SunFlower: A new solar tower simulation method for use in field layout optimization

Cite as: AIP Conference Proceedings **2033**, 210015 (2018); https://doi.org/10.1063/1.5067217
Published Online: 08 November 2018

Pascal Richter, Gregor Heiming, Nils Lukas, and Martin Frank

SunFlower: A New Solar Tower Simulation Method for Use in Field Layout Optimization

Pascal Richter[1, a)], Gregor Heiming[1], Nils Lukas[1] and Martin Frank[1]

[1]*RWTH Aachen University, Department of Mathematics, MathCCES, Schinkelstr. 2, 52062 Aachen, Germany*

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

Abstract. SunFlower is a new model for the simulation of Solar Tower Power Systems. The model has been cross-validated with the NREL tool SolTrace and is currently compared to other tools. This tool is freely accessible via a user-friendly web interface. The idea behind this Web App is to provide a high accurate tool with strong support for comparable programs, such that the exchange of simulation results is made easier.

INTRODUCTION

The design of the heliostat field layout is a challenging task of exceptional importance. It is the sub-system with the highest cost and its optimal design highly depends on the specifications for each project. The efficiency of the power plant is affected by the blocking and shading of mirrors depending on the heliostat placement. There are many different methods for heliostat field layout optimization, each offering their own set of advantages and drawbacks.

The different optimizers are based on models, which for each placement approximate the efficiency, received irradiation, thermal energy, levelized costs of electricity (LCOE), or any other objective function. Depending on the model accuracy the determined optimal layout differs from the field layout for optimal plant performance in reality. Thus, to compare optimized layouts from two different institutions which base on models with different accuracy, one needs a high-accurate reference model which can quantify the goodness of each layout. To achieve an acceptance of such a tool, it should be free, offer interfaces to other models, it should be fast in runtime, and the computation of the annual performance should be supported.

To overcome the lack of such a missing reference model we present a free model called SunFlower for the simulation of Solar Tower Power Systems. SunFlower is designed to have adjustable accuracy. Mathematical methods are used to strongly accelerate the simulation time. The tool is freely available online at http://www.solar.rwth-aachen.de/webapp.html and provides interfaces to other existing models (see last section of this paper).

SUNFLOWER SIMULATION TOOL

Starting in the 1970s, several different codes have been developed which simulate the irradiation power of a central receiver system. They mainly differ in the flux calculation method, where ray tracing and mathematical simulation techniques such as Hermite polynomial expansion or convolution are used [4]. Monte Carlo ray-tracers generate millions of randomized rays, where the directions of the rays are perturbed with a certain probability. This makes the obtained results very accurate if enough rays are used. On the other hand, the calculations are computationally expensive and therefore slow. Analytical codes consider the perturbation analytically; such that deterministic results are obtained. To avoid the increase of computational costs, simplifications of the models are made. Depending on these simplifications, the obtained results may not be as accurate as the results obtained by Monte Carlo ray-tracers.

SolarPACES 2017
AIP Conf. Proc. 2033, 210015-1–210015-9; https://doi.org/10.1063/1.5067217
Published by AIP Publishing. 978-0-7354-1757-1/$30.00

Ray-Tracing Method

The tool SunFlower is filling the gap between stochastic Monte Carlo ray-tracers and deterministic analytical models with respect to accuracy and run time. SunFlower offers the following ray-tracing methods:

- Monte Carlo
- Quasi-Monte Carlo using Sobol sequence
- Convolution method
- Area convolution method

The main difference between Monte Carlo ray-tracers and the convolution methods is how to deal with the optical errors. Whereas Monte Carlo tools consider the uncertainty by generating a massive number of rays according to the standard deviation of the optical errors, the convolution method computes theses errors analytically to save computation time. Thus, the ray is represented as an error cone, which causes a flux at the receiver's surface around its ideal hit point, see Fig. 1. Thus, the convolution method uses a hierarchical approach of ray-tracing methods [1, 5], where the complete flux is computed by numerical integration with the use of Gauss-Legendre quadrature rule. Therefore, the surface is partitioned in a number of regions, each with a representative ray. Each ray is weighted by the irradiance of its representative area. The area convolution method is the aggregation of several error cones, which can be computed analytical exactly for a locally flat plate.

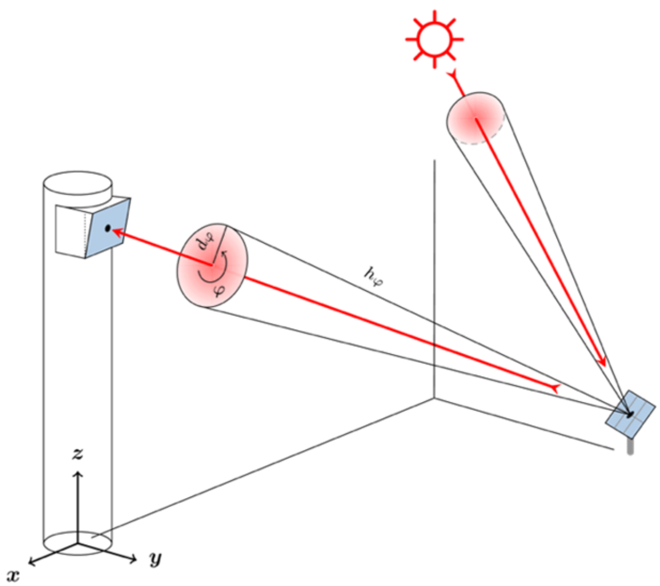

FIGURE 1. Sun rays are modeled as error cones with a distribution given by the standard deviation considering sun shape, tracking and slope errors.

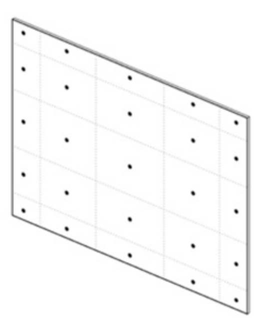

FIGURE 2. Numerical integration of the mirrors' flux with Gaussian quadrature rule. Each region has a representative ray, which is weighted by its area.

Shading and Blocking

The influence on the reflection by shading, blocking and ray interception at the receiver is determined just for this single ray as representative for the partitioned region. The number of representative rays per mirror is given by the selected order of the Gaussian quadrature rule. Because the effect of shading and blocking is typically small and confined to the edges [1], it is advantageous to have a fine discretization near the heliostat edge and a coarse discretization in the middle of the surface. With the choice of Gauss-Legendre quadrature rule for placing the representative rays, this is automatically satisfied, see Fig. 2 and 7 (a).

A heliostat is shaded or blocked by neighboring heliostats, the tower or the terrain. The computation of the shading and blocking effects with the neighboring heliostats is the most expensive part of a simulation. With the herein used hierarchical approach of a ray-tracing method, for each ray shading and blocking effects must be detected. Instead of using the expensive brute-force approach of a pairwise comparison of each ray with all heliostats we only consider a subset that can potentially shade or block a heliostat [5].

For this purpose, a two-dimensional bitboard index structure is used. The idea is to cover the two-dimensional x-y space with an equidistant grid such that the space is sub-divided in distinct quadratic cells. Inside those cells the information is stored if there is a nearby heliostat, see Fig. 3. For nearest-neighbor search, only the surrounding cells have to be checked, instead of all heliostats. With this approach, the blocking and shading method can strongly be accelerated, see Fig. 4. It can be seen, that the heliostat diagonal size H_{size} is a good choice for the bitboard cell size.

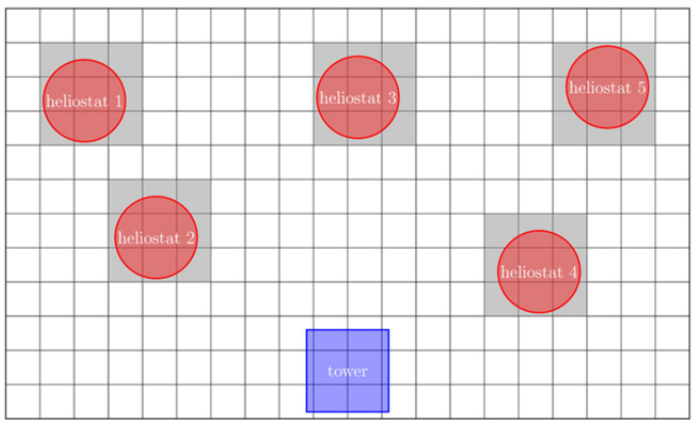

FIGURE 3. Schematic view of the bitboard structure.

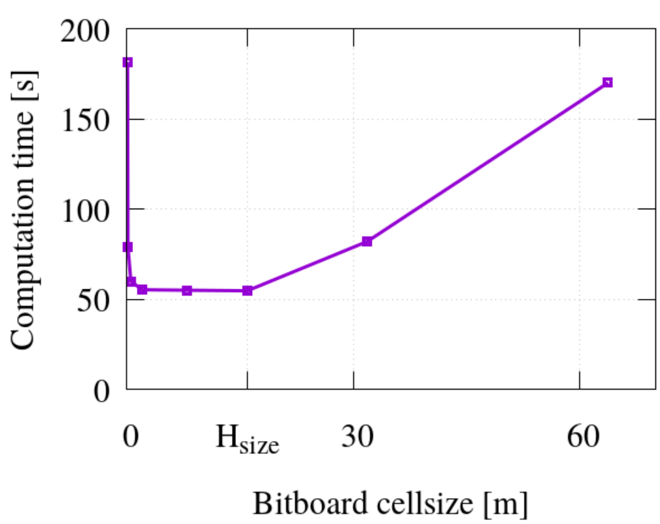

FIGURE 4. Comparison of the runtime with different Bitboard cell sizes.

Accuracy vs. Runtime

The accuracy of the SunFlower software is primarily adjusted by the number of simulated moments (for the annual performance, see next Subsection 'Time Integration'), and the number of rays per mirror. By using a higher number of rays also a higher accuracy can be reached. Figure 5 shows the accuracy of the simulation result vs. the runtime (by using different numbers of rays) on the same machine. Accuracy is here defined as (1 – relative error), where the reference value corresponds to the SolTrace result with 10 million rays. As shown below, SunFlower is already accurate for a relatively small runtime, see Fig. 5.

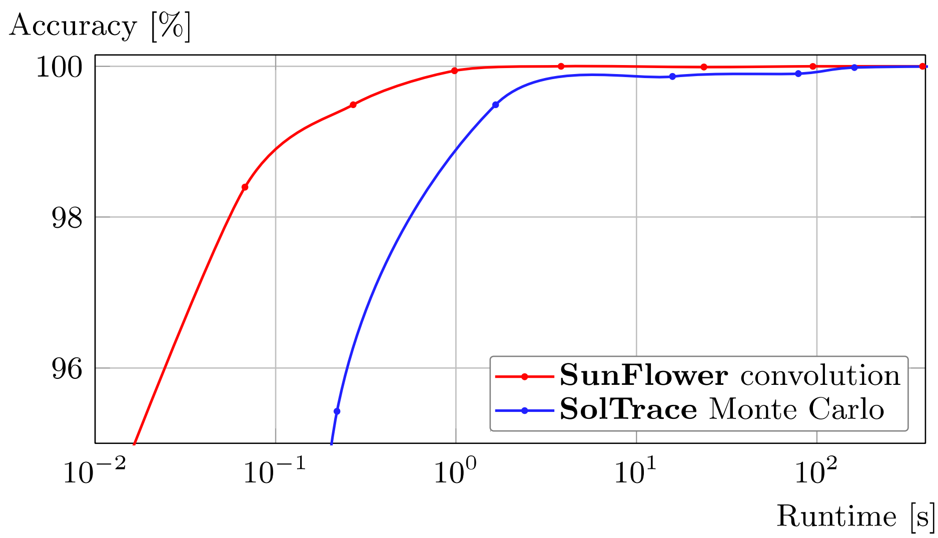

FIGURE 5. Comparison of accuracy (in %) vs. runtime of SolTraces's Monte Carlo and SunFlower's convolution method. For a desired accuracy of 99.9 % the convolution method is about 100 times faster than the Monte Carlo method.

Time Integration

The annual received optical radiation of the whole power plant is defined by

$$E_{\text{year}} = \int_0^{8760} P(t)\,\mathrm{d}t = A \cdot \int_0^{8760} I_{\text{DNI}}(t) \cdot \eta_{\cos}(\mathrm{t}) \cdot \eta_{\text{sb}}(\mathrm{t}) \cdot \eta_{\text{ref}}(\mathrm{t}) \cdot \eta_{\text{aa}}(\mathrm{t}) \cdot \eta_{\text{spl}}(\mathrm{t})\,\mathrm{d}t\,, \tag{1}$$

with mirror area A, direct normal irradiation I_{DNI}, cosine efficiency $\eta_{\cos}$, shading & blocking effects η_{sb}, mirror reflectivity η_{ref}, atmospheric attenuation η_{aa}, and spillage η_{spl}. The time integral over t [in hours] is solved numerically. In common practice, an iteration with constant time step [6, 9] is used, which corresponds to the midpoint rule. Noone et al. [1] propose an iteration with constant solar angle step, which allows the same accuracy with fewer iterations. There exist other approaches which just regard the sun angle instead of the time integral [13]. In our model, different quadrature methods can be used with either time or solar angle as integration variable, where the Gauss-Legendre quadrature gives the most promising results. An investigation regarding the number of simulated points in a year was performed. The results for a place near Mumbai (India) are shown in Fig. 6. Thus, for different numbers of considered days in a year (x-axis) and considered moments per day (y-axis) the relative error (in %) is shown as color plot. E.g. for 24 days each with 5 moments of a day, the error for the annual performance is lower than 1.3 %.

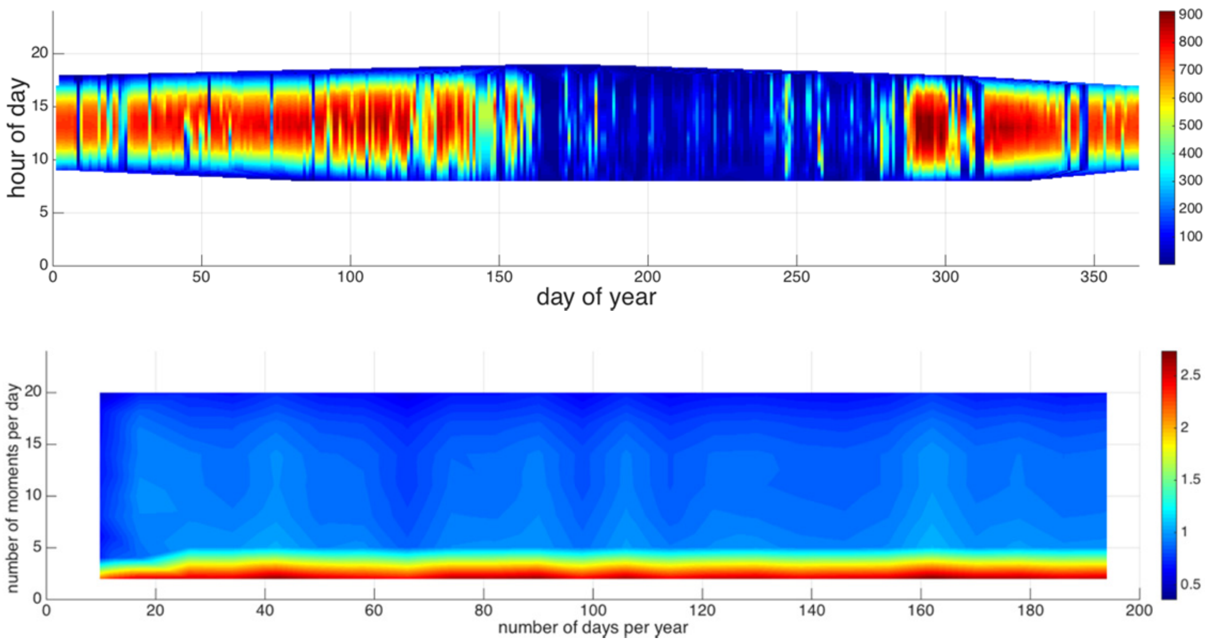

FIGURE 6. (Top) Measurements of the DNI of a place near Mumbai at a latitude of 18.97° and a longitude of 72.83°. The monsoon period around July is easy to detect. (Bottom) Relative error (in %) of the computed annual performance for different resolutions: Number of used days (trapezoidal rule) and number of used moments per day (Gaussian quadrature rule). The reference value corresponds to the annual performance of 365 days, considering each hour per day.

Object Shapes, Topography, and Restrictions

SunFlower is able to consider different shapes for the heliostats. Thus, a standard rectangular heliostat (e.g. Sanlúcar 120) or a pentagonal heliostat (e.g. Stellio 100) can be chosen, see Fig. 10. A heliostat can have multiple facets, which are either focused or flat and the facets can be canted on the heliostat scaffold. Additionally, SunFlower supports heliostats which can be clustered into so-called pod systems, as used already in the Helio100 test facility (100 kW) in Stellenbosch (South Africa). The heliostat's elevation depends on the topography of the specified terrain, which can automatically be loaded from Google Maps (via the web interface of our tool).

Different tower shapes (rectangular or cylindrical) are considered for the shadow and different receivers (flat rectangular, cavity, cylindrical external) are taken into account. The power plant site can be specified as an arbitrary polygon. Furthermore, restricted parts of the field, e.g. service roads or passing pipelines (which is the case for the solar tower in Jülich in Germany), can be specified. This is important when coupling the simulation model with an optimization.

Validation and Verification

SunFlower has been validated against the NREL software SolTrace [7] and is currently compared to the open-source tools Tonatiuh [8] and SolarPILOT [12]. For all three tools, SunFlower provides an interface, such that one can compare the results of the different simulation approaches. It is planned that this interface is extended for several other simulation tools, like STRAL [11], and the MIT code [1]. In Fig. 5, it was already shown, that SunFlower requires less rays to obtain the same accuracy as SolTrace.

Figure 7 shows the projection of one heliostat of 1.83m × 1.22m onto a 4m^2 receiver. SunFlower here uses 50 × 50 rays, while SolTrace simulated the same configuration using a million rays. The previously described Gauss-Legendre ray distribution can be identified in Fig. 7(a). This distribution does not change with optical errors as it does in SolTrace. Optical errors are considered by assuming error cones instead of simple rays. The obtained flux from the same test case is presented in Fig. 8 for the case, where no optical errors are considered. Figure 9 shows the flux maps for the test case with considering optical errors due to tracking, slope, and sun shape. It can be seen, that the solution is smearing out at the border.

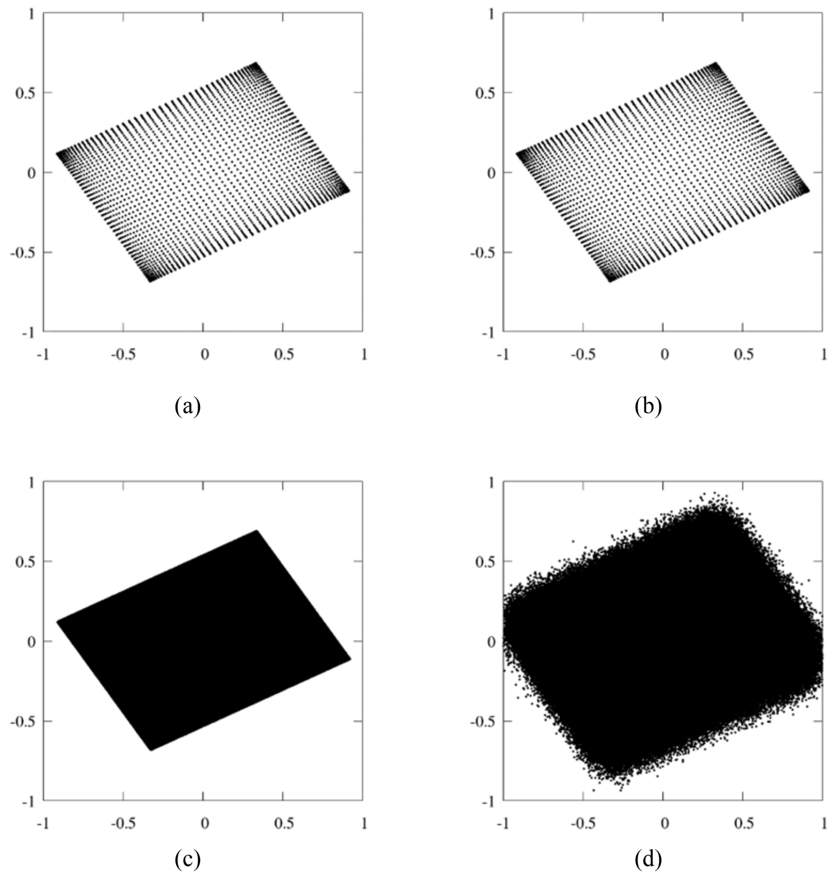

FIGURE 7. Ray distribution on the receiver in a test case with one single heliostat obtained by SunFlower without optical errors (a) and with optical errors (b), and SolTrace without optical errors (c) and with optical errors (d).

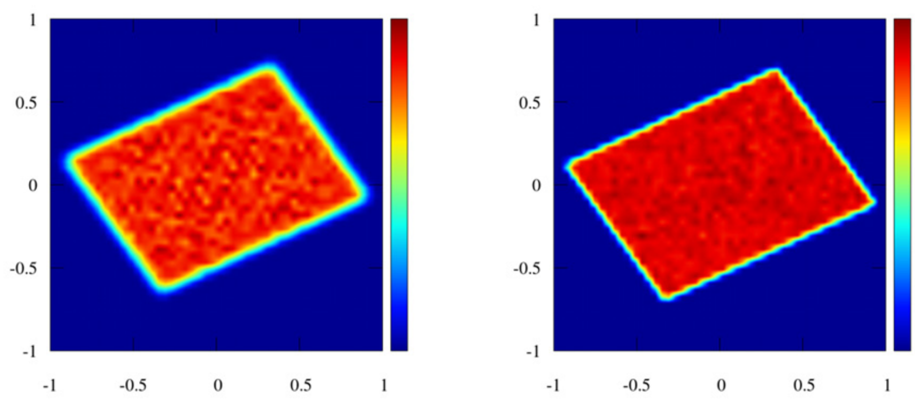

FIGURE 8. Flux maps without considering optical errors in SunFlower (left) and SolTrace (right).

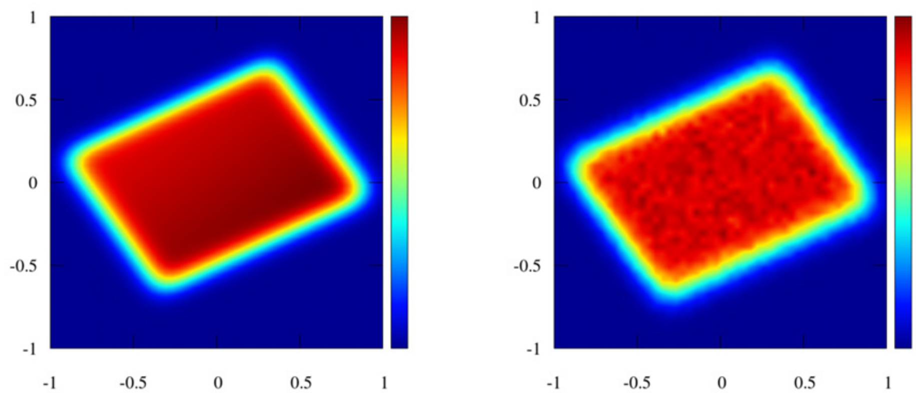

FIGURE 9. Flux maps with considering optical errors in SunFlower (left) and SolTrace (right).

SUNFLOWER WEB APPLICATION

A mobile-friendly web-application was created based on Google Polymer that provides access to the simulation model. Empirical measurements like topological or sun intensity data for any location can be verified and synchronized within the cloud. Subsequently, this data can be utilized by our services when performing simulations or optimizations. Furthermore, an in-browser 3D visualization of the user-specified power plant was designed based on WebGL to ease various parameter configurations for the heliostats and their field layout, see Figs. 10 and 11. The visualization reacts to different topography, meteorological models and measured irradiation data for scalable spatial and temporal resolution. The power can be computed either annually or for single moments using the meteorological radiation clear-sky model (MRM) [10] or by including measurements from a file such as EnergyPlus (EPW) or Meteonorm (TMY3). Once registered, the power plants and a history of the simulations are visible only to the user. However, it is also possible to share the model settings with the whole user-base. Detailed simulation outputs are provided, e.g. single heliostat contributions, average power, single efficiency values, field analysis, and receiver flux map. In the future, the SunFlower method will be combined with an optimization tool. The application is available online: http://www.solar.rwth-aachen.de/webapp.html.

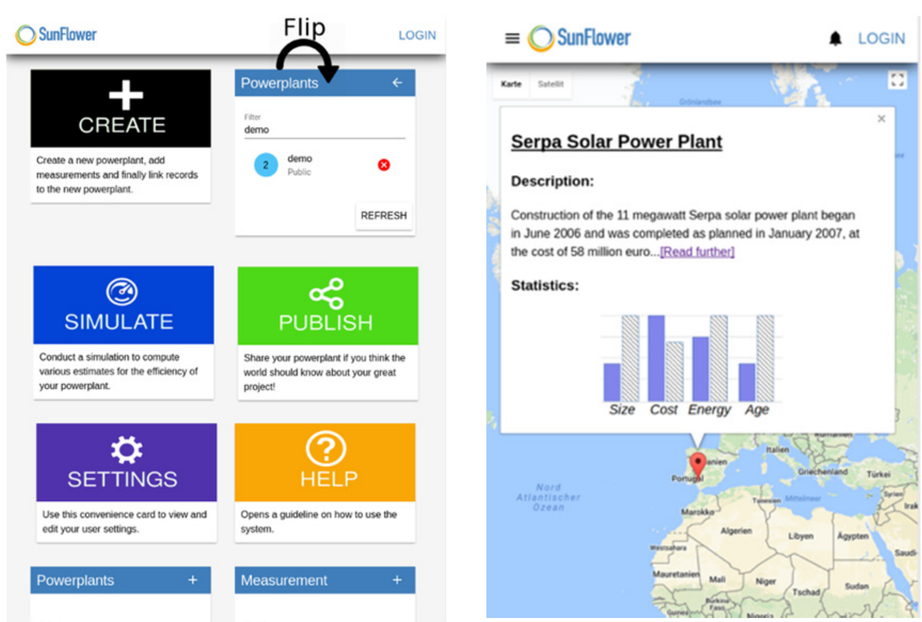

FIGURE 10. Screenshots of SunFlower's web interface.

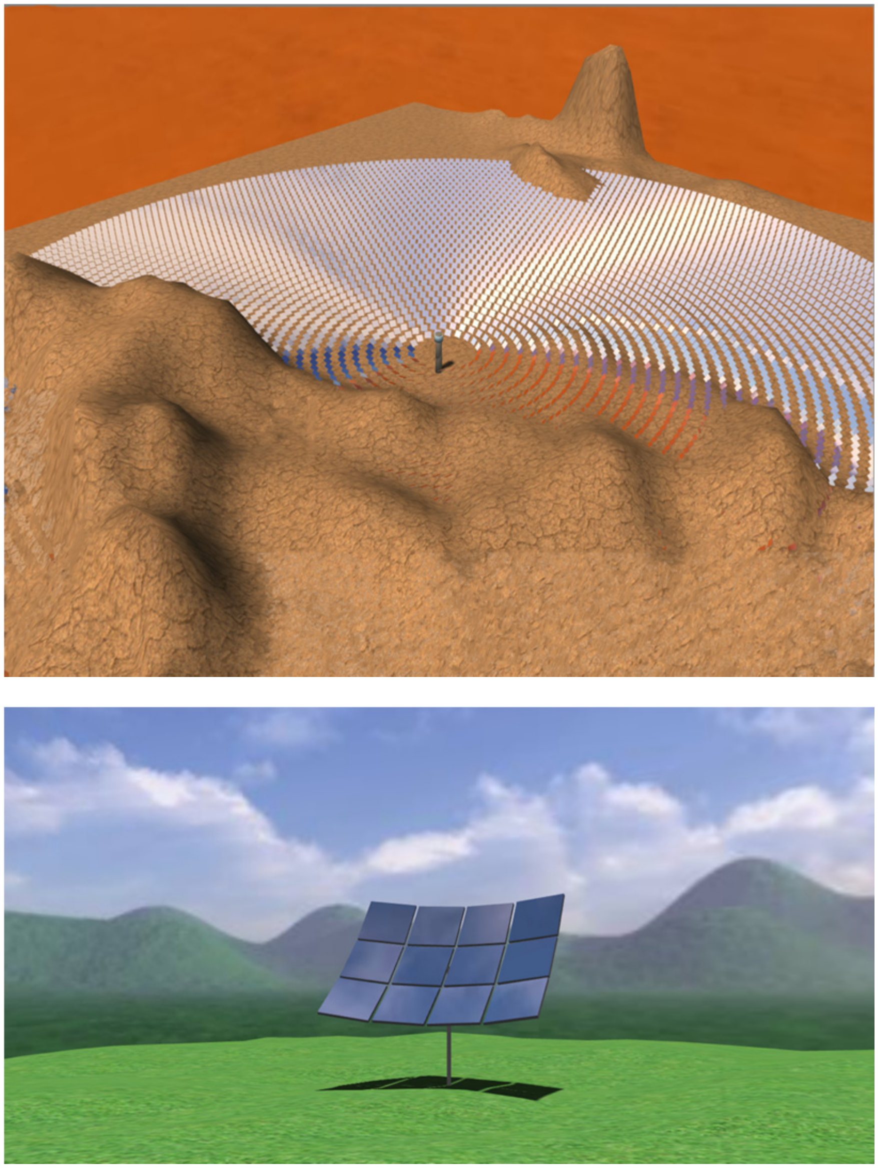

FIGURE 11. Screenshots of SunFlower's WebGL visualization.

OUTLOOK

In July 2018, a beta version of SunFlower is freely available as a web application at http://www.solar.rwth-aachen.de/webapp.html. In the near future, SunFlower will offer the possibility to compute the net annual energy production. For this, a thermal model of the receiver and a power block model are implemented. This parameter then can be used to compute different economic indicator functions, e.g. the levelized cost of electricity (LCOE), the net present value (NPV), or the internal rate of return (IRR). It is planned, that SunFlower is extended for the heliostat field layout optimization. Furthermore, we will include optimization algorithms for aiming strategies and optimal cable routings for the connection of heliostats.

ACKNOWLEDGMENTS

The authors acknowledge the financial support by the Federal Ministry for Economic Affairs and Energy of Germany in the project SolFieOpt (project number 0324039).

Supported by:

Federal Ministry for Economic Affairs and Energy

on the basis of a decision by the German Bundestag

REFERENCES

1. C. J. Noone, M. Torrilhon, and A. Mitsos, "Heliostat field optimization: A new computationally efficient model and biomimetic layout" in *Solar Energy*, 86(2), 792-803 (2012).
2. R. Buck, "Heliostat field layout using nonrestricted optimization". In SolarPACES Conference, 2012.
3. Lutchman, S., Paul Gauché, and A. Groenwold, "On selecting a method for heliostat field layout optimization", Stellenbosch University, 2013.
4. M. Geyer and W.B. Stine. "Power from the sun", Website. Available online at www.powerfromthesun.net/book.html, visited on July 1st, 2017.
5. B. Belhomme, R. Pitz-Paal, P. Schwarzbözl, and S. Ulmer, "A new fast ray tracing tool for high-precision simulation of heliostat fields" in *Journal of Solar Energy Engineering*, 131(3):031002 (2009).
6. M.M. Elsayed, H. Allah, and O.M. Al-Rabghi, "Yearly-averaged daily usefulness efficiency of heliostat surfaces" in *Solar energy*, 49(2):111-121 (1992).
7. T. Wendelin, "SolTRACE: a new optical modeling tool for concentrating solar optics" in *ASME 2003 International Solar Energy Conference*. American Society of Mechanical Engineers, 2003.
8. 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, 2005.
9. B.L. Kistler, "A user's manual for DELSOL3: A computer code for calculating the optical performance and optimal system design for solar thermal central receiver plants", Technical report, Sandia National Labs., Livermore, CA (USA), 1986.
10. V. Badescu, "Modeling solar radiation at the earth's surface", Springer, 2014.
11. N. Ahlbrink, B. Belhomme, R. Flesch, D. Maldonado Quinto, A. Rong, P. Schwarzbözl, "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*, 2012.
12. SolarPILOT, Website. Available online at http://www.nrel.gov/csp/solarpilot.html, visited on July 1st, 2017.
13. V. Grigoriev, C. Corsi, M. Blanco, "Fourier sampling of sun path for applications in solar energy" in *Proceedings of the SolarPACES 2016 conference*, 2016.