$P'(A_i') = \frac{1}{2\pi}\int_{\theta_1}^{\theta_2}\int_0^{\infty}\exp\left[-\frac{1}{2}\left(R^2 + 2rR\cos\theta + r^2\right)\right] r dr d\theta$$
$$= \frac{1}{2\pi}e^{-\frac{R^2}{2}}\int_{\theta_1}^{\theta_2}\int_0^{\infty} r e^{-\frac{r^2}{2}}e^{-pr}\, dr d\theta,\ p = R\cos\theta. \tag{6}$$

This equation can be solved using integration by parts on the inner integral. For the case of $|\theta_1| \leq \frac{\pi}{2}$, and $|\theta_2| \leq \frac{\pi}{2}$ we get

$$P'(A_i') = \frac{e^{-\frac{R^2}{2}}}{\pi}\left(\frac{\theta_2 - \theta_1}{2} - \sum_{k=1}^{K} a_k J_{k+1}\right) \tag{7}$$

with factors a_k as the minmax polynomial fit of $\mathrm{erfc}(w)/z(w)$, and $w = \frac{p}{\sqrt{2}}$.

$$\begin{cases} J_0 & = \theta_2 - \theta_1 \\ J_1 & = \frac{R}{\sqrt{2}}(\sin\theta_2 - \sin\theta_1) \\ J_{k+1} & = \frac{1}{k+1}\left(\left(\frac{R}{\sqrt{2}}\right)^{k+1}\left[\cos^k\theta\sin\theta\right]_{\theta_1}^{\theta_2} + k\frac{R^2}{2}J_{k-1}\right) \\ & = \frac{1}{k+1}\left[\left(h_2 g_2^k - h_1 g_1^k\right) + k\frac{R^2}{2}J_{k-1}\right],\ k \geq 2 \end{cases} \tag{8}$$

where

$$g_i = \frac{R}{\sqrt{2}}\cos\theta_i,\ h_i = \frac{R}{\sqrt{2}}\sin\theta_i,\ i = 1, 2. \tag{9}$$

For angles outside of the range of $|\theta_1| \leq \frac{\pi}{2}$, and $|\theta_2| \leq \frac{\pi}{2}$ the angular regions are transformed to the base case, for more information we refer to [6,7]. The fitting of $\mathrm{erfc}(w)/z(w)$ has been done with different polynomials depending on the desired accuracy of $P'(A_i')$, thus offering a trade-off between run-time and accuracy. It has been proven that $P'(A_i')$ can be evaluated with an accuracy of up to 12 decimal digits [6]. With all this, $P'(A_i')$ can be evaluated and thus the desired probability $P_{\mathrm{int}}(D)$ calculated. In combination with the perspective projection of the receiver, an almost exact of the ray disturbance of the ideal reflected ray.

Altogether, we have three extensions to the state-of-the-art analytical ray-tracers: rasterization of the heliostat surface to allow an adjustable tradeoff between accuracy and runtime, perspective projection of receiver, and fast and accurate evaluation of the integral over a polygon.

CASE STUDY

In a large case based on the PS10 we compared our new ray-tracer against a bidirectional Monte Carlo ray tracing method and an implementation of HFLCal capable of simulating multiple rays per facet. To see how the number of rays influences the accuracy of the ray-tracers, different numbers of rays per facet were simulated ranging from one to eight rays per facet for the HFLCal comparison and 12 to 126 rays per facet for Monte Carlo. The results are normalized by the average result of the bidirectional Monte Carlo method simulating ten million rays twenty times. For HFLCal the receiver was discretized into 400 pieces. A comparison of the accuracy against the runtime is given in Fig. 5. For the bidirectional Monte Carlo ray-tracer each setup was simulated 20 times and the highest and lowest 5% results got cut off. The fluctuations of the results are illustrated by the shaded region.

As shown in Fig. 6, our convolution ray-tracer is more than 20 times faster than HFLCal especially when simulating multiple rays per facet. Moreover, using 10 thousand receiver pieces the differences in their results went down to less than 0.02 %. In Fig. 7 it can be seen that our method is also capable of achieving highly accurate results that are always within the boundaries of the bidirectional Monte Carlo method. Furthermore, our ray-tracer converges faster and does not fluctuate.

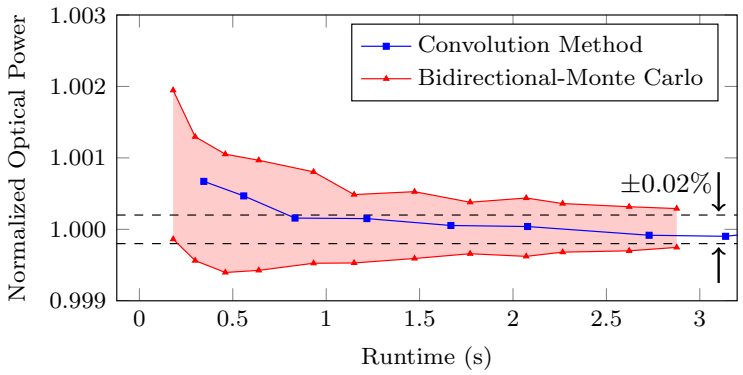

FIGURE 5. Accuracy vs. runtime of our convolution method and the bidirectional-Monte Carlo ray-tracer, using the PS10 test case with 624 heliostats.

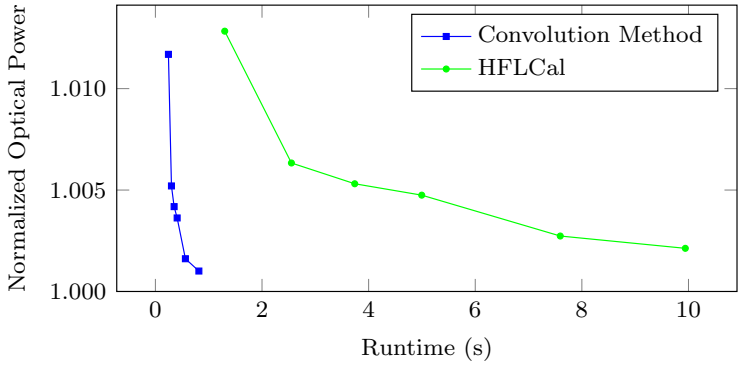

FIGURE 6. Accuracy of the optical power vs. runtime of our convolution method and the HFLCal method, using the PS10 test case with 624 heliostats.

Beside comparing the summed optical power on the receiver surface, we now investigate the accuracy for each single receiver cell of the flux map. In Figs. 7, 8 and 9, we see the flux maps for the Monte Carlo, HFLCal and convolution method with about 140 000 rays for the PS10 test case with cavity receiver, using 40 by 40 receiver pieces and a DNI of 820 W/m^2. All methods show a similar flux map.

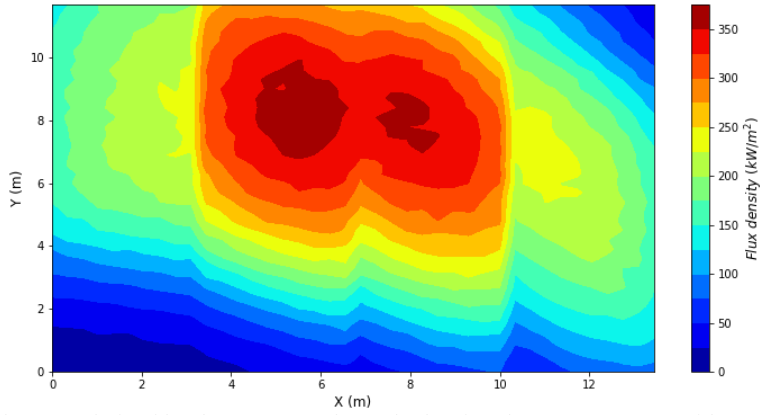

FIGURE 7. Flux map derived by the Monte Carlo method, using the PS10 test case with 624 heliostats.

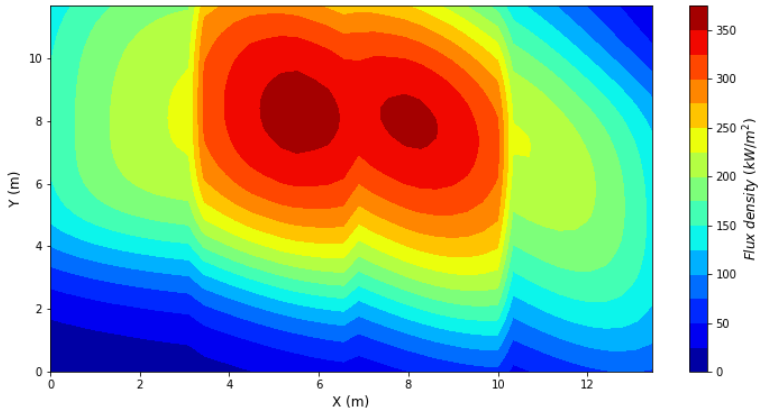

FIGURE 8. Flux map derived by our convolution method, using the PS10 test case with 624 heliostats.

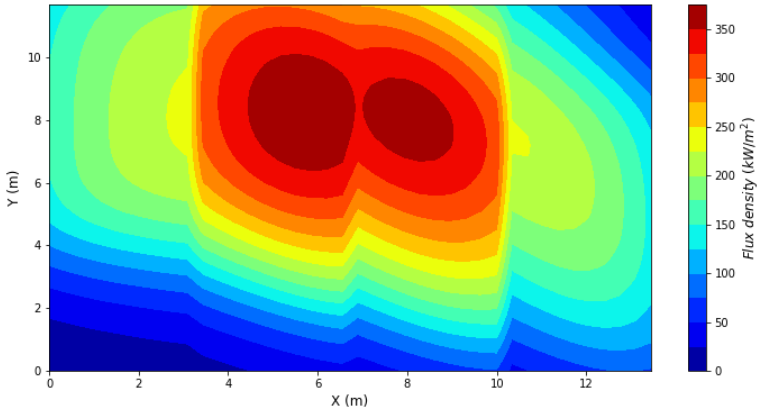

FIGURE 9. Flux map derived by the HFLCal method, using the PS10 test case with 624 heliostats.

To measure its accuracy, we compute the root mean square error (RMSE) to the reference solution, using once again the bidirectional Monte Carlo method with 140 000 rays. The receiver resolution then got quadratically increased, thus splitting one receiver piece into four new ones each iteration. Afterwards the fluxmaps are summed back to the 40 by 40 resolution. With an increase of the receiver resolution the run-time increases while the RMSE decreases. Fig. 10 illustrates the relative difference between the flux maps. As expected, the fluxmaps of our convolution method have the exact same RMSE. This is caused by the fact that the perspective projection of a splitted receiver piece take up the same area as the projection of the original piece. Moreover, the integration of each piece is almost exact. Thus, the optical power of the four splitted pieces sum up to the same power as the original one. Since the accuracy of HFLCal largely depends on the receiver resolution, it takes about four times as long to reach a similar RMSE. Fig. 10 also shows that the 40 by 40 fluxmap of our convolution ray-tracer is three times as accurate than the one generated with HFLCal.

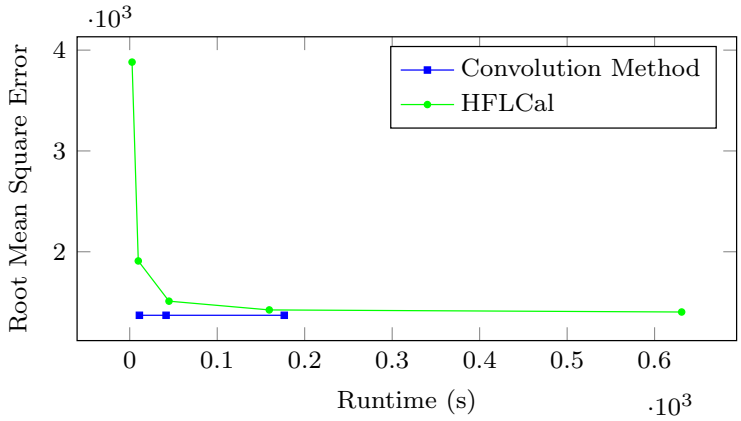

FIGURE 10. Accuracy of the flux map vs. runtime of our convolution method and the HFLCal method, using the PS10 test case with 624 heliostats.

CONCLUSION

The above shown investigation indicates the main advantage of the new convolution ray tracing method. It is faster than existing analytical ray-tracers such as HFLCal and can also handle multiple rays per facet making it applicable for achieving highly accurate results. Since the method is deterministic, the results for a given setup are identical and thus do not fluctuate. Within a case study our method is applied for the central receiver system PS10 in Spain. Besides the precision, our convolution method has also been shown to reach a high accuracy in a comparatively small time. For this reason, it is especially useful in an optimization process.

REFERENCES

1. Blanco, M. J (2006). Tonatiuh: An object oriented, distributed computing, Monte-Carlo ray tracer for the design and simulation of solar concentrating systems. *Technical report, The University of Texas at Brownsville*.
2. Wendelin, T. (2003). Soltrace: a new optical modeling tool for concentrating solar optics. In ASME 2003 International Solar Energy Conference, pages 253–260. *American Society of Mechanical Engineers*.
3. Ahlbrink, N., Belhomme, B., Flesch, R., Maldonado Quinto, D., Rong, A., & Schwarzbözl, P. (2012). Stral: Fast ray tracing software with tool coupling capabilities for high-precision simulations of solar thermal power plants. *In Proceedings of the SolarPACES 2012 conference*.
4. Schwarzbözl, P., Pitz-Paal, R. & Schmitz M. (2009). Visual HFLCAL - a software tool for layout and optimisation of heliostat fields. *In SolarPACES Conference*.
5. Sánchez-González, A. & Santana, D. (2015). Solar flux distribution on central receivers: A projection method from analytic function, *Renewable Energy*, 74, 576–587.
6. Di Donato, Jr. A. R., Jarnagin, M. P. & Hageman R. K. (1978). Computation of the bi-variate normal distribution over convex polygons. *Technical report, Naval Surface Weapons Center*.
7. Hövelmann, F. (2019). Accelerated raytracer for solar tower power plants, *Bachelor thesis, RWTH Aachen University*.
8. Rabl. A. (1985). Active solar collectors and their applications. *Oxford University Press.*
9. Schmitz M., Schwarzbözl P., Buck R., & Pitz-Paal R. (2006). Assessment of the potential improvement due to multiple apertures in central receiver systems with secondary concentrators. *Solar energy, 80(1):111–120.*
10. Di Donato A. R. & Hageman R. K. (1980). Computation of the integral of the bivariate normal distribution over arbitrary polygons. *Technical report, Naval Surface Weapons Center.*