

Diese Arbeit wurde vorgelegt am
Lehr- und Forschungsgebiet Theorie der hybriden Systeme

**Modelling and Optimization of large scale Solar Tower
Power Plants**
**Modellierung und Optimierung von
Solarturm-Kraftwerken**

Masterarbeit
Informatik

Dezember 2018

Vorgelegt von Presented by	Linus Franke Viktoriaallee 2 52066 Aachen Matrikelnummer: 333202 linus.franke@rwth-aachen.de
Erstprüferin First examiner	Univ.-Prof. Dr.rer.nat. Erika Ábrahám Lehr- und Forschungsgebiet: Theorie der hybriden Systeme RWTH Aachen University
Zweitprüfer Second examiner	Prof. Dr. Martin Frank Steinbuch Centre for Computing Karlsruhe Institute of Technology
Externer Betreuer External supervisor	Dr. rer. nat. Pascal Richter Steinbuch Centre for Computing Karlsruhe Institute of Technology

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich diese Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Stellen meiner Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Dasselbe gilt sinngemäß für Tabellen und Abbildungen. Diese Arbeit hat in dieser oder einer ähnlichen Form noch nicht im Rahmen einer anderen Prüfung vorgelegen.

Aachen, im Dezember 2018

Linus Franke

Contents

I. Models	6
1. Optical model	6
1.1. Site	6
1.1.1. Coordinate system	6
1.1.2. Definition of area	6
1.1.3. Restricted area	7
1.1.4. Elevation	7
1.2. Meteorological Information	7
1.2.1. Sun position	7
1.3. Heliostats	8
1.3.1. Heliostat geometry	8
1.3.2. Minimal distance between heliostats	9
1.3.3. Local heliostat coordinate system	9
1.3.4. Alignment of the heliostats	9
1.3.5. Canting	10
1.3.6. Focal points of facets	12
1.4. Tower	12
1.5. Receiver	13
1.5.1. Flat tilted cavity receiver	13
1.5.2. Cylindric cavity receiver	13
1.5.3. Cylindric external receiver	14
1.6. Ray-tracer	15
1.6.1. Efficiencies and losses	15
1.6.2. Flux map	19
1.6.3. Monte Carlo method	19
1.6.4. Convolution method	20
1.6.5. Cell-wise convolution method	21
2. Thermal model	25
2.1. Receiver	26
2.2. Simplified receiver model	26
2.2.1. Incident radiation	26
2.2.2. Reflection	27
2.2.3. Radiation	27
2.2.4. Convection	28
2.3. Thermal storage	29
3. Electrical model	29

4. Annual integration	31
4.1. Quadrature	31
4.2. Real weather data	32
4.3. MRM	32
4.4. JSON file	33
5. Economic model	35
5.1. Investment cost	35
5.2. Operation and maintenance costs	39
5.3. Economic evaluation	40
6. Settings	41
6.1. Site settings	41
6.2. Meteorological information	41
6.3. Heliostat settings	41
6.4. Tower settings	42
6.5. Receiver settings	42
6.5.1. Ray-tracer settings	42
6.6. Thermal model settings	42
6.7. Electrical model settings	42
6.8. Economic model settings	42
II. Feasibility tests	47
7. Validation of optical model	47
7.1. Cross validation against SolTrace	47
7.1.1. One heliostat	47
7.1.2. Blocking and shading	48
7.1.3. Complete solar tower power plant	49
7.2. Accuracy vs number of rays	50
7.3. Validation of ray tracing methods	50
7.4. Conclusion	51
8. Bitboard resolution and preselection	55
8.1. Only preselection	56
9. Parallelization in hierarchical ray tracing	56
III. Optimization	58
10. Algorithms	58
10.1. Patterns	58
10.1.1. North-South Cornfield	59

10.1.2. Radial staggered	59
10.1.3. Hexagon	61
10.1.4. Spiral	63
10.1.5. Contracted honeycombs	64
10.1.6. JSON file	65
10.2. Genetic algorithm	65
10.2.1. Discrete Value Mapping	66
10.2.2. Chromosome Factory	67
10.2.3. Stopping Criteria	68
10.2.4. Selection Operation	68
10.2.5. Coupling Operation	68
10.2.6. Crossover Operation	68
10.2.7. Choosing Operation	68
10.2.8. Mutation Operation	69
10.2.9. Replacement and Replenish Operation	71
10.2.10. JSON file	71
10.3. Variable neighborhood descent	72
10.3.1. Neighborhoods	72
10.3.2. JSON file	72
11. Multi-Step optimizer	73
11.1. Settings	73
11.1.1. JSON file	73
12. Optimizing large-scale solar tower power plants	74
12.1. Used algorithms	74
12.2. Test case PS10	74
12.2.1. North-South Cornfield	75
12.2.2. Radial Staggered	75
12.2.3. Hexagon	75
12.2.4. Spiral	75
12.2.5. Contracted Honeycombs	75
12.2.6. Genetic algorithm	75
12.2.7. Variable neighborhood descent	75
12.3. Economical evaluation	76
IV. Summary	79
13. Conclusion	79
14. Outlook	80
References	82

Introduction

Renewable energy production has grown over the last decades. Therefore large-scale renewable power plants are build, see Table 1. This growth is due to technological advancements and research in renewable energies. Since 2010 the levelized cost of energy (LCOE) has fallen for every renewable energy production method, allowing most to compete with fossil fuel energy production. This is achieved by reaching the cost range of fossil fuel energy production [1]. Furthermore, few renewable energy production methods have a LCOE even below the fossil fuel cost range or are predicted to continuously be below most fossil fuel costs.

Name	Location	Type	Capacity[MW]
Gansu Wind Farm	China	onshore wind	6800
Walney Extension	UK	offshore wind	659
Three Gorges Dam	China	hydroelectric	22500
Solar Star	US	photovoltaic	579
Ivanpah Solar Power Facility	US	concentrating solar power	392

Table 1: Small overview over a few of the biggest renewable energy plants.

For regions with high direct solar irradiation, also direct normal irradiation (DNI), concentrating solar thermal power (CSP) plants are a promising dispatchable renewable energy production. CSP plants with a thermal storage haven astonishing easy principle and use technology also used by fossil fuel energy production, the steam turbine. A large number of mirrors of different sizes and shapes reflect sunlight on an absorber where a fluid is heated up by the concentrated irradiation. Those absorbers are typically tower mounted receivers using air, water/steam, thermal oil or a molten salt as heat transferring fluid (HTF). The high temperatures of the HTF allow for a high cost-efficiency [47]. In a thermal energy exchange water is turned into steam powering a steam turbine.

The thermal energy can also be stored in huge thermal energy storage tanks which then can provide the electricity on demand. Those storage capabilities of this technology are a huge benefit and necessary since renewable energy production hardly ever matches the current electricity demand. This allows countries with high renewable energy productions to even out fluctuations in their power grid. It even helps to further increase the usability of non-dispatchable renewable energy technologies.

Today multiple large-scale CSPs are connected to national power grids providing power when needed. In the US the Ivanpah Solar Power Facility and the Crescent Dunes Solar Energy Project have a electric capacity of 392 and 125 MW respectively. In South Africa the Khi Solar One has a capacity of 50 MW. Currently under construction are Noor III in Morocco, Ashalim power station A in Israel, Cerro Dominador Solar Thermal Plant in Chile and Redstone Solar Thermal Power in South Africa with expected electric capacities of 150, 121, 110 and 100 MW respectively. Furthermore, the Sandstone Solar Energy Project in the US is announced with 1.6 GW and in Chile two projects with 13 hours of thermal energy storage are announced with 390 and 450

MW. In addition to these large-scale commercial power plants are multiple smaller research facilities in use, like the Solarturm in Jülich, Germany.

In order to optimize the LCOE or other economical values the Annual Energy Production(AEP) is a factor that has to be maximized. A significant part of maximizing the AEP is the collection of sunlight. More collected sunlight corresponds to a higher AEP. Therefore the positioning, or layout, of the mirrors reflecting the sunlight is an essential task within this optimization.

The position of a mirror underlies multiple effects which lessen the reflected sunlight on the receiver. Whilst some of them are angle or distance dependent. Other effects incorporate the interplay of different mirrors. Shadows can be thrown when an object lies in between the mirror and the sun or the reflected sunlight is blocked when an object lies in between the mirror and the receiver.

The minimization of those effects can be achieved by moving the positions, e.g. changing the layout of the mirrors on the plant. Current research regards different layouts with own advantages and disadvantages [34].

In this thesis an optimization of a CSP is presented. The first part presents a model which is capable of calculating optical, thermal and electric energy production over a year as well as economical values such as LCOE, internal rate of return or net present value. The latter depending on the AEP and investment cost. Figure 0.1 shows the interaction of the models.

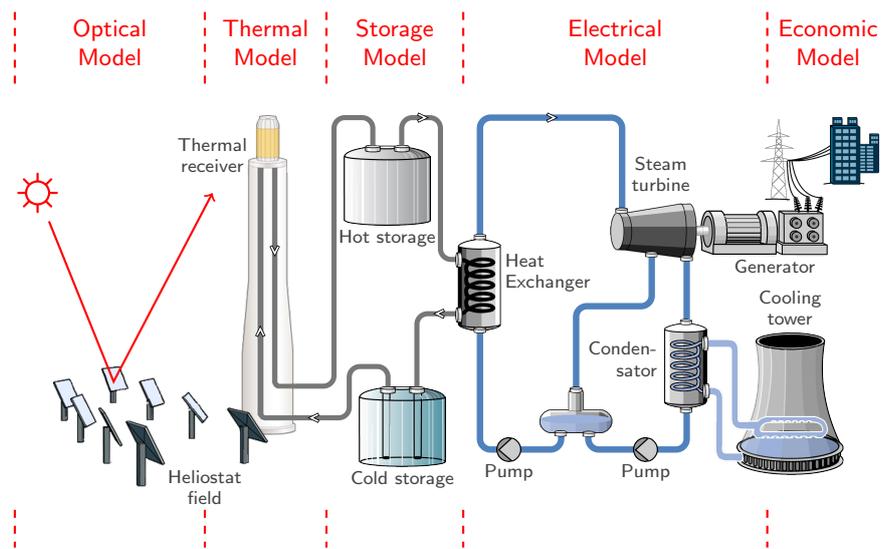


Figure 0.1: Model overview [47]

The optimization of the layout requires a fast and accurate model. These capabilities are presented in the second part. Here a cross-validation is carried out. Furthermore several methods for speeding up the optical model are presented and evaluated, whilst keeping the accuracy.

The third part introduces state of the art algorithms as well as new methods to

optimize the positioning of the mirrors. A combination of algorithms is chosen to show the effects of the optimization on an existing CSP.

A summary as well as an outlook on this thesis and research to be done is given in the fourth part.

In the following an overview over related work and state of the art in this field is given.

Related work

In the following, existing tools for simulation of solar tower plants and optimization of the heliostat layout problem are listed. The summary is inspired by the reviews of Bode and Gauché [8], Garcia et al. [18] and Richter [47].

Model

Starting in the 1970s, a bunch of different codes has been developed to calculate the collected irradiation power in a central receiver system. Ray tracing as well as mathematical simulation techniques were used to calculate the flux. The latter being Hermite polynomial expansion or convolution [20].

In order to model errors which occur in reality Monte-Carlo ray-tracers generate millions of randomized rays. Each ray gets perturbed with a certain probability. Whilst this makes the obtained results very accurate the calculation is computationally expensive.

The usage of Gaussian distribution in analytical simulation techniques provides deterministic results. In order to obtain these results in suitable time simplifications on the models are made. Due to these simplifications the results may not be as accurate as when using a Monte-Carlo ray tracing approach.

Ray-tracers

Starting development in 1978, *MIRVAL* [30] is one of the first Monte-Carlo ray-tracers. A commercial version named *SPRAY* is commercially available via the German Aerospace Center(DLR).

A freely and state of the art tool is *SolTrace* developed since 1999 by the US National Renewable Energy Laboratory (NREL). *SolTrace* utilizes parallelization in order to reduce simulation time. It is capable of directly showing the flux distribution and the obtained power as well as returning the simulated rays for post-processing steps.

The development of the open-source ray-tracer *Tonatiuh* [7] was started in 2004 by teams of the University of Texas in Brownsville and the Spanish National Renewable Energy Center (CENER). The returning post-processing script contains the rays, the resulting power and the flux distribution [8].

STRAL [6] is a backward ray-tracer developed by the DLR. Instead of generating the rays above the heliostat field, they are generated directly on the mirror surface. In comparison to the previously mentioned ray-tracers this practice does not lose rays and therefore has a shorter runtime. The tool comes with the capability to handle

highly resolver mirror surface geometries. It is commercially available or through collaborations with the DLR [8].

The Monte-Carlo ray-tracer *SOLFAST* (SOLar Facilities Simulation Tools) was developed by HPC-SA and PROMES-CNRS and uses an integral formulation instead of collision-based ray tracing. A cross-validation with *SolTrace* and *Tonatiuh* was made [52].

Mathematical simulation models

The software suite *UHC*, also called *RCELL suite*, was developed by the University of Houston in 1974 [18]. It was used to design the Solar One CSP [44]. Based on the *UHC* the commercially distributed software *TieSol* was developed, see above.

In 1978 Sandia developed the code *DELSOL* [27]. For flux computations it uses Hermite polynomial convolution. Additionally to other early developed codes it can optimize additional parameters such as tower height and receiver site, since it implements an economical model. The Windows software adaption *WINDELSOL* provides even more features [18].

HELIOS is another code developed at Sandia [17]. It uses detailed heliostat surfaces in order to calculate the flux distribution based on cone optics. This provides an accurate flux distribution. *HELIOS* is hard to use and not available anymore.

The company Interatom started the development of *HFLCAL* (Heliostat Field Layout CALCulation) [26] in the eighties. The DLR made it commercially available after further development in the nineties [58]. The tool uses a computationally efficient approach utilizing a simplified convolution of the heliostats' flux [18]. Due to permanent improvement since the eighties, *HFLCAL* got some good features like automatic multi-aiming and several different receiver models with secondary concentrators etc. [58].

ISOS is a code developed at the National Autonomous University of Mexico (Universidad Nacional Autónoma de México) [51]. In order to compute a 3D flux from a single heliostat the code requires the input of an external ray-tracer [8].

HFLD is commercially available [8] and developed at the Chinese Academy of Sciences [64]. It traces four rays from the edges of each heliostats, this is called edge-ray principle. Therefore it is sufficiently fast for optimization algorithms.

The CRS4 research center developed the model *CRS4-2* based on tessellation of the heliostats [31].

Optimization of the heliostat field layout

Four main concepts in the heliostat field layout optimization can be found: the field growth method, the pattern method, the free variable method and the hybrid or multi-step optimization strategy [34].

- The *field growth method* utilizes a discretization of the field in order to add the heliostats step by step on pre-defined points. The algorithm terminates when a defined stopping criteria, e.g. minimum AEP, is met. The runtime can not be parallelized due to the positioning of each heliostat depending on the previously

placed heliostats. Furthermore the efficiency and runtime are highly dependent on the number of pre-defined points on the field. A greedy heuristic is used by Sánchez and Romero [54] to employ this concept.

- In the *pattern method* all heliostats are arranged in geometric patterns. Each pattern can then be described by certain adjustable parameters. Instead of optimizing hundreds or thousands of x and y positions in this case only a few parameters have to be optimized. These then influence the resulting x and y positions. Therefore only the best suitable adaption of the base pattern is searched and not necessary the optimal x and y positions for the optimal plant performance [34]. State of the art research considers north-south cornfields [57], radial staggered [32], hexagons [3, 42] and spirals [39]. A disadvantage of this method is the reduced search space during construction.
- The *free variable method* directly optimizes the x - y coordinates. The complexity of this problem requires appropriate heuristics in order to solve it. The field of optimization many heuristics were developed over the last years. Such as non-linear programming, general gradient-based methods or nature-inspired heuristics like genetic, evolutionary, viral, simulated annealing and particle swarm heuristics. In [34] a gradient-based method is presented.
- The *multi-step optimization strategy* does not specify how exactly the x - y positions are optimized. It rather combines two or more optimization methods. This strategy aims at reducing the search space to a solution near the global maximum first and then refining this solution locally. For the last step either a greedy heuristic or a linear programming algorithm might be a good choice. Buck [9] showed when using a pattern based optimization first and the refining the solution with a greedy heuristic this provides better results then each algorithm alone.

Part I.

Models

1. Optical model

A solar field is given by N heliostats H_i $i \in \{1, \dots, N\}$, each with a mirror area A_i . For the time-dependent solar angles θ_{solar} as well as γ_{solar} and the direct normal irradiation I_{DNI} . The optical model utilizes a ray tracing approach to compute the cumulated optical radiation on a tower-mounted receiver for a given moment. During the tracing cosine effects η_{cos} , shading and blocking η_{sb} , heliostat reflectivity η_{ref} , atmospheric attenuation η_{aa} and spillage losses η_{spl} are taken into account.

The developed optical model is a renewed model from the model in Richter [47], Richter et al. [50]. While it has a comparable scope similar to the old model, it tackles some design issues. This section aims at a complete overview over this model. In Section 1.1 the site specific parameters are explained. Furthermore, Section 1.2 describes the meteorological information such as the modeling of the sun. Sections 1.3, 1.4 and 1.5 describe the modeling of the heliostats, tower and receiver respectively. In Section 1.6 we present a Monte-Carlo ray tracing as well as our newly invented ray tracing methods, which use the Gaussian distribution in order to analytically calculating a flux distribution.

1.1. Site

1.1.1. Coordinate system

From the user's view the coordinate system's origin corresponds to the provided latitude and longitude. The \mathbf{x} axis points towards East and the \mathbf{y} axis points towards North. The \mathbf{z} axis points vertically upwards. One unit at each axis represents one meter. In Figure 1.1 the coordinate system is shown in a 2D case from the bird perspective.

1.1.2. Definition of area

The site area is given by a list of boundary points, that form a polygon describing the site. The boundary points can be given as Cartesian coordinates in the coordinate system (see 1.1.1) as well as geo coordinates, e.g. latitude and longitude pairs. Each element placed in the solar tower power plant model is checked if they are inside the polygon. This check also includes the expansion and therefore preventing each object to lean over the boundary. Figure 1.1 shows an exemplary definition of an area with boundary points. The object in the mid of the figure is valid whilst the object outside of the polygon is not.

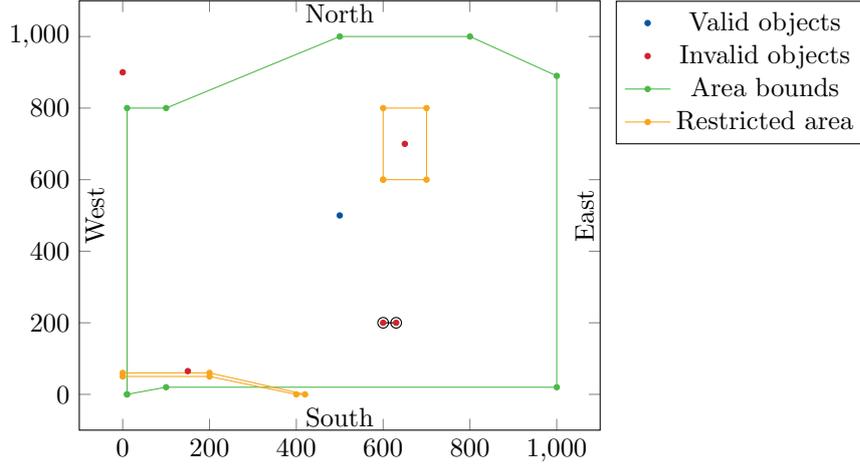


Figure 1.1: Exemplary area with an abstract outline and two restricted areas.

1.1.3. Restricted area

The restricted areas are defined in the same way as the boundary of the site, but they can contain multiple polygons. Again each placed object is checked against a violation by lying (partially) inside a defined polygon. Figure 1.1 depicts two restricted areas. Two objects violate the restricted areas. One lies in the rectangular area and the other lies partially within the bounds of the other area.

1.1.4. Elevation

The topographical information of the site can be given as (x, y, z) tuples. As this data is provided at discrete points, a bilinear interpolation is used to get the elevation of a single object between these discrete points. If a heliostat is outside of the bounding box around the provided discrete points, it inherits the elevation from the nearest neighbor in this set.

1.2. Meteorological Information

In our models we need some meteorological information which is first used by the optical model. In the optical model we need information about the sun which is given by its azimuth γ_{solar} , altitude θ_{solar} and irradiation I_{DNI} in $[\text{W m}^{-2}]$, see Figure 1.2.

1.2.1. Sun position

The direction of the sunlight is given by the solar vector τ_{solar} which can be calculated by Equation (1.1). A visualization of the solar vector τ_{solar} in our three-dimensional

Cartesian coordinate system can be found in Figure 1.2.

$$\tau_{\text{solar}} = \begin{pmatrix} \sin(-\gamma_{\text{solar}}) \cdot \cos(\theta_{\text{solar}}) \\ \cos(-\gamma_{\text{solar}}) \cdot \cos(\theta_{\text{solar}}) \\ \sin(\theta_{\text{solar}}) \end{pmatrix} \quad (1.1)$$

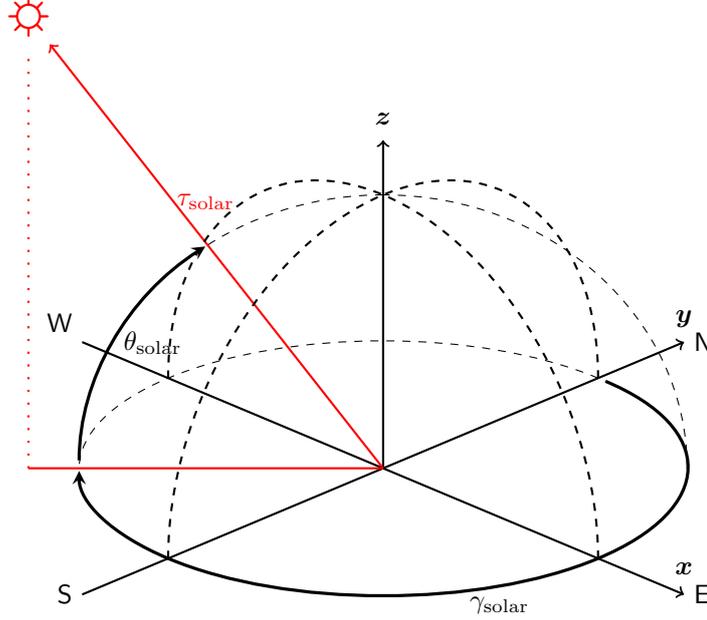


Figure 1.2: The solar position τ_{solar} is given by solar altitude θ_{solar} and solar azimuth γ_{solar} . The Figure is derived from Richter [47, p. 7]

1.3. Heliostats

1.3.1. Heliostat geometry

Each heliostat H_i is raised on a pedestal and has a mirror center-position p_i . All heliostats have an arbitrary layout of small mirrors, called facets, mounted on the mirror frame, see Figure 1.3. These facets have either a rectangular or a triangular shape with a right angle.

The layout of the facets on the heliostats can be described via two methods. The first method describes heliostats of a rectangular form which is given by the number of facets in horizontal $n_{\text{Facets,horizontal}}$ and vertical direction $n_{\text{Facets,vertical}}$, a uniform length ℓ_{Facet} and width w_{Facet} for each facet as well as horizontal w_{Gap} and vertical ℓ_{Gap} gaps between the facets. The total mirror area as well as the overall height and width of a heliostat can then be described with

$$\begin{aligned} A_{\text{helio}} &= (n_{\text{Facets,horizontal}} \cdot w_{\text{Facet}}) \cdot (n_{\text{Facets,vertical}} \cdot \ell_{\text{Facet}}) \\ h_{\text{helio}} &= n_{\text{Facets,vertical}} \cdot \ell_{\text{Facet}} + (n_{\text{Facets,vertical}} - 1) \cdot \ell_{\text{Gap}} \\ w_{\text{helio}} &= n_{\text{Facets,horizontal}} \cdot w_{\text{Facet}} + (n_{\text{Facets,horizontal}} - 1) \cdot w_{\text{Gap}} \end{aligned} \quad (1.2)$$

The heliostat's expansion d_{helio} is the diameter of the minimum bounding sphere, $d_{\text{helio}} = \sqrt{h_{\text{helio}}^2 + w_{\text{helio}}^2}$.

The second method of describing the layout of facets on a heliostat is by describing each single facet and giving a relative position to the center of the heliostat according to its local coordinate system (see Section 1.3.3). Then the heliostat area A_{helio} is the sum of all facet areas. The height and weight can be calculated by the distances of the farthest points in the corresponding directions.

1.3.2. Minimal distance between heliostats

For security reasons or to make sure every heliostat is accessible for cleaning and maintenance, it may be desired to have a minimal distance between two heliostats. The distance is measured between the two bounding spheres of the heliostats. In Figure 1.1 there are two heliostats which are too close to each other. The distance between these bounding spheres is less than the specified minimal distance.

1.3.3. Local heliostat coordinate system

The local coordinate system preserves the length of the global coordinate system, i.e. one length unit stays one meter. Its origin is the heliostat mirror-position p_i raised by its pedestal height. The orientation is according to the alignment of the heliostat at different sun positions.

The x -axis x_i of heliostat i is defined as the horizontal direction, i.e. parallel to the lower edge of the heliostat. y_i is parallel to the vertical edge of the heliostat. The z -axis is the normal vector n_i of the heliostat scaffold. Figure 1.3 shows the local coordinate system for a flat heliostat with four times seven facets.

When transforming a point q from local to global coordinates or vice versa, the following vector equation has to be solved for the global position vector or the local vector respectively:

$$\begin{pmatrix} | \\ q_{\text{global}} \\ | \end{pmatrix} = \begin{pmatrix} | & | & | \\ x_i & y_i & n_i \\ | & | & | \end{pmatrix} \cdot \begin{pmatrix} | \\ q_{\text{local}} \\ | \end{pmatrix} \quad (1.3)$$

When solving for q_{local} , an explicit formula for inverting 3×3 matrices can be used. The determination of x_i , y_i and n_i is described in the following Section.

1.3.4. Alignment of the heliostats

All heliostats are aligned such that the reflected ray with origin in p_i aims at the receiver aiming point $p_{\text{aim},i}$. The aiming point depends on the receiver geometry see Section 1.5 and Equations (1.8, 1.11, 1.13). The normalized reflective vector can be computed by

$$r_i = \frac{p_{\text{aim},i} - p_i}{|p_{\text{aim},i} - p_i|}. \quad (1.4)$$

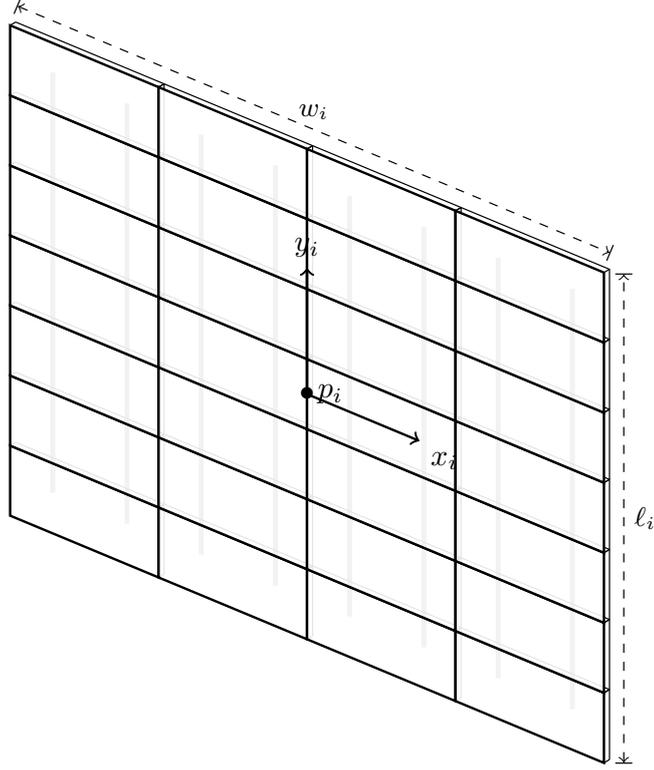


Figure 1.3: Horizontal and vertical heliostat axes x_i and y_i of heliostat H_i with center point p_i [47, p. 12]

Because the incoming solar vector τ_{solar} , see Section 1.2.1, and outgoing reflective vector r_i are known (see Figure 1.4), the normal vector n_i of heliostat H_i can be computed by

$$n_i = \frac{r_i + \tau_{\text{solar}}}{|r_i + \tau_{\text{solar}}|}. \quad (1.5)$$

The axis x_i and y_i (see Figure 1.3) of an aligned heliostat H_i can be computed by

$$x_i = \frac{n_i \times (0, 0, 1)^{\text{tr}}}{|n_i \times (0, 0, 1)^{\text{tr}}|} \quad (1.6a)$$

and

$$y_i = n_i \times x_i. \quad (1.6b)$$

While the heliostat scaffold is aligned, the alignment of the facets on the scaffold is fixed and therefore stays the same.

1.3.5. Canting

The orientation and positioning of facets on a large heliostat can be done in different ways. This is called canting. In canting, each individual facet of a heliostat is viewed and aligned to a focus point for a specific sun direction \vec{s} . Given sun direction \vec{s} the

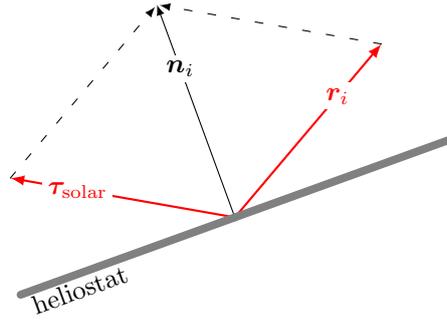
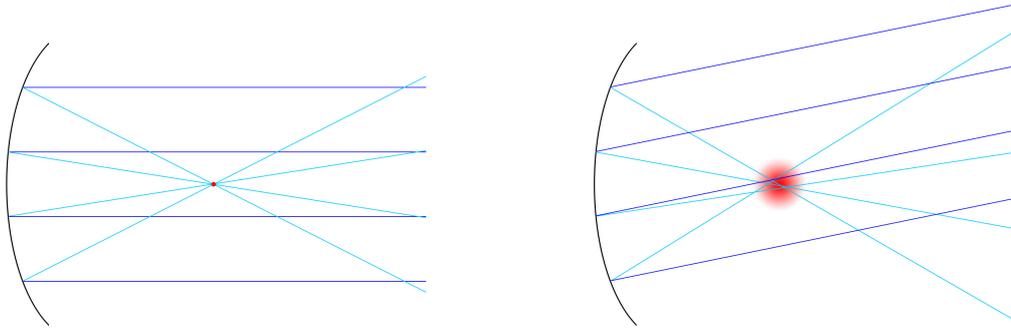


Figure 1.4: Reflection on the surface of a mirror

reflected light concentrates in the focus point, see Figure 1.5a. For other sun directions than \vec{s} the light comes in at a tilted angle. Therefore the heliostat will now focus the light to an area instead of a single point, see Figure 1.5b.



(a) Light is focused to a single point

(b) Light is focused to an area

Figure 1.5: Effects of same canting focus point at different sun directions.

In the 3D case each facet on a heliostat is oriented at one focus point for a given sun direction \vec{s} , see Figure 1.6. The calculation of the orientation of the facets can be done in two ways which will be described in the following.

On-axis canting calculates a paraboloid centered in the heliostats position with the focus point on the receiver where the sunlight emerges from the receiver. This means that the heliostat, receiver and the sun are all on one common axis. The facets are positioned around the symmetry axis of the paraboloid.

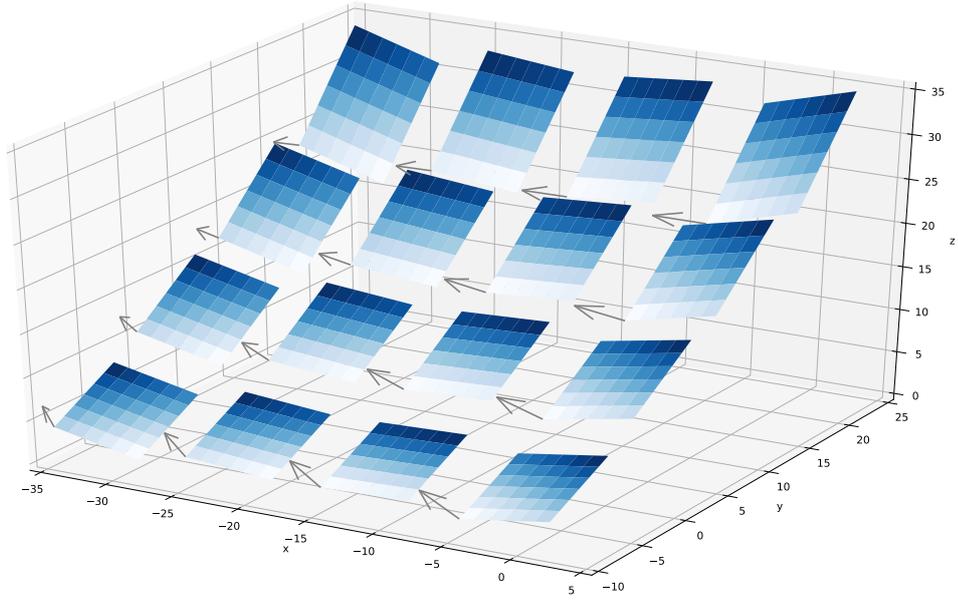


Figure 1.6: Canting visualization with each facet and their normal vector on one heliostat.

Off-axis canting calculates a paraboloid on the heliostats position with the focus point on the receiver, where the sunlight emerges from a specified direction \vec{s} . In our model \vec{s} can be given by a date and time or by the angles azimuth and altitude. Therefore the heliostat, receiver and sun are not on one common axis. The facets are positioned on one side of the paraboloid.

1.3.6. Focal points of facets

Heliostat facets can either be flat or focused. When focused they have a parabolic form with a focal length f_i . The focal length can be set fixed to a given value from a set of focal lengths or to the ideal focal length. When giving a set of focal length the model chooses the focal length nearest to the distance of the heliostats position p_i to the aiming point on the receiver $p_{\text{aim},i}$. The ideal focal length always corresponds to the distance between the aiming point on the receiver surface and the heliostat position $f_i = |p_{\text{aim},i} - p_i|$.

1.4. Tower

The tower is placed at the position $p_{\text{tower}} = (x_{\text{tower}}, y_{\text{tower}}, z_{\text{tower}})^T$, while z_{tower} is given by the topography, see Section 1.1.4. The tower is assumed either to be a cuboid with length ℓ_{tower} , width w_{tower} and height h_{tower} , or a cylinder with radius r_{tower} and height h_{tower} , see Figure 1.7. The tower expansion d_{tower} is the diameter of the minimum

bounding circuit in the $\mathbf{x-y}$ plane. So, $d_{\text{tower}} = \sqrt{\ell_{\text{tower}}^2 + w_{\text{tower}}^2}$ is calculated for a rectangular tower or $d_{\text{tower}} = 2 \cdot r_{\text{tower}}$ for a cylindric tower.

Its dimensions are later interesting for detecting its projected shadow on the heliostat field. The tower has an orientation angle α_{tower} in the $\mathbf{x-y}$ plane. This angle is defining the main facing direction of the tower. The angle is measured in a clockwise manner from the North. In the northern hemisphere, the heliostats are mainly placed in the north of the tower ($\alpha_{\text{tower}} = 0^\circ$), whereas in the southern hemisphere they are mainly placed in the south of the tower ($\alpha_{\text{tower}} = 180^\circ$). The vector in facing direction, which determines the receiver orientation is given by

$$\vec{f}_{\text{tower}} = \frac{d_{\text{tower}}}{2} \cdot \begin{pmatrix} -\sin \alpha_{\text{tower}} \\ \cos \alpha_{\text{tower}} \\ 0 \end{pmatrix}. \quad (1.7)$$

1.5. Receiver

At the receiver the reflected rays are collected and their radiation is transferred into heat. The receiver is mounted h_{top} meters below the top of the tower. For the calculation of a flux map (see Section 1.6.2) we discretize the receiver into cells.

There are several concepts which divide into two groups: cavity and external receiver [4]. In our model we distinguish between a flat tilted cavity receiver, cylindric cavity receiver and cylindric external receiver, see Figure 1.7.

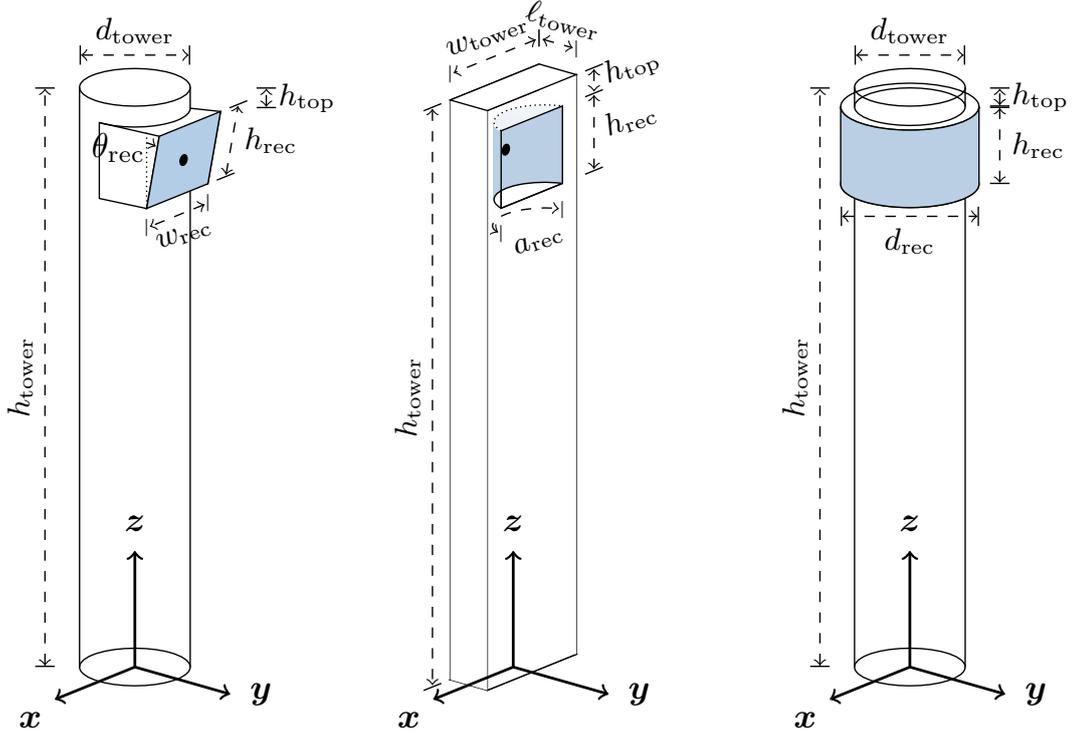
1.5.1. Flat tilted cavity receiver

A Flat tilted cavity receiver can be found in the CESA-1 central receiver facility in Andalusia, Spain or the solar tower Jülich in Germany. They represent a cavity or volumetric receiver. The receiver is modeled as a bounded plane in the $\mathbf{x-z}$ plane, which is tilted by a zenith angle θ_{rec} in the facing direction of the tower \vec{f}_{tower} . The receiver has a width of w_{rec} and height h_{rec} , see Figure 1.7a. For this type of receiver, the model assumes that each heliostat H_i aims towards the center of the aperture, so that

$$p_{\text{aim},i} = p_{\text{tower}} + \vec{f}_{\text{tower}} + \begin{pmatrix} -\frac{h_{\text{rec}}}{2} \cdot \sin \theta_{\text{rec}} \cdot \sin \alpha_{\text{tower}} \\ \frac{h_{\text{rec}}}{2} \cdot \sin \theta_{\text{rec}} \cdot \cos \alpha_{\text{tower}} \\ h_{\text{tower}} - h_{\text{top}} - \frac{h_{\text{rec}}}{2} \cdot \cos \theta_{\text{rec}} \end{pmatrix}. \quad (1.8)$$

1.5.2. Cylindric cavity receiver

This receiver has the form of a half-cylinder, such as the PS10 receiver in Andalusia, Spain. The receiver has an arc length of a_{rec} and a height of h_{rec} . For this type of receiver, the model assumes that each heliostat H_i aims towards the center of the



(a) Flat tilted cavity receiver (b) Cylindric cavity receiver (c) Cylindric external receiver

Figure 1.7: Different receiver types on different tower types. The Figure is derived from Richter [47, p. 11]

aperture, see Figure 1.7b. The distance from the tower center to the receiver center point is

$$d_{\text{mid}} = r_{\text{tower}} - \frac{4 \cdot r_{\text{tower}}^2 - (2 \cdot \frac{a_{\text{rec}}}{\pi})}{2} \quad (1.9)$$

for cylindrical towers or

$$d_{\text{mid}} = \left| \frac{l_{\text{tower}}}{2} - \frac{a_{\text{rec}}}{\pi} \right| \quad (1.10)$$

for rectangular towers. With this distance the aiming point is given by:

$$p_{\text{aim},i} = p_{\text{tower}} + \begin{pmatrix} -d_{\text{mid}} \cdot \sin \alpha_{\text{tower}} \\ d_{\text{mid}} \cdot \cos \alpha_{\text{tower}} \\ h_{\text{tower}} - h_{\text{top}} - \frac{h_{\text{rec}}}{2} \end{pmatrix}. \quad (1.11)$$

1.5.3. Cylindric external receiver

This receiver is cylindrical wrapped around the tower for 360°. Cylindric external receivers are used in the Solar One and Solar Two central receiver facilities at Barstow

in California, USA and the 19 MW plant Gemasolar in Andalusia, Spain. The receiver is modeled with diameter d_{rec} of the cylinder in the $\mathbf{x}\text{-}\mathbf{y}$ plane and height h_{rec} , see Figure 1.7c.

For this type of receiver, the model assumes, that each heliostat H_i aims towards the closest point at the center of the aperture. In the $\mathbf{x}\text{-}\mathbf{y}$ plane, the angle between the \mathbf{x} -axis and the line from heliostat H_i at position $p_i = (x_i, y_i, z_i)^\top$ to the receiver center, is given by

$$\alpha_i = \text{atan} \left(\frac{y_i}{x_i} \right). \quad (1.12)$$

So, the aiming point for a heliostat H_i is given by

$$p_{\text{aim},i} = p_{\text{tower}} + \begin{pmatrix} \frac{d_{\text{rec}}}{2} \cos \alpha_i \\ \frac{d_{\text{rec}}}{2} \sin \alpha_i \\ h_{\text{tower}} - h_{\text{top}} - \frac{h_{\text{rec}}}{2} \end{pmatrix}. \quad (1.13)$$

1.6. Ray-tracer

To calculate the concentrated power of reflected rays at the receiver we have to trace them from the sun along the heliostat surface to the receiver. Therefore we use a hierarchical ray tracing approach [39, 6]. For our ray tracing methods, each facet surface is partitioned in a number of cells, see Figure 1.8. Each cell has one representative ray for which different effects, like blocking and shading, are calculated for the whole cell. The number of representative rays per heliostat facet is given by the number of rays in horizontal direction times the number of rays in vertical direction per facet. Here p_i^k is defined as the origin of the k -th representative ray at the surface of heliostat H_i and $\vec{r}_{k,i}$ as the direction of the k -th representative ray of heliostat H_i .

We then calculate the complete flux at the receiver surface according to our different methods. The integration over the flux on the receiver surface equals the optical power collected at the receiver.

1.6.1. Efficiencies and losses

When tracing each ray the ray is exposed to different effects and losses which influence the amount of power which is received by the receiver. In the following each effect and loss is described.

Cosine effects In order to reflect the sun light onto the receiver surface heliostats track the sun position as illustrated in Figure 1.4. The cosine effect η_{cos} describes the reduced projected area of the respective cell due to the tilted alignment of the heliostat. Cosine effects depend on the solar position and the location of the individual heliostat in relation to the receiver. The heliostat surface normal bisects the angle between the

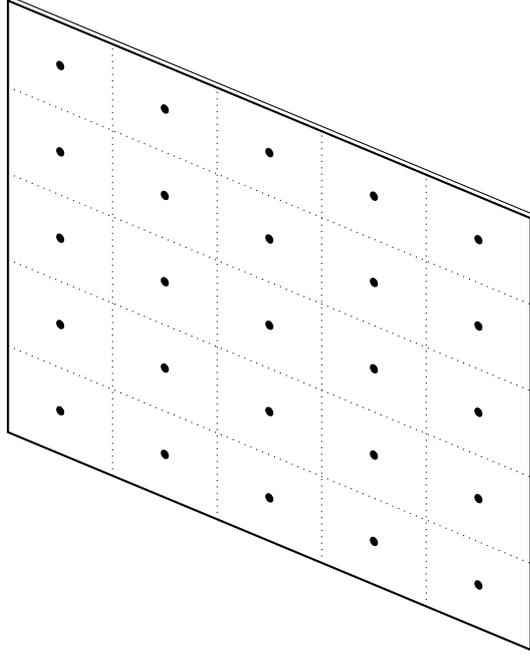


Figure 1.8: Discretization of a heliostat facet by 5 by 5 cells. Each cell has a representative ray, which is weighted by its area.

solar rays and a line from the heliostat to the receiver [20]. The effective reflection area of the heliostat is reduced by the cosine of one-half of this angle. It can easily be calculated using the law of reflection. The scalar product of the solar vector τ_{solar} and the heliostat normal n_i is related to the angle of incidence [39] (see Figure 1.4) so that

$$\eta_{\text{cos},i} = \langle \tau_{\text{solar}}, n_i \rangle, \quad (1.14)$$

where $\langle \cdot, \cdot \rangle$ describes the scalar product of two vectors.

Shading and blocking Shading effects appear when an object, e.g. a heliostat or a tower, is in-between the sun and another object. Blocking effects appear when the first object is in-between the receiver and the second object.

In this work we calculate shading effects by the tower as well as shading and blocking effects between the heliostats. In the following each computation is shortly described.

- **Tower shadow**

In Section 1.4 we defined a cuboid and a cylindrical tower, for these shapes incoming rays from the sun to the heliostats have to be checked if they hit the tower. The tower shadow can be described as a corridor with width d_{tower} facing away from the sun starting at the tower position p_{tower} . The heliostats in this corridor form a subset of potentially tower shaded heliostats.

This subset can be computed by calculating the minimal distance between the line from the tower position p_{tower} straight into the sky and the line from the

heliostat center p_i to the sun. If this distance is smaller or equal to $\frac{d_{\text{tower}}}{2} + \frac{d_{\text{helio}}}{2}$ the heliostat is potentially tower shaded.

For heliostats which are potentially shaded by the tower each single ray coming to the heliostat has to be checked. This means for each ray from the sun to a reflection point p_i^k on the heliostat surface a possible intersection has to be calculated. If there is an intersection the corresponding area of p_i^k is shaded by the tower.

- **Heliostat shading**

As for tower shading, a subset of heliostats can be computed by checking the minimal distance. Except in this case we don't look at the distance between two lines, but at the distance between a line and the center p_j of a heliostat H_j . The line originates at the center p_i of heliostat H_i and travels to the sun. If the minimal distance is smaller or equal to d_{helio} , then heliostat H_i is potentially shaded by heliostat H_j . Each ray from the sun with a reflection point p_i^k on heliostat H_i has to be checked if it hits heliostat H_j . If an intersection can be found the corresponding area to p_i^k is shaded.

- **Heliostat blocking**

For the computation of heliostat blocking we can utilize the same approach as in heliostat shading, when using the reflected vector $p_{\text{aim},i} - p_i$ instead of the vector from the sun to p_i .

Heliostat reflectivity The mirror surface reflects the solar radiation in direction of the receiver. Due to dust or absorbency of the mirror surface parts of the radiation are scattered or lost at the mirror surface. This loss is often modeled as constant value in literature [45], neglecting the dependency of the reflectivity on incidence angle and solar spectrum. We model the reflectivity as constant value η_{ref} .

Atmospheric attenuation efficiency When the light travels from the heliostat to the receiver it progressively loses power over the distance due to the atmosphere. This effect is called atmospheric attenuation efficiency and depends on the distance d_i . This is calculated between the position p_i of heliostat H_i , from which the ray originates, and the receiver aiming point $p_{\text{aim},i}$. The computation is given by

$$d_i = |p_i - p_{\text{aim},i}|. \quad (1.15)$$

We use the formula from Schmitz et al. [56], who extended the formula from Leary and Hankins [30] for distances less than 1000 m. The approach from Schmitz et al. [56] has the goal to correspond well with the model of Pitman and Vant-Hull [43] for a visual range of about 40 km (see Figure 1.9). The computation from [56] is

$$\eta_{\text{aa},i} = \begin{cases} 0.99321 - 1.176 \cdot 10^{-4} d_i + 1.97 \cdot 10^{-8} d_i^2 & , d_i \leq 1000 \text{ m} \\ \exp(-1.106 \cdot 10^{-4} d_i) & , d_i > 1000 \text{ m} \end{cases}. \quad (1.16)$$

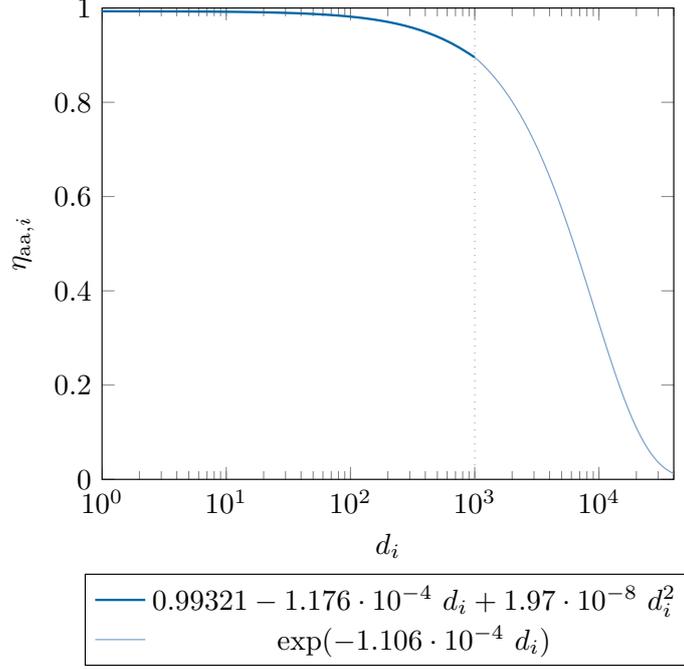


Figure 1.9: Atmospheric attenuation $\eta_{aa,i}$ computation as in Schmitz et al. [56].

Optical errors When modeling rays originating in the sun getting reflected at a heliostat to a receiver, errors can occur. These errors are called optical errors and can be described in means of a Gaussian distribution with a standard deviation of σ in [mrad], see Figure 1.10. In our model the errors can be set to specific values as well as default values.

- Sun Error σ_{sun} : Occurs due to the fact that we model the sun as a plane rather than a sphere. Therefore we add this error to model the sun as an angular Gaussian distribution based on the idea of Rabl [46].
- Tracking Error σ_{tracking} : The motor aligning the heliostat will in most time result in a slight deviation from the intended alignment.
- Slope Error σ_{slope} : The slope error describes the property of the mirror surface not being perfect, e.g. having a certain roughness which creates an irregular surface, reflecting rays in slightly different directions than the perfect reflection.

In our model we use one representative Gaussian distribution with standard deviation σ_{beam} which can be computed by

$$\sigma_{\text{beam}} = \sqrt{\sigma_{\text{sun}}^2 + \sigma_{\text{tracking}}^2 + \sigma_{\text{slope}}^2}. \quad (1.17)$$

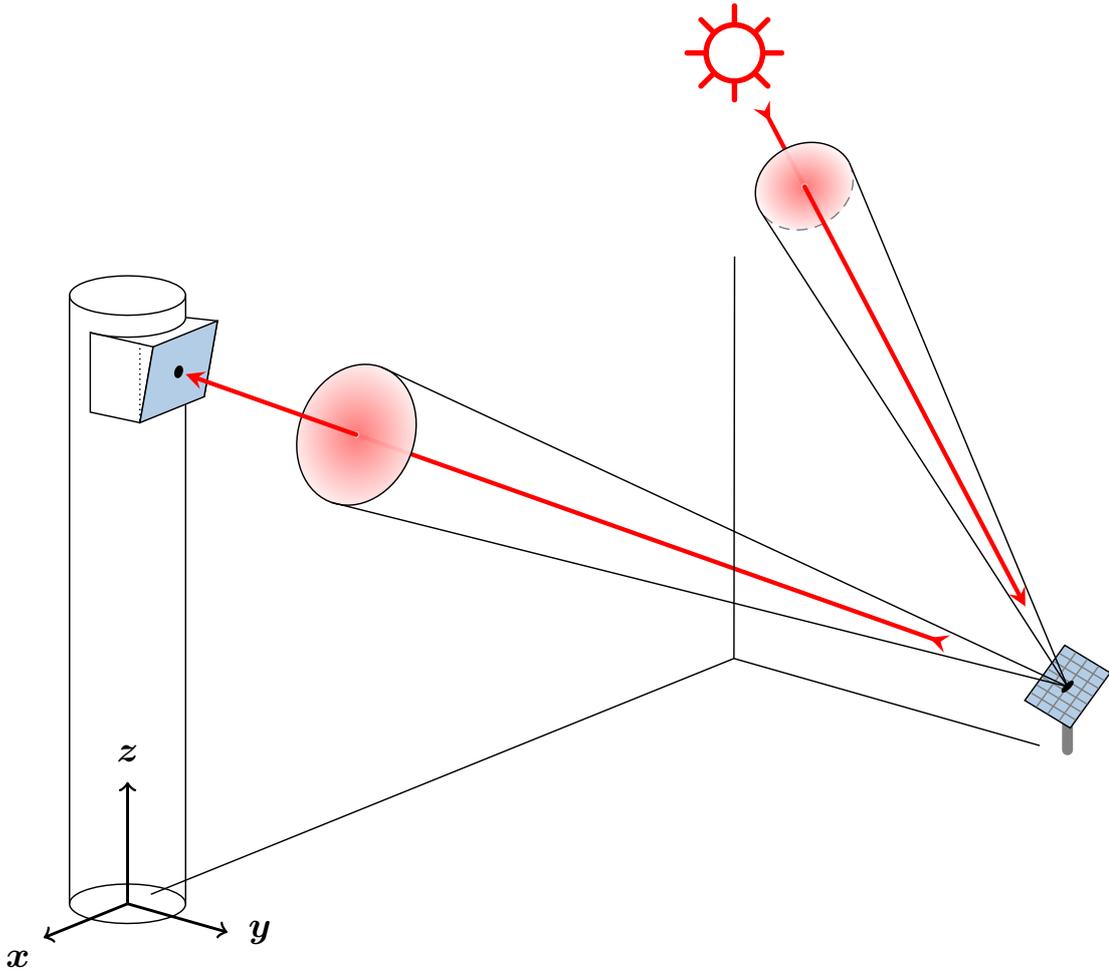


Figure 1.10: Error cones for optical errors [47].

1.6.2. Flux map

To calculate the distribution of the collected power on the receiver's surface we compute a so-called flux map. To achieve this, each receiver surface gets discretized into smaller cells. The total collected power is the sum of the power collected in each cell of the discretization. In the following, the calculation for the power collected in each cell is described in the corresponding methods.

1.6.3. Monte Carlo method

The Monte Carlo method relies on the law of large numbers. It therefore uses large amounts of randomized rays, of which each ray has its own source point on a heliostat. The randomization is achieved by perturbing each ray with a certain probability based on the Gaussian distribution with standard deviation σ_{beam} . In Figure 1.11a the Monte Carlo method is depicted for one heliostat aiming at the midpoint of the receiver at

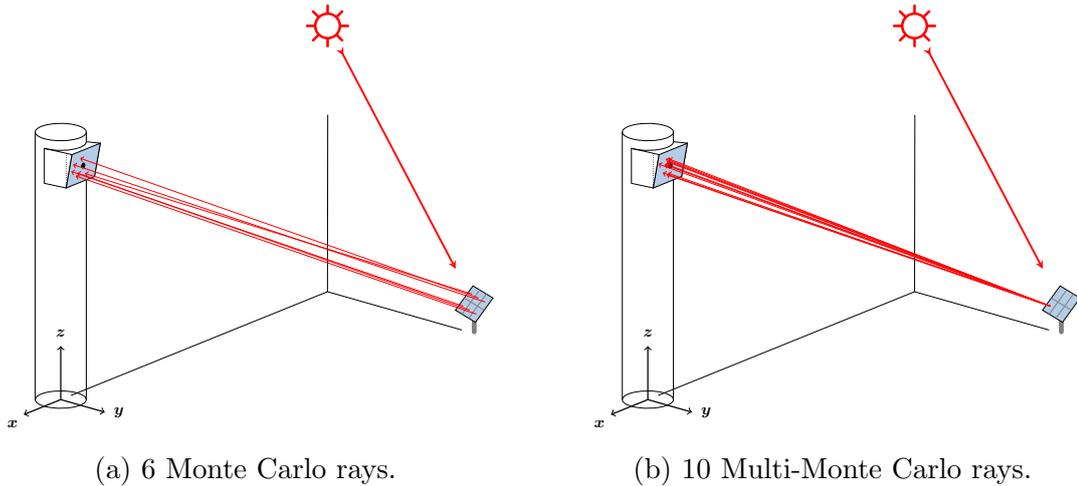


Figure 1.11: Monte Carlo and multi-Monte Carlo method with low resolution.

the top of the tower, whilst for the better clarity only 6 rays are shown. As the Figure shows, the rays do not hit the midpoint of the receiver, since they are slightly perturbed by the Monte Carlo method.

Multi-Monte Carlo The usage of multiple outgoing rays from one point on a heliostat defines the multi-Monte Carlo method. This aims at getting more diversity for each heliostat cell instead of just shooting one ray per cell. In Figure 1.11b the multi Monte Carlo method is shown for one source point with 10 rays.

Flux map In the Monte Carlo method for each receiver hitting ray the intersection with a cell of the receiver is calculated. Then the power of the ray is added up to the sum of power collected in that cell.

1.6.4. Convolution method

The convolution method shoots a ray which gets analytically distributed by the Gaussian normal distribution with standard deviation σ_{beam} on the receiver. This results in an error cone projected at the receiver with direction $\vec{r}_{k,i}$ and origin at the reflection point p_i^k on the heliostats surface, similar to the outgoing error cone in Figure 1.10. For the evaluation we use a normal distribution (see Figure 1.12 left) which we can evaluate at different points offset from the intersection point to get the flux (see Figure 1.12 right). The method is based on the method described in Richter et. al [47, 50].

Evaluation of the error cone To evaluate the flux produced by the error cone, we use the angle between the direction of the reflected ray $\vec{r}_{k,i}$ with origin p_i^k and the vector \vec{r}^j . The latter can be calculated with $p_{\text{rec}} - p_i^k$. The point at the receiver surface p_{rec} is where we want to evaluate the Gaussian distribution. Then we calculate the

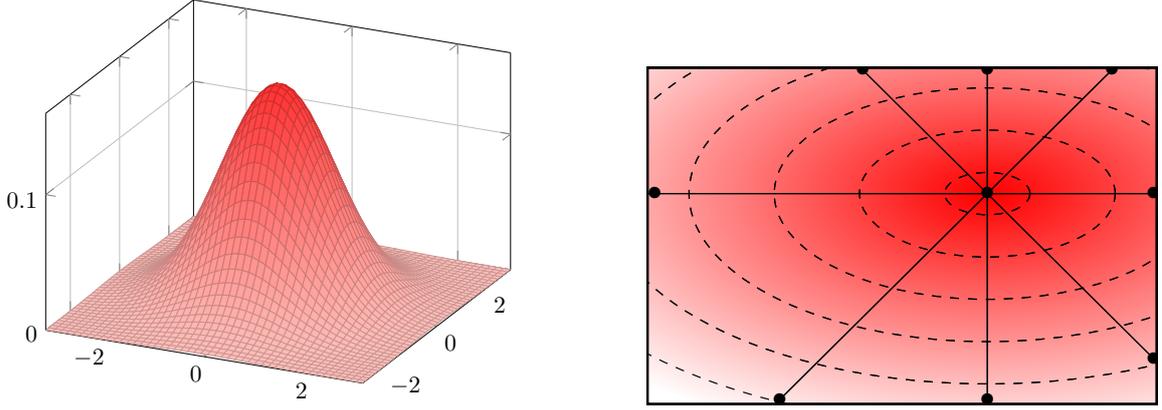


Figure 1.12: Gaussian normal distribution on a receiver [47]

length $|\vec{r}'|$ of the vector \vec{r}' . This length is used to approximate the length of the ray $\vec{r}_{k,i}$ until it hits the receiver. This makes the calculation faster since we don't have to calculate the exact intersection. We then compute the deviation σ_{rec} on p_{rec} with Equation (1.18). Next, we calculate the minimal distance $d_{k,i,\text{rec}}$ of the point p_{rec} to the ray with direction $\vec{r}_{k,i}$ and origin p_i^k . With this we can compute the probability density function value $P(d_{k,i,\text{rec}}, \sigma_{\text{rec}})$ for a point on the receiver with distance $d_{k,i,\text{rec}}$ to the center of the distribution according to Equation (1.19) from Abramowitz and Stegun [2] with $\mu = 0$.

$$\sigma_{\text{rec}} = |\vec{r}'| \cdot \tan(\sigma_{\text{beam}}) \quad (1.18)$$

$$P(d_{k,i,\text{rec}}, \sigma_{\text{rec}}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(d_{k,i,\text{rec}})^2}{2\sigma_{\text{rec}}^2}} \quad (1.19)$$

Flux map For the calculation of the flux map we use the previously described evaluation. For each ray we evaluate the error cone at the midpoints of the different receiver cells. We assume that the power for one ray is equivalent over one receiver cell. Therefore each cell contains the sum of the evaluations of all rays weighted by its own area.

1.6.5. Cell-wise convolution method

The cell-wise convolution method analytically aggregates each possible perturbed ray of a heliostat cell. It therefore projects the shape of the cell onto the receiver where the edges are blurred according to the Gaussian distribution with standard deviation σ_{beam} . Figure 1.13 shows the single rays of a heliostat cell in one dimension, the summed distribution as flux can be found in Figure 1.14.

This blurred projection of a heliostat cell onto a receiver requires different transformations. The evaluation at the receiver as well as the necessary transformations are described in the following.

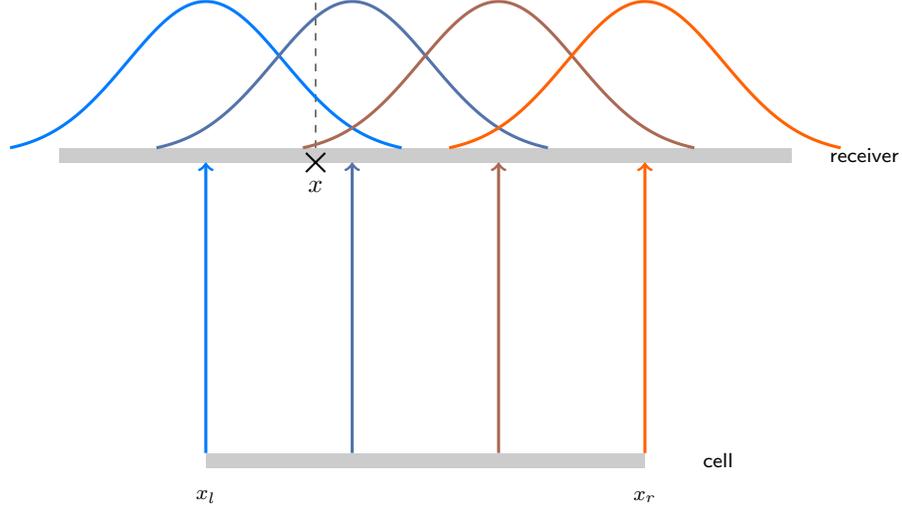


Figure 1.13: In the one-dimensional case, each ray reflected in the cell results in a Gaussian distribution on the receiver.

Evaluation at the receiver The evaluation of a projection at a point p_{rec} on the receiver assumes the heliostat cell from which the rays originates to be flat, centered in the origin, parallel to the \mathbf{x} - \mathbf{y} -plane and the sun has to be centered above the heliostat cell. The intensity at p_{rec} is computed by

$$I(p_{\text{rec}}) = \sum_{i=1}^n I_i(p_{\text{rec}}), \quad (1.20)$$

where $I_i(p_{\text{rec}})$ is the intensity produced by the i -th ray of the heliostat cell in point p_{rec} . Since each ray has variance σ_{beam}^2 the distribution of ray i is given by

$$I_i(x) = \frac{1}{\sqrt{2\sigma^2\pi}} \exp\left(-\frac{(x - \mu_i)^2}{2\sigma^2}\right). \quad (1.21)$$

Inserting this into the Equation (1.20) the total intensity can be computed by

$$I(x) = \frac{1}{\sqrt{2\sigma^2\pi}} \sum_{i=1}^n \exp\left(-\frac{(x - \mu_i)^2}{2\sigma^2}\right). \quad (1.22)$$

When considering infinitesimal rays in a cell in which $n \rightarrow \infty$, we need to integrate over all $\mu \in [x_{\text{min}}, x_{\text{max}}]$. This integral can be solved by

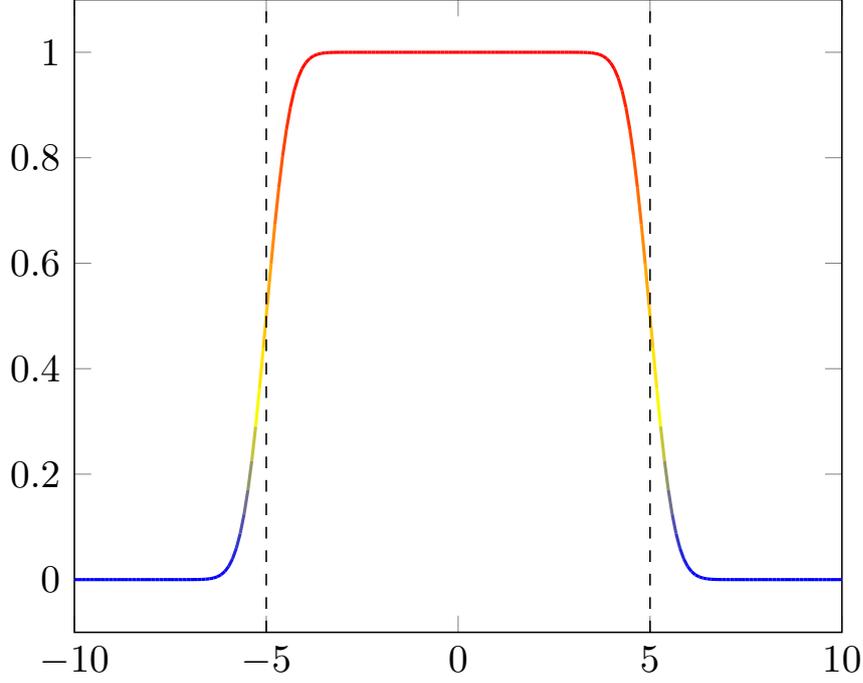


Figure 1.14: The flux of a cell with width $x_l = -5, x_r = 5$ and $\sigma^2 = 0.5$ on the receiver (one-dimensional case). The dashed lines mark the edges of the cell.

$$\begin{aligned}
 I(x) &= \frac{1}{\sqrt{2\sigma^2\pi}} \int_{x_{\min}}^{x_{\max}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) d\mu \\
 &= \frac{1}{\sqrt{2\sigma^2\pi}} \int_{x_{\min}-x}^{x_{\max}-x} \exp\left(-\frac{\mu^2}{2\sigma^2}\right) d\mu \\
 &= \frac{1}{2} \left[\operatorname{erf}\left(\frac{x_{\max}-x}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{x_{\min}-x}{\sqrt{2}\sigma}\right) \right].
 \end{aligned} \tag{1.23}$$

For the two dimensional case we need to integrate over the whole area of a heliostat cell, i.e. $\mu_x \in [x_{\min}, x_{\max}]$ and $\mu_y \in [y_{\min}, y_{\max}]$. Then the intensity $I(x, y)$ for a point $p_{\text{rec}} = (x, y)^T$ can be calculated by

$$\begin{aligned}
 I(x, y) &= \int_{x_{\max}-x}^{x_{\min}-x} \int_{y_{\max}-y}^{y_{\min}-y} \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{\mu_x^2 + \mu_y^2}{2\sigma^2}\right) d\mu_y d\mu_x \\
 &= \frac{1}{4} \left(\operatorname{erf}\left(\frac{y_{\max}-y}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{y_{\min}-y}{\sqrt{2}\sigma}\right) \right) \\
 &\quad \left(\operatorname{erf}\left(\frac{x_{\max}-x}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{x_{\min}-x}{\sqrt{2}\sigma}\right) \right).
 \end{aligned} \tag{1.24}$$

Facets and cells When looking at non-flat facets with a certain focal length, we discretize a facet such that one ray represents a cell of a facet according to Figure 1.8. As a simplification we assume each cell to be flat so that we do not have to consider curved shapes in the projection of heliostat cells onto the receiver. A sample discretization of a facet into 10 evenly spaced cells is shown in Figure 1.15.

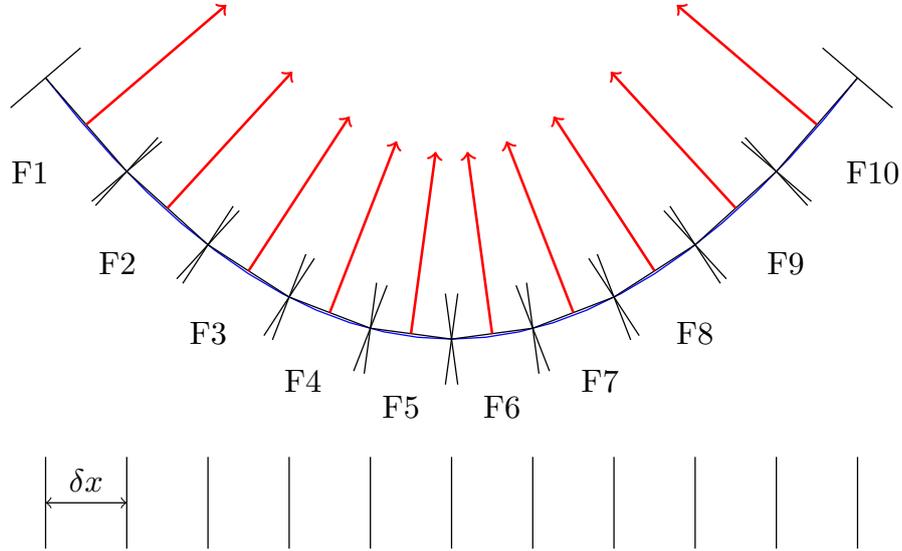


Figure 1.15: Evenly-spaced discretization of a facet with $N = 10$

Cell alignment In order to correctly project the shape of the heliostat cell onto the receiver we need to align the heliostat cell such that it is centered in the origin and parallel to the \mathbf{x} - \mathbf{y} -plane. The alignment of the cell consists of the following transformations:

- Translate the cell with its midpoint p_{cell} into the origin
- Rotate the cell such that its normal n_{cell} equals global \mathbf{z} -axis
- Rotate the cell such that its \mathbf{x} -axis x_{cell} equals global \mathbf{x} -axis

In the following the sun direction, according to these transformations, is denoted by \vec{v}_{sun} . The cell is shortened in the \mathbf{x} - and \mathbf{y} -axis according to the cosine effect with factors $\gamma_x = |\sin \angle \vec{v}_{\text{sun}} \vec{e}_x|$, $\gamma_y = |\sin \angle \vec{v}_{\text{sun}} \vec{e}_y|$. Whereas $\angle \vec{x} \vec{y}$ is the angle between the vectors \vec{x} and \vec{y} :

$$\begin{aligned} x'_{\min} &= \gamma_x x_{\min} \\ x'_{\max} &= \gamma_x x_{\max} \\ y'_{\min} &= \gamma_y y_{\min} \\ y'_{\max} &= \gamma_y y_{\max} \end{aligned}$$

The rotation from the sun up to the point that it is centered above the transformed cell is described by a rotation from the vector \vec{v}_{sun} onto the heliostat cell normal n_{cell} . Applying all these transformations to the different p_{rec} allows us to use the interval from Equation 1.24 for all evaluations.

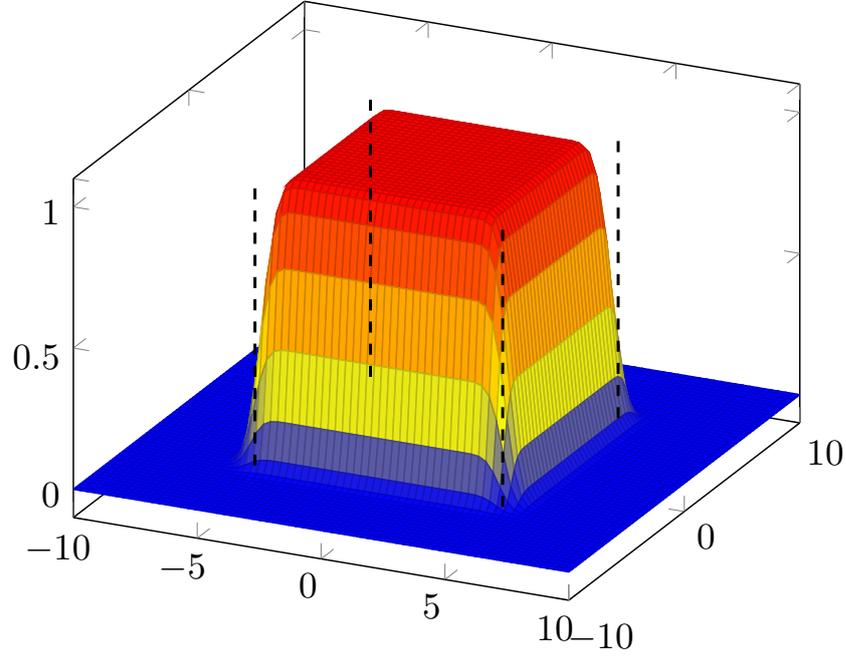


Figure 1.16: The flux of a single heliostat cell on the receiver in the two-dimensional case with $x'_{\min} = -5, x'_{\max} = 5, y'_{\min} = -5, y'_{\max} = 5, \sigma = 0.4$. The dashed lines mark the edges of the cell.

Flux map For the calculation of the flux map we use the previously described evaluation. For each cell we evaluate the projection at the midpoints of the different receiver cells. We assume that the power for one ray is equivalent over one receiver cell. Therefore each cell contains the sum of the evaluations of all heliostat cells weighted by its own area. Figure 1.16 shows an example flux for the projection of one heliostat cell.

2. Thermal model

Heiming [24] developed a simplified thermal model which bases on molten salt as heat transfer fluid (HTF) and a cylindrical external receiver. Therefore this section mainly relies on the work of Heiming [24].

2.1. Receiver

The receiver converts radiant power collected by the reflection of light on the heliostat surface into thermal power, i.e. inner power of the heat transfer medium. The HTF can be of a different kind, for example ambient air [25], pressurized air [33], water/steam [16], particles [53], or molten salt [29]. In our model only molten salt is considered as HTF. Furthermore the receiver model is simplified. This is caused by the need for annual energy optimization (see Part III).

Geometry The receiver consists of multiple panels of receiver height. Each panel contains a set of tubes through which the HTF flows in parallel. The flow direction then alternates panel-wise [24]. The defined flow of the HTF is divided into two circuits. Both start in the northern most panels and exit on the southern most panels. The first flow alternates through the panels to the west and then crosses to the east where it alternates through the panels to the south. The second flow travels to the east, crosses to the west and then goes to the south. This flow pattern was shown by Wagner [see 63, Figure 20]. As the exact number of panels is not stated it can be computed by the diameter of the tubes and the diameter of the receiver,

$$N_{\text{tubes/panel}} = \left\lfloor \frac{\pi \cdot D_{\text{rec}}}{D_{\text{tube,outer}} \cdot N_{\text{panels}}} \right\rfloor, \quad (2.1)$$

with $\lfloor x \rfloor = \lfloor |x| + 0.5 \rfloor \cdot \text{sgn}(x)$.

2.2. Simplified receiver model

The simplified model for the calculation of the thermal energy produced is based on [63] and is still used in the System Advisor Model (SAM) [36]. In the simplified receiver model the following energy balance (2.2) from Heiming [24] applies for the thermal power of the HTF \dot{Q}_{htf} , the incident radiation \dot{Q}_{inc} , the reflections losses \dot{Q}_{ref} , the radiation losses \dot{Q}_{rad} and the convection losses \dot{Q}_{conv} ,

$$\dot{Q}_{\text{htf}} = \dot{Q}_{\text{inc}} - \left(\dot{Q}_{\text{ref}} + \dot{Q}_{\text{rad}} + \dot{Q}_{\text{conv}} \right). \quad (2.2)$$

Each heat flow \dot{Q} is the sum of its sub heat flows \dot{q}_i , i.e. the heat flows of the single cells of the receiver. As in Heiming [24] the following equations will mark length-related quantities with a prime and area-related quantities with a double prime (e.g., \dot{q}_{inc}'' [W m⁻²] is the area-related incident radiation, also called flux).

In the following sections the computation of the incident radiation and the heat losses for the thermal receiver are shown.

2.2.1. Incident radiation

A flux map can be used to calculate the incident radiation. Due to the insignificant amount of ambient radiation, considering the large amount of incident radiation, it is not considered here. Generally the flux map can be defined as

$$\dot{q}_{\text{inc}}'' = \dot{q}_{\text{inc}}''(x, \varphi). \quad (2.3)$$

Heiming [24] then defines the flux of a receiver cell as integration over the corresponding area:

$$\dot{q}_{\text{inc}} = \int_{\delta\varphi} \int_{\delta x} \dot{q}_{\text{inc}}''(x, \varphi) dx d\varphi. \quad (2.4)$$

Since the flux map computed in Section 1.6 is all ready discretized into smaller cells with their flux we only need to match the smaller cells from the optical model to the bigger cells in the thermal receiver model. The overall incident radiation is then computed by

$$\dot{Q}_{\text{inc}} = \sum \dot{q}_{\text{inc}}. \quad (2.5)$$

2.2.2. Reflection

Since each material is reflecting some radiation the receiver cannot hold to the whole incident radiation. This is described by the fractions for absorptivity α , reflectivity ρ and transmissivity τ that sum up to 1 [28]. The transmissivity τ is 0 since the receiver is opaque. Therefore $\rho = 1 - \alpha$ holds. The following equations from Heiming [24] drop the symbols ρ and τ and express the reflectivity in terms of α .

The heat loss at each position of the receiver surface can be described in terms of the reflection by

$$\dot{q}_{\text{ref}} = (1 - \alpha) \cdot \dot{q}_{\text{inc}}. \quad (2.6)$$

For surface materials where the absorptivity is wavelength dependent this circumstance should be considered in the notation. This holds for non-gray or selective surface materials. In the following the solar spectrum on the surface of the earth is notated by λ_{sun} .

$$\dot{q}_{\text{ref}} = (1 - \alpha(\lambda_{\text{sun}})) \cdot \dot{q}_{\text{inc}}. \quad (2.7)$$

Inserting this equation to the local energy balance, it can be simplified from

$$\dot{q}_{\text{htf}} = \dot{q}_{\text{inc}} - (\dot{q}_{\text{ref}} + \dot{q}_{\text{rad}} + \dot{q}_{\text{conv}}) \quad (2.8a)$$

to an efficiency factor of the incoming incident radiation

$$\dot{q}_{\text{htf}} = \alpha \dot{q}_{\text{inc}} - (\dot{q}_{\text{rad}} + \dot{q}_{\text{conv}}). \quad (2.8b)$$

2.2.3. Radiation

Since the temperature of the receiver surface is above absolute zero the receiver emits radiation to the environment. According to Kirchoff's law (2.9) the emissivity ε corresponds to the absorption α for each wavelength of light [28].

$$\varepsilon(\lambda) = \alpha(\lambda) \quad (2.9)$$

We assume that there are three temperatures given in Kelvin T_{sky} , T_{amb} , and T_{dp} . Where T_{sky} describes the temperature at the horizon, e.g. the sky, T_{amb} describes the ambient temperature of the environment below the horizon, and T_{dp} is the ambient dew point temperature. In the following h denotes the solar time in hours which is 0 at solar noon, negative in the morning, and positive in the afternoon. The temperature T_{sky} can be calculated by Equation (2.10) given by Duffie and Beckman [15].

$$T_{\text{sky}} = T_{\text{amb}} \left(0.711 + 0.0056 (T_{\text{dp}} - 273.15) + 0.000073 (T_{\text{dp}} - 273.15)^2 + 0.013 \cos \left(\pi \left(\frac{180 - h \cdot 15}{180} \right) \right) \right)^{1/4} \quad (2.10)$$

Since half the radiation from the receiver is in direction of the sky, we use a view factor of $\frac{1}{2}$. The Stefan-Boltzmann constant $\sigma = 5.67 \cdot 10^{-8} \text{ W m}^{-2} \text{ K}^{-4}$ is denoted by σ and the wall surface temperature by T_{wall} which computation can be found in [24]. Heiming [24] then gives the following equations for the radiation to the sky and the ground respectively:

$$\dot{q}_{\text{rad,sky}} = \frac{1}{2} h_{\text{rad,sky}} A (T_{\text{wall}} - T_{\text{sky}}) \quad (2.11)$$

$$\text{with } h_{\text{rad,sky}} = \sigma \varepsilon (T_{\text{wall}}^2 + T_{\text{sky}}^2) (T_{\text{wall}} + T_{\text{sky}})$$

$$\dot{q}_{\text{rad,amb}} = \frac{1}{2} h_{\text{rad,amb}} A (T_{\text{wall}} - T_{\text{amb}}) \quad (2.12)$$

$$\text{with } h_{\text{rad,amb}} = \sigma \varepsilon (T_{\text{wall}}^2 + T_{\text{amb}}^2) (T_{\text{wall}} + T_{\text{amb}})$$

The full radiation loss is then simply the sum of both [63], i.e.,

$$\dot{q}_{\text{rad}} = \dot{q}_{\text{rad,sky}} + \dot{q}_{\text{rad,amb}}. \quad (2.13)$$

When choosing selective materials as receiver surface the absorptivity is high in the wave length range of the sun's radiation while emissivity is low in the range of wavelengths that are emitted at typical receiver surface temperatures. This leads to minimization of radiation losses at the receiver surface [24].

2.2.4. Convection

The transfer or dissipation of heat in fluids, like liquids or gas, is called convection. There are two types of convection of interest here. The first one is called forced convection caused by a flowing fluid. The second one is natural convection which depends on gravity and thermal buoyancy.

Heiming [24] states that the convection can be computed similar to Fourier's law of heat conduction. This is a product of heat transfer coefficient, surface area and temperature difference:

$$\dot{Q}_{\text{conv}} = h_{\text{conv}} A (T_{\text{wall}} - T_{\text{film}}) \text{ with } T_{\text{film}} = \frac{T_{\text{wall}} + T_{\text{amb}}}{2} \quad (2.14)$$

The forced convection coefficient can be computed by Equation (2.15a) from Heiming [24]. In Equation (2.15b) from Heiming [24], the computation of the natural convection is shown. The conductivity of air at film temperature T_{film} is denoted by k_{film} . The combination of forced and natural convection to the mixed convection is taken from Siebers and Kraabel [60] and can be found in Equation (2.15c) Siebers and Kraabel [60] recommend the exponent $a = 3.2$.

$$h_{\text{conv,for}} = Nu_{\text{conv,for}} \frac{k_{\text{film}}}{D_{\text{rec}}} \quad (2.15a)$$

$$h_{\text{conv,nat}} = Nu_{\text{conv,nat}} \frac{k_{\text{film}}}{h_{\text{rec}}} \quad (2.15b)$$

$$h_{\text{conv,mixed}} = (h_{\text{conv,for}}^a + h_{\text{conv,nat}}^a)^{1/a} \quad (2.15c)$$

The computation of the Nusselt numbers $Nu_{\text{conv,for}}$ and $Nu_{\text{conv,nat}}$ can be found in Heiming [24] and is based on the work of Siebers and Kraabel [60] and Siebers et al. [61].

2.3. Thermal storage

The advantage of modern solar tower power plants over photovoltaics is the use of a thermal storage system. This stores the heated fluid when the request is low and then can later on use the HTF when needed. Therefore peak loads or storage strategies which release the most power when the reimbursement is high can be satisfied.

The thermal storage model assumes that the installed storage space is large enough. From this storage the HTF is pumped into our steam generator, when there are peak loads, i.e. at times with the highest reimbursement.

3. Electrical model

In the electrical model the previously collected thermal power gets converted into electrical power. The model is also called Power Block or Power Conversion Unit and consists of a steam generator, a turbine, a generator and a cooling system. Instead of modeling each component on its own we use a lookup table for the whole power conversion unit [24]. The efficiency of the power block depends on the load and the ambient temperature, whereas a higher load and lower ambient temperature result in a better efficiency.

In Figure 3.1 from Heiming [24] the lookup table is pictured as efficiency over different loads at different temperatures from data of the company TSK Flagsol. When calculating the efficiency for some design point we use bilinear interpolation between the next higher measured load and temperature and the next lower load and temperature. In Figure 3.2 from Heiming [24] the data is smoothed by the bilinear interpolation.

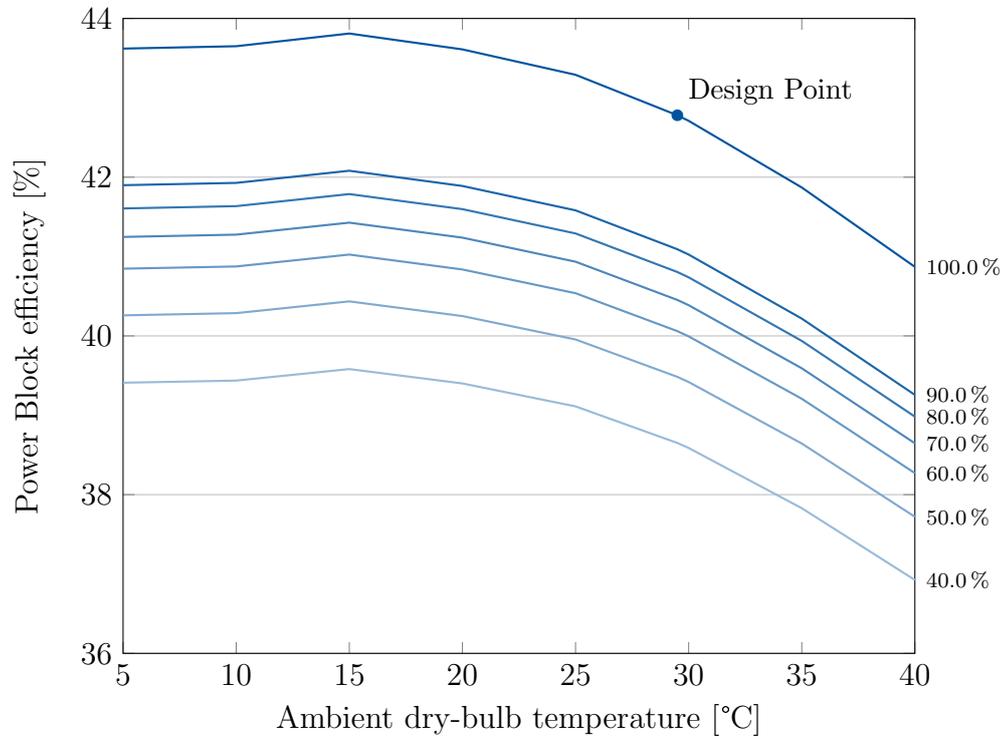


Figure 3.1: Characteristic Diagram of a $100 \text{ MW}_{\text{th}}$ power conversion unit. The lines represent the temperature-dependent efficiencies for different loads.

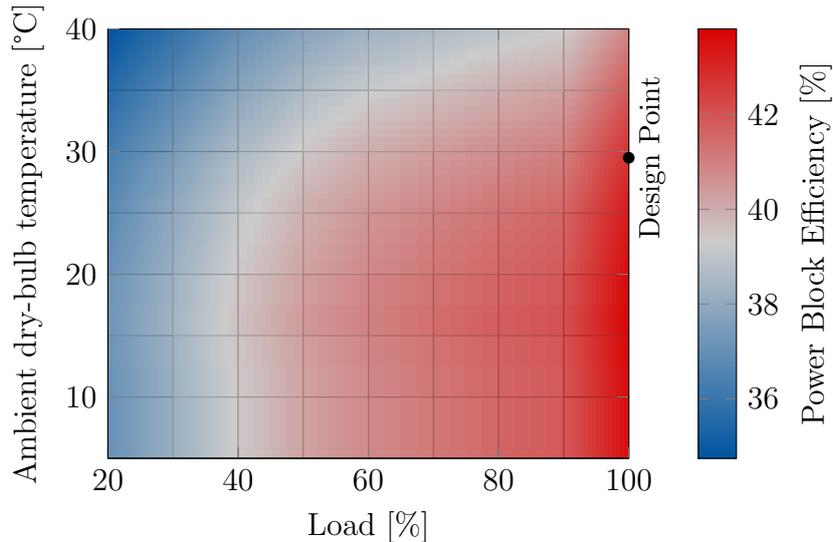


Figure 3.2: Efficiency of a 100 MW_{th} power block depending on ambient dry-bulb temperature and load.

4. Annual integration

The annual integration is used to calculate the annual energy produced by a solar tower power plant. The annual integration equals the sum over the produced energy of all days of the year, whereas the daily energy equals to the integral over the hourly calculated power from sunrise to sunset, see Equation (4.1). Since the exact calculation of Equation (4.1) would yield to high runtimes, Tinnes [62] provides a study on quadrature methods that can be used to approximate the integral. We use the previously described models (see Sections 1-3) to calculate P for a given moment. This section gives an overview of the used quadrature method in **SunFlower** to calculate sample points in a year to approximate Equation (4.1) as well as how to get the DNI and other meteorological data at that sample points using real weather data and the meteorological radiation model (MRM). This section is based on the work of Tinnes [62] as part of the **SunFlower** project.

$$E_{\text{annual}} = \sum_{d=1}^{365} E_d = \sum_{d=1}^{365} \left(\int_{\text{sunrise}}^{\text{sunset}} P(t, d) dt \right) \quad (4.1)$$

with d : number of days, t : time, P : produced power.

4.1. Quadrature

To get the specified day points, which have to be simulated, we use the trapezoidal quadrature method. This leads to equidistant days during the year. For the times at the days we get equidistant times between sunset and sunrise. Whilst we do not

simulate sunrise and sunset, because of an irradiation of 0 W m^{-2} , we take them into account as weights when calculating the daily energy. In Figure 4.1 we see the trapezoidal quadrature-rule for calculating the energy over a day with 7 daypoints.

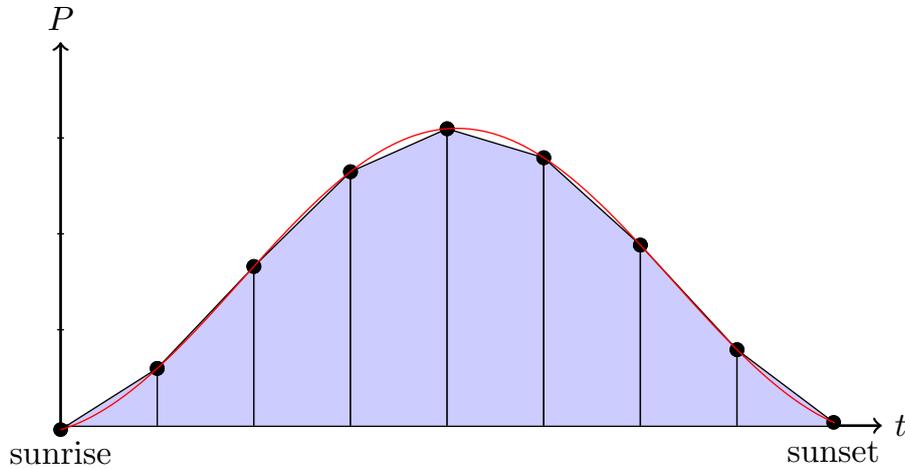


Figure 4.1: Visualization of the trapezoidal quadrature-rule for 7 daypoints[62].

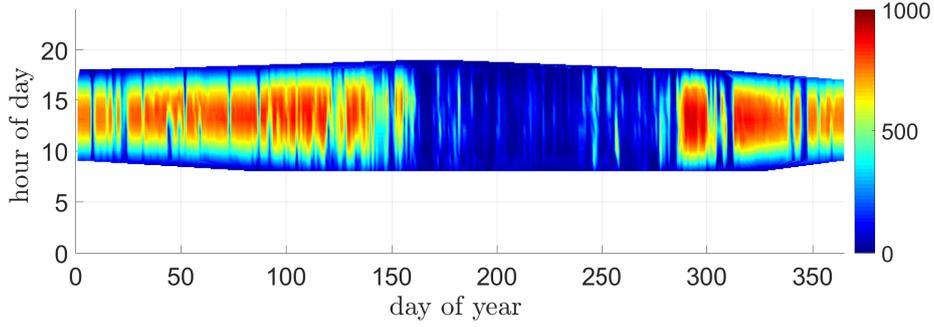
4.2. Real weather data

For the input of real weather data we support the typical meteorological year (TMY) data files from the National Renewable Energy Laboratory [5] and the energy plus weather (epw) data files from EnergyPlus [40]. Both datasets provide hourly measured DNI, environment pressure, dew point temperature and other values for a specified location for one year. Since these models have seasonal dependencies and are influenced by different weather situations we have to reconstruct the data according to Tinnes [62]. However we let the user decide if the specified points from the quadrature should be used without reconstruction leading to the actual measured values at one day for the given period. Figure 4.2a shows the measured DNI for Mumbai from EnergyPlus [40] and Figure 4.2b shows the measured DNI for Daggett as TMY3 [5].

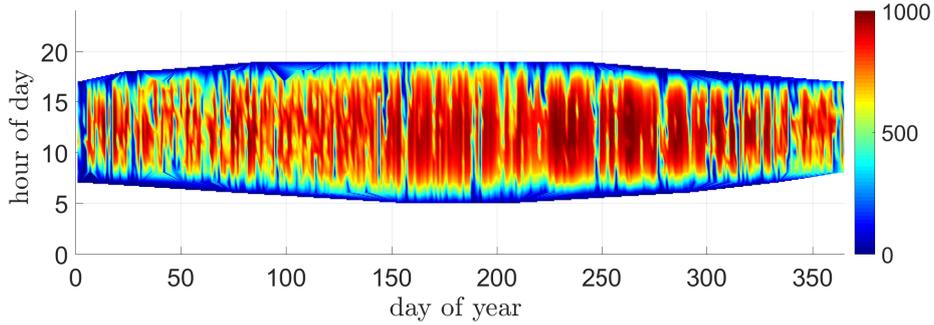
Reconstruction The reconstruction is an aggregation of the data for the given period. Therefore we take the average of the data for all days in the given period. The averaged data is then used in the different models as an aggregated day representing the whole period.

4.3. MRM

The MRM provides the irradiation I_{dni} as function over latitude, longitude, angstrom alpha α , visibility τ_{vis} , air pressure p_{air} , water pressure p_{water} , ground albedo ρ_{ground} , time t and altitude θ_{solar} . Since the irradiation is the only value calculated by the MRM



(a) Measured DNI at Mumbai (India) for each hour of a year[62].



(b) Measured DNI at Daggett (USA) for each hour of a year[62].

Figure 4.2: Measured DNI at the locations Mumbai and Daggett[62].

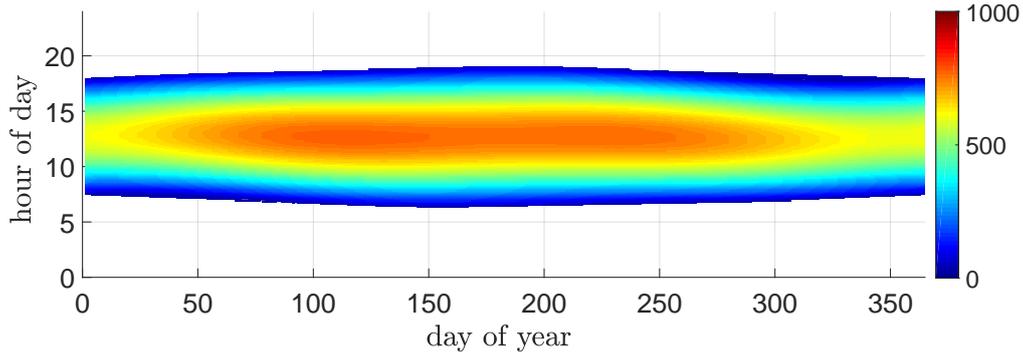
we have to choose symbolic values for the rest of the data, see Table 2. The DNI values show symmetrical behavior over one year as can be seen in Figure 4.3 for the locations Mumbai and Daggett. Therefore we do not have to reconstruct the calculated values and can use them in the models of Sections 1-3 [62].

Parameter	Value
dew point temperature	10°C
environment temperature	15°C
wind speed	3 m/s

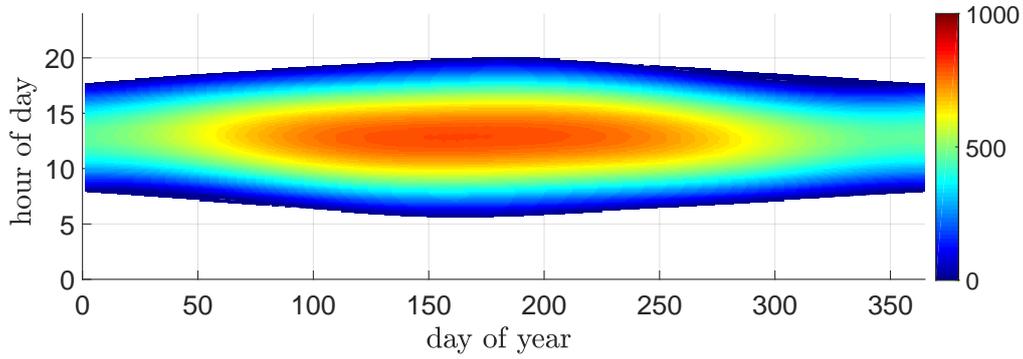
Table 2: Values not computed by MRM

4.4. JSON file

In Table 3 the parameters of the JSON file specifying the settings for the MRM are shown. In Table 4 the parameters of the JSON file specifying the settings for the annual integration are shown.



(a) Modeled DNI for one year at Mumbai with the MRM model[62].



(b) Modeled DNI for one year at Daggett with the MRM model[62].

Figure 4.3: Modeled DNI values, computed with the MRM model at the locations Mumbai and Dagget[62].

Parameter name	Unit	Data Type	Range
angstrom alpha	-	double	[-4, 2]
ground albedo	factor	double	[0, 1]
visibility	km	double	[1, 335]
air pressure	hPa	double	[800, 1100]
water pressure	hPa	double	[0, 101]

Table 3: JSON file for the MRM as list of its parameters.

Parameter name	Unit	Data Type	Range
number of periodpoints	count	int	[1,365]
number of daypoints	count	int	[1,24]
annual integration method	-	enum	{const day, aggregated day}

Table 4: JSON file for the annual integration as list of its parameters.

5. Economic model

This section describes the economic factors when modeling a solar tower power plant based on the work of Heiming [24]. The costs for a solar tower power plant are split into two parts. The investment costs, called capital expenditure (CAPEX). And the running costs paid on a regular basis, called the operational expenditure (OPEX). In the following these costs are described followed by an economic evaluation of the modeled plant based on these costs.

5.1. Investment cost

When building a solar tower plant an investment for parts of the building process have to be paid. These costs are the investment cost summarized under the term capital expenditure (CAPEX). The sum for the CAPEX, measured in [M\$], can be found in Equation (5.1) from Heiming [24]. According to [4] we will use the symbols I and c for investment and specific costs respectively.

$$CAPEX = I_{\text{land}} + I_{\text{hel}} + I_{\text{tower}} + I_{\text{rec}} + I_{\text{stor}} + I_{\text{pcu}} \quad (5.1)$$

When building such large scale projects as solar tower power plants present concepts of scaling and volume effects can be used to estimate the costs based on previous projects. The scaling effect derives a scaling factor from existing projects according to their sizes. With this scaling factor s we now can estimate the cost for the new project c_{new} according to the reference cost c_{ref} , the reference size A_{ref} and the new size A_{new} with Equation (5.2) [55].

$$c_{\text{new}} = c_{\text{ref}} \cdot \left(\frac{A_{\text{new}}}{A_{\text{ref}}} \right)^s \quad (5.2)$$

The volume effect describes the effects of decreasing costs due to increasing quantities and production experience. With reference production Volume V_{ref} , new production Volume V_{new} , a progress ratio pr and reference costs c_{ref} we can compute the estimated costs c_{new} with Equation (5.3) [55].

$$c_{\text{new}} = c_{\text{ref}} \cdot pr^{\log_2 \frac{V_{\text{new}}}{V_{\text{ref}}}} \quad (5.3)$$

With these effects we now can estimate the costs for the different investment costs from Equation (5.1) based on existing data.

Land The investment cost for the land of the building site is the sum of the terrain cost itself I_{terrain} and the cost for improving the terrain I_{improv} . The terrain cost is computed per square meter of the area of the site as can be seen in Equation (5.4a). The improvement can be estimated by previously build solar tower plants with a scaling factor as can be seen in Equation (5.4b). Then Equation (5.4c) is the sum for the

complete cost of the land I_{land} where reference values from Augsburg [4] can be used, see Table 8.

$$I_{\text{terrain}} = c_{\text{terrain}} A_{\text{terrain}} \quad (5.4a)$$

$$I_{\text{improv}} = I_{\text{improv,ref}} \cdot \left(\frac{A_{\text{terrain}}}{A_{\text{terrain,ref}}} \right)^{s_{\text{improv}}} \quad (5.4b)$$

$$I_{\text{land}} = I_{\text{terrain}} + I_{\text{improv}} \quad (5.4c)$$

Heliostats According to Heiming [24] the costs for a heliostat is the sum of several sub-costs: Material and labor costs, also called direct costs $I_{\text{hel,dir}}$, the optical costs $I_{\text{hel,optic}}$ that take the heliostats' slope into account, overhead costs $I_{\text{hel,overhead}}$ for management and engineering and indirect costs $I_{\text{hel,indir}}$ for additional tooling:

$$I_{\text{hel}} = I_{\text{hel,dir}} + I_{\text{hel,optic}} + I_{\text{hel,overhead}} + I_{\text{hel,indir}} \quad (5.5)$$

- **Direct costs** The investment on the direct cost again splits up into sub-costs which are listed in Table 5. These sub-costs $C_{\text{hel,dir},i}$ take the number of heliostats N_{hel} , the mirror area A_{hel} and the production volume V_{hel} , which is equal to the number of heliostats for one solar tower plant, into account and adding scaling and volume effects on them as well as a price index pi which describes a change of price over time of the reference price. The Equations (5.6) are the corresponding equations from Heiming [24].

$$I_{\text{hel,dir}} = N_{\text{hel}} \cdot C_{\text{hel,dir}} \quad (5.6a)$$

$$C_{\text{hel,dir}} = \sum_i C_{\text{hel,dir},i} \quad (5.6b)$$

$$C_{\text{hel,dir},i} = C_{\text{hel,dir},i,\text{ref}} \cdot \left(\frac{A_{\text{hel}}}{A_{\text{hel,ref}}} \right)^{s_{\text{hel,dir},i}} \cdot pr_{\text{hel,dir},i}^{\log_2 \frac{V_{\text{hel}}}{V_{\text{hel,ref}}}} \cdot pi_{\text{hel,dir},i} \quad (5.6c)$$

Heliostat direct cost i	$C_{\text{hel,dir},i,\text{ref}}$ [\$/u]	$s_{\text{hel,dir},i}$	$pr_{\text{hel,dir},i}$	$pi_{\text{hel,dir},i}$
Foundation	200	0.2274	0.9806	1.0816
Pedestal and structure	3 777	1.4700	0.9900	1.8070
Drives	6 000	0.6000	0.9400	1.3702
Mirrors	4 996	1.0420	0.9700	1.0861
Control and Communications	875	0.2311	0.9600	1.2841
Wiring	877	0.4479	1.0000	1.0302
Shop Fabrication	480	0.4264	0.9800	1.0000
Installation and Checkout	450	0.2610	1.0000	1.0000
Total reference direct costs	17 655			

Table 5: Estimated direct cost parameters of a heliostat [4].

- **Optical cost** The optical cost is related to the heliostat canting as described in Section 1.3.5. Equation (5.7) from Heiming [24] as reference value $\sigma_{\text{slope,ref}}$ the value from Table 8 can be taken.

$$I_{\text{hel,optic}} = N_{\text{hel}} \cdot c_{\text{hel,optic}} \quad (5.7a)$$

$$c_{\text{hel,optic}} = 0.01 \cdot 10^{-3} \left(\frac{1}{(\sigma_{\text{slope}})^2} - \frac{1}{(\sigma_{\text{slope,ref}})^2} \right) A_{\text{hel}} \quad (5.7b)$$

- **Overhead** Overhead costs are computed by a share or of the direct costs $I_{\text{hel,dir}}$ whilst considering a volume effect [4].

$$I_{\text{hel,overhead}} = or \cdot I_{\text{hel,dir}} \cdot pr_{\text{hel,overhead}}^{\log_2 \frac{V_{\text{hel}}}{V_{\text{hel,ref}}}} \quad (5.8)$$

- **Indirect cost** The indirect costs contain the costs for engineering and constructing the heliostats. The reference values for Equation (5.9) from Heiming [24] can be found in Table 6.

$$I_{\text{hel,indir}} = \sum_j I_{\text{hel,indir},j} \quad (5.9a)$$

$$I_{\text{hel,indir},j} = I_{\text{hel,indir},j,\text{ref}} \cdot \left(\frac{A_{\text{hel}}}{A_{\text{hel,ref}}} \right)^{s_{\text{hel,indir},j}} \cdot pr_{\text{hel,indir},j}^{\log_2 \frac{V_{\text{hel}}}{V_{\text{hel,ref}}}} \cdot pi_{\text{hel,indir},j} \quad (5.9b)$$

Heliostat indirect cost j	$I_{\text{hel,indir},j,\text{ref}}$ [\$]	$s_{\text{hel,indir},j}$	$pr_{\text{hel,indir},j}$	$pi_{\text{hel,indir},j}$
Engineering	250 000	0.9551	0.96	1.2623
Facilities and Tooling	800 000	0.9551	0.86	1.1460
Equipment Lease	200 000	0.9551	0.86	1.1460
Total reference indirect costs	1 250 000			

Table 6: Estimated indirect cost parameters of a heliostat [4].

Tower For the investments on the tower Heiming [24] used Equation (5.10). However since there is only one tower in our modeled solar tower power plants the volume effect doesn't play a role.

$$I_{\text{tower}} = I_{\text{tower,ref}} \cdot \left(\frac{h_{\text{tower}}}{h_{\text{tower,ref}}} \right)^{s_{\text{tower}}} \cdot pr_{\text{tower}}^{\log_2 \frac{V_{\text{tower}}}{V_{\text{tower,ref}}}} \cdot pi_{\text{tower}} \quad (5.10)$$

Receiver The receiver investments are similar to those of the tower. Further on the volume effect doesn't matter here either because there is only one receiver.

$$I_{\text{rec}} = I_{\text{rec,ref}} \cdot \left(\frac{A_{\text{rec}}}{A_{\text{rec,ref}}} \right)^{s_{\text{rec}}} \cdot pr_{\text{rec}}^{\log_2 \frac{V_{\text{rec}}}{V_{\text{rec,ref}}}} \cdot pi_{\text{rec}} \quad (5.11)$$

Storage The storage costs consider scaling effect, volume effect and price index. As can be seen in Equation (5.12) from Heiming [24].

$$I_{\text{stor}} = I_{\text{stor,ref}} \cdot \left(\frac{S_{\text{stor}}}{S_{\text{stor,ref}}} \right)^{s_{\text{stor}}} \cdot pr_{\text{stor}}^{\log_2 \frac{V_{\text{stor}}}{V_{\text{stor,ref}}}} \cdot pi_{\text{stor}} \quad (5.12)$$

Power conversion unit The investment on the power conversion unit considers scaling effect, volume effect and price index. The calculation is shown in Equation (5.13) from Heiming [24]. The reference cost can be found in Table 7.

$$I_{\text{pcu}} = \sum_k I_{\text{pcu},k} \quad (5.13a)$$

$$I_{\text{pcu},k} = I_{\text{pcu},k,\text{ref}} \cdot \left(\frac{S_{\text{pcu},k}}{S_{\text{pcu},k,\text{ref}}} \right)^{s_{\text{pcu},k}} \cdot pr_{\text{pcu},k}^{\log_2 \frac{V_{\text{pcu},k}}{V_{\text{pcu},k,\text{ref}}}} \cdot pi_{\text{pcu},k} \quad (5.13b)$$

PCU cost k	$S_{\text{pcu},k,\text{ref}}$	$I_{\text{pcu},k,\text{ref}}$	$s_{\text{pcu},k}$	$pr_{\text{pcu},k}$	$pi_{\text{pcu},k}$
Steam Generator	34.0 MW _{th}	1.6 M\$	0.6734	0.9526	1.4400
Steam Turbine and Generator	13.5 MW _{el}	8.8 M\$	0.6829	0.9526	1.2971
Cooling System	13.5 MW _{el}	7.4 M\$	0.2514	0.9526	1.2254
Master Control	–	1.6 M\$	–	–	1.1690
Total reference PCU costs		19.4 M\$			

Table 7: Estimated cost parameters of a power conversion unit [4].

Quantity	Value	Unit	Quantity	Value	Unit
c_{terrain}	0.5	[\$/m ²]	$A_{\text{terrain,ref}}$	2.8	[km ²]
$I_{\text{improv,ref}}$	1.1	[M\$]	s_{improv}	0.3687	[–]
$A_{\text{hel,ref}}$	148	[m ²]	$V_{\text{hel,ref}}$	1625	[u]
$\sigma_{\text{slope,ref}}$	4.14	[mrad]	or	20	[%]
pr_{overhead}	0.96	[–]	$I_{\text{tower,ref}}$	1.6	[M\$]
$h_{\text{tower,ref}}$	75	[m]	s_{tower}	1.797	[–]
pr_{tower}	0.9526	[–]	$V_{\text{tower,ref}}$	1	[u]
pi_{tower}	1.0816	[–]	$I_{\text{rec,ref}}$	9.1	[M\$]
$A_{\text{rec,ref}}$	100	[m ²]	s_{rec}	0.5283	[–]
pr_{rec}	0.9526	[–]	$V_{\text{rec,ref}}$	1	[u]
pi_{rec}	1.44	[–]	$I_{\text{stor,ref}}$	3.7	[M\$]
$S_{\text{stor,ref}}$	88.2	[MWh _{th}]	s_{stor}	0.6202	[–]
pr_{stor}	0.9526	[–]	pi_{stor}	2.2	[–]

Table 8: Reference and scaling values provided by Augsburger [4].

5.2. Operation and maintenance costs

The operation and maintenance costs are based on the equations of Morin [35]. In the following we will denote the running cost with C which consist of expenditures for staff, water, spare parts and insurance. These individual cost will be summed up to a per-year value which will be called OPEX, see Equation (5.14).

$$OPEX = C_{\text{staff}} + C_{\text{water}} + C_{\text{spare}} + C_{\text{insur}}. \quad (5.14)$$

In the following all parameters for the running costs as given by Morin [35] can be found in table 9. Further on some equations hold for parameters in € for them the factor f_{curr} converts euros to dollars to stay consistent with the previous calculation.

Staff Morin [35] states that the personal for the heliostat field grows linearly with its area, while the staff for the power conversion unit is fixed. Equation (5.15) therefore holds for a linear factor $f_{\text{staff,field}}$. Due to part-time jobs non-integer values are possible for the required personnel [24].

$$C_{\text{staff}} = (f_{\text{staff,field}} \cdot A_{\text{field}} + N_{\text{staff,pcu}}) \cdot c_{\text{staff}} \cdot f_{\text{curr}} \quad (5.15)$$

Water In solar tower power plants water is used for mirror cleaning and for wet-cooling systems for the power conversion unit. In this work we assume a wet-cooling system. Since the water consumption is only a very small part of the total annual cost and the mirror cleaning is only a small part of the water consumption Morin [35] states that the total mirror-area can be neglected in this calculation. Therefore we only consider the annual energy production (AEP) E_{annual} as factor since the water consumption in the cooling system of the power block depends on this. The calculation of the running costs for the water can be found in Equation (5.16).

$$C_{\text{water}} = E_{\text{annual}} \cdot f_{\text{water}} \cdot c_{\text{water}} \cdot f_{\text{curr}} \quad (5.16)$$

Spare parts The cost for spare parts depends on the CAPEX and is calculated as a fraction of it [24]:

$$C_{\text{spare}} = CAPEX \cdot f_{\text{spare}} \quad (5.17)$$

Insurance As the spare parts insurance payments depend on the CAPEX and are modeled as a fraction of it:

$$C_{\text{insur}} = CAPEX \cdot f_{\text{insur}} \quad (5.18)$$

Quantity	Value	Unit	Quantity	Value	Unit
c_{staff}	48 000	[€/a]	$f_{\text{staff,field}}$	$3 \cdot 10^{-5}$	[m ⁻²]
c_{water}	0.5	[€/m ³]	f_{water}	0.295	[m ³ /(MWh _{el})]
f_{spare}	1	[%]	f_{insur}	1	[%]

Table 9: Parameters for the running costs as given by Morin [35].

5.3. Economic evaluation

For the economical evaluation of the modeled solar tower power plant the previously described CAPEX and OPEX as well as the AEP E_a , the plant life time N_y and the interest rate i are used as parameters. Due to consistency with Augsburg [4] we use Equation (5.19) to transform the annual expenditure OPEX to operational expenditures OM which are measured in [¢/(kWh_{el})].

$$OM = \frac{OPEX}{E_a}. \quad (5.19)$$

Levelized cost of electricity The levelized cost of electricity (LCOE), also called levelized energy cost (LEC), is used for the evaluation of energy costs. It is the most common value and used for easy comparison of power plants of different types [24]. Heiming [24] uses the following definition for the LCOE:

$$\begin{aligned} LCOE &= \frac{\text{Annual costs}}{\text{Annual energy production}} = \frac{CAPEX \cdot f_{\text{annuity}} + OPEX}{E_a} \\ &= \frac{CAPEX \cdot f_{\text{annuity}}}{E_a} + OM \end{aligned} \quad (5.20)$$

with the annuity factor

$$f_{\text{annuity}} = \frac{(1+i)^{N_y} \cdot i}{(1+i)^{N_y} - 1}. \quad (5.21)$$

Net present value In Equation (5.22) from Augsburg [4] the computation for the total profit, e.g. the net present value (NPV) is given with the annuity factor as defined in Equation (5.21). The computation sums up all incomes and subtracts all expenses over the whole project life time.

$$\begin{aligned} NPV &= \frac{(1+i)^{N_y} - 1}{i \cdot (1+i)^{N_y}} (ToE - OM) \cdot E_a - CAPEX \\ &= \frac{ToE - OM}{f_{\text{annuity}}} \cdot E_a - CAPEX \end{aligned} \quad (5.22)$$

Internal rate of return The internal rate of return (IRR) is the interest rate of Equation (5.22) where $NPV = 0$.

Payback period For plant operators the payback period is of high interest. It gives the time in years after which the plant is starting to make profit. In Equation (5.23) from Augsburg [4] the symbol ToE describes the price paid per kWh_{el} on the electricity market.

$$N_{y,\text{payback}} = \frac{\log\left(\frac{(ToE-OM)\cdot E_a}{(ToE-OM)\cdot E_a - CAPEX\cdot i}\right)}{\log(1+i)} \quad (5.23)$$

6. Settings

In this section each individual setting is presented. The settings are grouped into JSON files. In the following each JSON file is listed.

6.1. Site settings

In Table 10 the parameters of the JSON file, specifying the settings for the site, are shown. The root node of the file has to be called "site".

Parameter name	Unit	Data Type	Range
latitude	degree	double	[-90, 90]
longitude	degree	double	[-180, 180]
timezoneOctMar	-	double	[-12, 12]
timezoneMarOct	-	double	[-12, 12]
minimal distance between objects	meter	double	[0, 100]
coordinates	-	enum	{Cartesian, Geo(lat & long)}
fieldboundaries			
-> points	array of points in Cartesian or Geo	array[[double, double]]	array of size 3 or greater
restricted_areas			
-> points	array of points in Cartesian or Geo	array[[double, double]]	array of size 3 or greater

Table 10: JSON file for the site as list of its parameters.

6.2. Meteorological information

In Table 11 the parameters of the JSON file, specifying the settings for the meteorological information, are shown. The root node of the file has to be called "meteodata".

6.3. Heliostat settings

In Table 12 the parameters of the JSON file, specifying the settings for the heliostats, are shown. The root node of the file has to be called "heliostat".

Parameter name	Unit	Data Type	Range
azimuth	degree	double	[0, 360]
altitude	degree	double	[0, 90]
irradiation	W/m^2	double	[0, 1400]
dew_point_temperature	$^{\circ}C$	double	[-40, 50]
environment_temperature	$^{\circ}C$	double	[-50, 60]
environment_pressure	Pa	double	[8000, 12000]
wind_speed	m/s	double	[0, 110]
day	-	int	[0, 365]
time	h	double	[0,24]
duration	h	double	[0, 24]
duration_multiplicity	$days$	int	[0, 365]

Table 11: JSON file for the sun as list of its parameters.

6.4. Tower settings

In Table 13 the parameters of the JSON file, specifying the settings for the tower, are shown. The root node of the file has to be called "tower".

6.5. Receiver settings

In Table 14 the parameters of the JSON file, specifying the settings for the receiver, are shown. The root node of the file has to be called "receiver".

6.5.1. Ray-tracer settings

In Table 15 the parameters of the JSON file, specifying the settings for the ray-tracer, are shown. The root node of the file has to be called "ray-tracer".

6.6. Thermal model settings

In Table 16 the parameters of the JSON file, specifying the settings for the thermal model, are shown. The root node of the file has to be called "thermal_receiver". Table 17 shows the different materials to choose from.

6.7. Electrical model settings

In Table 18 the parameters of the JSON file, specifying the settings for the power block, are shown. The root node of the file has to be called "powerblock".

6.8. Economic model settings

In Table 18 the parameters of the JSON file, specifying the settings for the economic model, are shown. The root node of the file has to be called "economic".

Parameter name	Unit	Data Type	Range
heliostat_type_name	text	std::string	
facet_surface_form	-	enum	{flat, focused}
facet_surface_form_focused			
-> ideal_focal_length	boolean	bool	true; false
-> focal_length	meter	double []	[0, 2000]
canting	-	enum	{none, on_axis, off_axis_time, off_axis_angle, manual}
off_axis_canting			
-> reference	-	enum	{datetime, azimuthaltitude}
-> datetime	day x hour	<int,double>	[1, 365]x[0, 24]
-> azimuthaltitude	degree x degree	double [2]	[0, 360] ²
reflectivity	norm factor	double	[0, 1]
pedestal_height	meter	double	[0, 30]
tracking_method	-	enum	{azimuth_elevation}
tracking_offset	meter	double	[0, 1]
facet_definition	-	enum	{single_facets, simple_rectangular}
facets(single)	list	std::list	
-> facet_position	meter x meter	double [2]	[-50, 50] ²
-> facet_shape	-	enum	{rectangular, triangular}
-> facet_dimension (rectangle)	meter x meter	double [2]	[0, 40] ²
-> facet_dimension (triangle)	meter x degree	double [2]	[0, 40]x[0, 180]
-> facet_rotation	degree	double	[0, 360]
-> facet_normal	direction vector	double [2]	
facets(general)			
-> number_facets	count x count	int [2]	[0, 100] ²
-> facet_gap	meter x meter	double [2]	[0, 1] ²
-> facet_size	meter x meter	double [2]	[0, 30] ²

Table 12: JSON file for the heliostats as list of their parameters.

Parameter name	Unit	Data Type	Range
type	-	enum	{rectangular, cylindrical}
height	meter	double	[0, 300]
orientation angle	degree	double	[-180, 180]
position	meter	double[2]	[-2000,2000] ²
tower rectangular			
-> size	meter	double[2]	(0, 100]
tower cylindrical			
-> diameter	meter	double	(0,100]

Table 13: JSON file for the tower as list of its parameters.

Parameter name	Unit	Data Type	Range
type	-	enum	{flat cavity, cylindrical cavity, cylindrical external}
height	meter	double	(0, 100]
distance to tovertop	meter	double	[0, 100]
num cells	amount	int[2]	[1, 1024] ²
flat cavity			
-> width	meter	double	(0, 100]
-> tilt angle	degree	double	[0, 90]
cylindrical cavity			
-> arc length	meter	double	(0, 200]
cylindrical external			
-> radius	meter	double	(0, 75]

Table 14: JSON file for the receiver as list of its parameters.

Parameter name	Unit	Data Type	Range
ray tracing method	-	enum	{MC, mMC, convolution, cell-wise_convolution}
monte carlo ray multiplier	factor	int	[1, 100]
num rays per facet width	amount	int	[1, 128]
num rays per facet height	amount	int	[1, 128]
sigma sun	mrاد	double	[0, 1000]
sigma tracking	mrاد	double	[0, 1000]
sigma slope	mrاد	double	[0, 1000]

Table 15: JSON file for the ray-tracer as list of its parameters.

Parameter name	Unit	Data Type	Range
wall material	-	enum	material
htf material	-	enum	material
hot salt temperature	°C	double	[200, 2000]
cold salt temperature	°C	double	[200, 2000]
receiver diameter	meter	double	(0, 150]
receiver height	meter	double	(0, 100]
tube diameter	meter	double	(0, 0.5]
tube wall thickness	meter	double	(0, 0.1]
flow type	number	int	[1, 8]
number of panels	amount	int	[1, 100]
number of cells per panel	amount	int	[1, 100]
pump efficiency	norm factor	double	[0, 1]

Table 16: JSON file for the thermal model as list of its parameters.

Parameter	Value
air	air
stainless steel	stainless steel
salt1	68% KCl, 32% MgCl ₂
salt2	8% NaF, 92% NaBF ₄
salt3	25% KF, 75% KBF ₄
salt4	31% RbF, 69% RbBF ₄
salt5	46.5% LiF, 11.5%NaF, 42%KF
salt6	49% LiF, 29% NaF, 29% ZrF ₄
salt7	58% KF, 42% ZrF ₄
salt8	58% LiCl, 42% RbCl
salt9	58% NaCl, 42% MgCl ₂
salt10	59.5% LiCl, 40.5% KCl
salt11	59.5% NaF, 40.5% ZrF ₄
salt12	60% NaNO ₃ , 40% KNO ₃

Table 17: List of Materials.

Parameter name	Unit	Data Type	Range
max thermal power	MW	double	(0, 1000]
temperatures	°C	double []	[-100, 200]
loads	percent	double []	[0, 100]
efficiencies	percent	double [] []	[0, 100]

Table 18: JSON file for the power block as list of its parameters.

Parameter name	Unit	Data Type	Range
interest rate	-	double	(0, 2]
number staff	amount	int	[0, 100]
plant lifetime	years	int	[1, 200]
storage capacity	MW_{th}	double	[0, 1000]
tariff of electricity	$\text{¢}/(\text{kWh}_{el})$	double	[0, 100]

Table 19: JSON file for the economic model as list of its parameters.

Part II.

Feasibility tests

In this part methods to reduce the computation time of the simulation are introduced. In the following we will validate the optical model and find a good ray resolution. Then we will discuss the bitboard resolution and preselection during shading and blocking calculation. Furthermore we will take a look at the parallelization of the hierarchical ray tracing method.

7. Validation of optical model

To validate the optical model of **SunFlower** first a cross validation with the tool SolTrace [65] is done. Then the accuracy is opposed to the number of sun rays evaluated by both tools when simulating an existing power plant. At last the new developed analytical ray tracing methods in **SunFlower** are validated against the own Monte-Carlo approach.

7.1. Cross validation against SolTrace

To validate the results of **SunFlower** we compare the outcomes of test cases with those of the Monte-Carlo ray-tracer SolTrace [65] by running each tool ten times. Each test case represents a setup of a solar tower power plant with a unique combination of settings. The setups therefore are designed to test the majority of setting combinations when modeling solar tower power plants. We have defined a general setup in Table 20 which utilizes the heliostat Sanlúcar 120. The Sanlúcar 120 is used in the PS10 plant and consists of 7 rows by 4 columns of facets summing up to roughly 120 m² as stated by [41]. A picture of the Sanlúcar 120 can be found in Figure 7.1. Since SolTrace can not consider tower shading we disable the tower shading feature of **SunFlower**.

For each of the following test cases the accumulated optical power at the receiver is computed for both tools and normalized by the average results of SolTrace which are defined as reference. The original output values for the optical power can be found in Table 24.

7.1.1. One heliostat

We designed multiple test cases for different heliostats. Additionally to the accumulated optical energy we also compute the flux at the receiver. The individual heliostats are assumed to be positioned at (0,100) in a distance of 100 meters to the base of the tower. The different settings combinations can be found in Table 21.

The comparison of the average results can be found in Figure 7.2a. Except for test case 1 the results of **SunFlower** are between the minimum and maximum of the SolTrace results. As additional measurement we calculated the gap. This is defined as the difference between the minimum and maximum result in ten runs of the respective

Parameter	Value
Latitude	37.442400
Longitude	-6.250188
Direct Normal Irradiance	800 W m ⁻²
Sun azimuth	180°
Sun altitude	70°
Sun Error	2.35 mrad
Slope Error	0 mrad
Tracking Error	0 mrad
Heliostat	Sanlúcar 120
Heliostat Reflectivity	88%
Rays per Heliostat	168000
Ray tracing Method for SunFlower	Monte-Carlo
Tower height	115 m
Tower type	Rectangular Tower
Tower length	18 m
Tower width	8 m
Tower position	(0,0)
Receiver type	Flat Cavity Receiver
Receiver height	12 m
Receiver width	13.78 m
Receiver Tilt Angle	11.5°
Receiver distance to Towertop	2.74 m

Table 20: Basic Setup for the validation test cases. The settings are inspired by the PS10 plant.

tool. The gap between the maximum and minimum of SunFlower averages at 2.675 Watts, while the gap in SolTrace averages at 486.483 Watts.

7.1.2. Blocking and shading

In this test case we place two heliostats. For the blocking cases the first heliostat is positioned at (0,740) while the second is positioned at (0,750). To create a blocking effect the sun is at 180 °azimuth and 80 °altitude. For the shading cases the first heliostat is positioned at (0,80) while the second is positioned at (0,100). To create a shadow the sun is at 180 °azimuth and 0 °altitude. The settings can be found in Table 22.

The comparison of the results can be found in Figure 7.2b. Except for test cases 7 and 8, the results of SunFlower are between the minimum and maximum of the SolTrace results. The gap between the maximum and minimum of SunFlower averages at 119.3 Watts for the blocking test cases (7-12) and 43.3 Watts for the shading test cases (13-18) while the gap in SolTrace averages at 753.8 and 519.2 Watts respectively.



Figure 7.1: Frontal view of the Sanlúcar 120 [41].

Test case	Canting			Focused facets	
	None	On-axis	Off-axis	Flat	Focused
1	✓			✓	
2	✓				✓
3		✓		✓	
4		✓			✓
5			✓	✓	
6			✓		✓

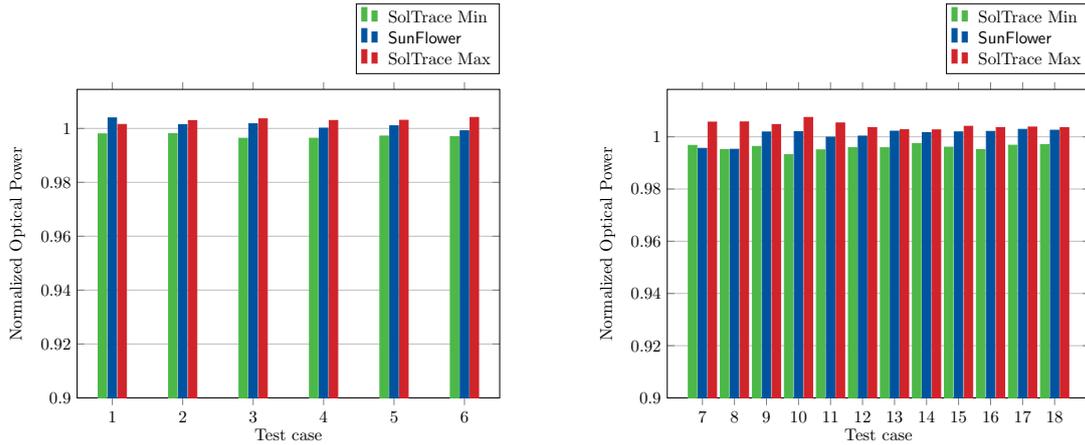
Table 21: Unique settings combinations for each test case with one heliostat.

7.1.3. Complete solar tower power plant

In the following scenes the complete PS10 is modelled with different settings. The heliostat positioning can be seen in Figure 7.3. The combination of the settings can be found in Table 22.

Since SolTrace can not handle the amount of facets present in the PS10 solar tower power plant. We abstracted the Sanlúcar 120 with a heliostat having the same mirror area but only containing one facet. This results in the canting test cases that the facets do not get canted. Therefore in Table 24 we have the test cases 19' and 20' to compare the results of SunFlower with the results of SolTrace and the regular test cases to show the results achieved with SunFlower on the PS10 solar tower power plant.

The results of SunFlower have an average gap of 57 811.9 Watts for test cases 19' and 20', whilst SolTrace has an average gap of 66 050 Watts. The average gap of



(a) Results of test cases where one heliostat is placed. (b) Results of test cases where two heliostats are placed. An alternating sun position ensures shading and blocking respectively.

Figure 7.2: Results of Monte-Carlo ray tracing validation on different test cases.

SunFlower for test cases 19 to 24 is 21 318.6 Watts.

7.2. Accuracy vs number of rays

When simulating solar tower power plants with Monte-Carlo ray tracing methods, it is imperative that enough rays are traced. This is due to the law of large numbers which Monte-Carlo methods use as their foundation. Therefore it is of interest to investigate how many rays have to be traced to get a certain accuracy regarding the evaluation of optical energy collected at the receiver.

We compare the outcome for the PS10 solar tower power plant for different number of traced rays. For the comparison the complete PS10 power plant with focused facets and off-axis canting on the Sanlúcar 120 is used. We run the setup for 1, 4, 16, 64, 256, 1 024, 4 096 and 16 384 rays per facet, this results in a overall number of 17 472, 69 888, 279 552, 1 115 208, 4 472 832, 17 891 328, 71 565 312 and 286 261 248 rays respectively. The outcome is then normalized by the outcome when we shoot 286 261 248 rays. Each number of rays gets traced for 16 different points a year to account for multiple sun positions. A plot of the influence of the number of rays on the accuracy can be found in Figure 7.4. For a total of 17 472 we have an error of less than 0.2%, which is less than the average deviation to SolTrace from the previous test cases. When reaching 1 115 208 rays the error is less than 0.005%.

7.3. Validation of ray tracing methods

To validate our new analytical ray tracing methods, we compare the outcome of these for the same test cases as in Section 7.1 against our Monte-Carlo method. For the test cases 1 to 18 we take as many rays as in Section 7.1. For test cases 19 to 24 the number

Test case	Sun position		Canting			Focused facets	
	Blocking	Shading	None	On-axis	Off-axis	Flat	Focused
7	(180°, 80°)		✓			✓	
8	(180°, 80°)		✓				✓
9	(180°, 80°)			✓		✓	
10	(180°, 80°)			✓			✓
11	(180°, 80°)				✓	✓	
12	(180°, 80°)				✓		✓
13		(180°, 0°)	✓			✓	
14		(180°, 0°)	✓				✓
15		(180°, 0°)		✓		✓	
16		(180°, 0°)		✓			✓
17		(180°, 0°)			✓	✓	
18		(180°, 0°)			✓		✓

Table 22: Unique settings combinations for each test case with blocking and shading effects.

Test case	Canting			Focused facets	
	None	On-axis	Off-axis	Flat	Focused
19	✓			✓	
20	✓				✓
21		✓		✓	
22		✓			✓
23			✓	✓	
24			✓		✓

Table 23: Unique settings combinations for each test case with PS10 plant heliostat layout.

of rays is the 99.8% point of **SunFlower** as evaluated in Section 7.2 of 1 ray per facet. The convolution method is averaging below 1% error compared to the result by the Monte Carlo method, whilst getting an error of 2% in test cases 19 and 21. The cell-wise convolution method manages to get accurate results in test cases with no canting. When simulating the whole PS10 power plant it has an error of 5% with flat facets and 7% with focused facets. For test cases with canting, the cell-wise convolution method does not produce reliable results. The results are pictured in Figure 7.5.

7.4. Conclusion

In this section we measured the results given by state of the art Monte-Carlo ray-tracer SolTrace and our tool **SunFlower** for 20 test cases. Each test case was run on each tool ten times. Comparing the results of **SunFlower** to those of SolTrace there are only four test cases where the results do not lie in-between the minimum and maximum of

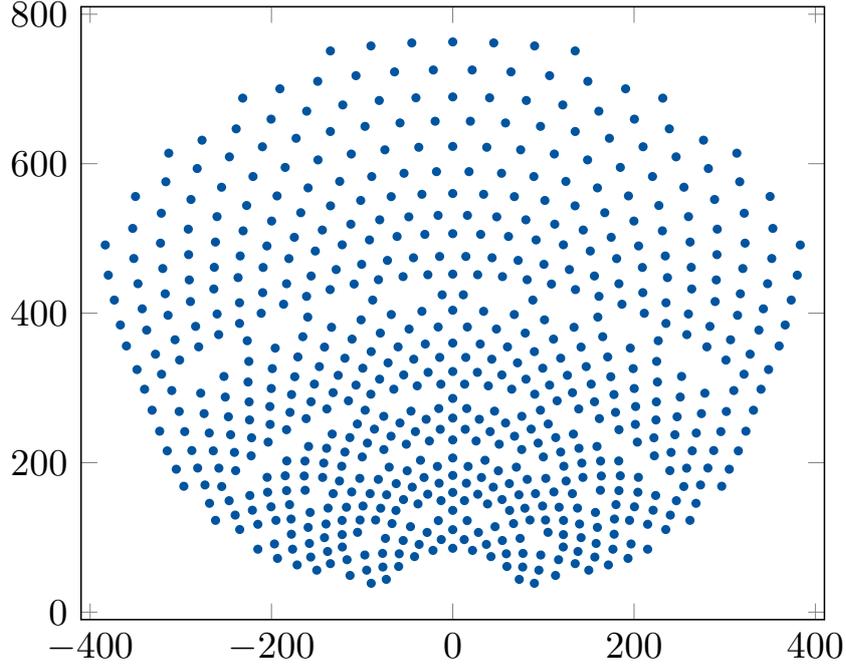


Figure 7.3: PS10 heliostat field layout.

SolTrace, but they are also near the results of SolTrace. Therefore we can conclude that our tool produces reliable results.

Furthermore we showed that we even can reduce the resolution of rays on our tool. We can recommend a resolution of $0.2 \frac{\text{rays}}{\text{m}^2}$ with an error of less than 0.05% when optimizing whole plant layouts. Whereas we recommend $945 \frac{\text{rays}}{\text{m}^2}$ with an error of less than 0.001% when more accurate results are needed or not whole plant layouts are simulated.

The validation of our analytical flux calculations showed that our convolution method produces an error of less than 1% on average. Therefore we consider the convolution method reliable for flux calculations. On the other hand the cell-wise convolution has problems calculating the flux for canted heliostats. Furthermore it produces errors of 5% and 7% when simulating flat and focused facet heliostats in the PS10 layout. Therefore at the current stage of development we can not recommend the cell-wise convolution.

Test case	SunFlower			SolTrace		
	min	avg	max	min	avg	max
1	82 805.3	82 838.3	82 855.6	82 349.5	82 517.1	82 639.2
2	83 602.3	83 613.6	83 623.2	83 334.6	83 498.9	83 740.7
3	81 528.7	81 582.2	81 618.0	81 145.4	81 442.6	81 731.1
4	83 141.1	83 141.1	83 141.1	82 830.3	83 133.1	83 378.0
5	83 674.8	83 674.8	83 674.8	83 356.8	83 593.8	83 846.7
6	83 674.2	83 674.2	83 674.2	83 489.4	83 747.6	84 089.2
7	73 159.0	73 219.9	73 281.5	73 304.0	73 545.7	73 958.9
8	74 447.0	74 551.7	74 620.0	74 541.9	74 909.7	75 335.3
9	70 813.0	70 890.9	70 968.3	70 500.1	70 761.3	71 093.9
10	71 323.9	71 394.3	71 497.4	70 766.9	71 256.0	71 784.5
11	81 179.1	81 203.1	81 228.0	80 807.8	81 215.2	81 644.5
12	81 156.4	81 178.4	81 199.1	80 818.5	81 156.7	81 444.7
13	78 091.5	78 123.6	78 150.5	77 628.9	77 956.0	78 165.7
14	78 876.7	78 882.9	78 891.3	78 555.1	78 758.1	78 965.5
15	67 191.4	67 245.8	67 286.5	66 853.5	67 120.5	67 390.1
16	67 271.1	67 313.1	67 361.9	66 844.4	67 174.4	67 412.7
17	79 177.3	79 177.3	79 177.3	78 697.7	78 952.6	79 246.7
18	79 176.3	79 176.3	79 176.4	78 738.6	78 977.7	79 252.7
19'	40 390 396.0	40 444 354.7	40 499 494.0	40 144 700	40 178 400	40 211 500
19	40 146 491.3	40 155 660.2	40 163 434.6	/	/	/
20'	46 874 261.2	46 876 703.6	46 880 786.9	46 838 800	46 872 300	46 904 100
20	41 086 482.7	41 096 312.3	41 113 109.6	/	/	/
21	42 083 186.2	42 096 105.4	42 130 365.7	/	/	/
22	42 858 804.9	42 872 978.9	42 890 698.9	/	/	/
23	46 814 649.7	46 815 879.3	46 817 764.8	/	/	/
24	46 855 424.5	46 856 298.3	46 857 577.5	/	/	/

Table 24: Exact results for each tool for all test cases.

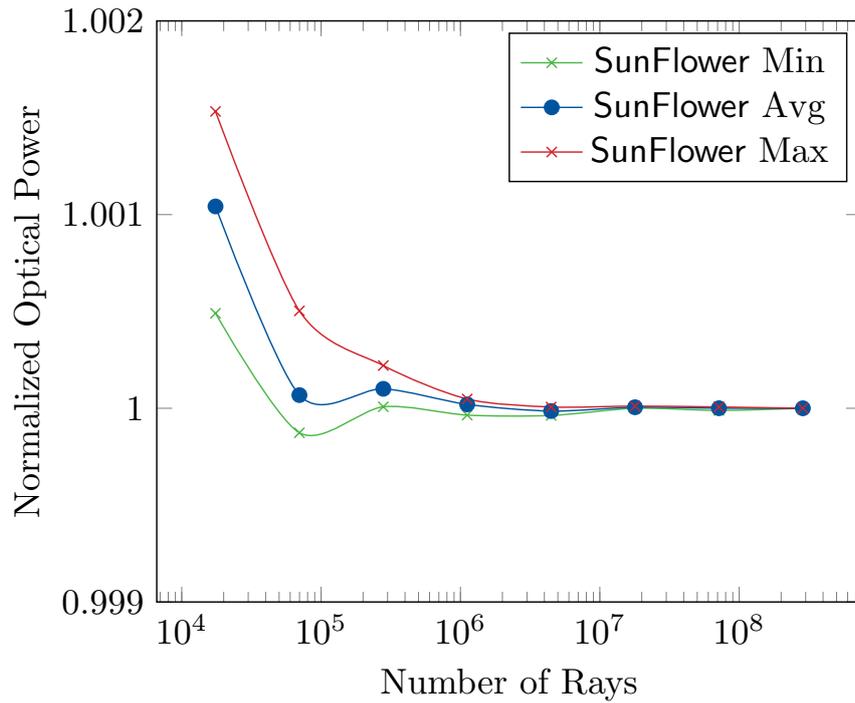


Figure 7.4: The impact of the number of rays on the accuracy. Each point represents the average normalized optical power over multiple sun positions.

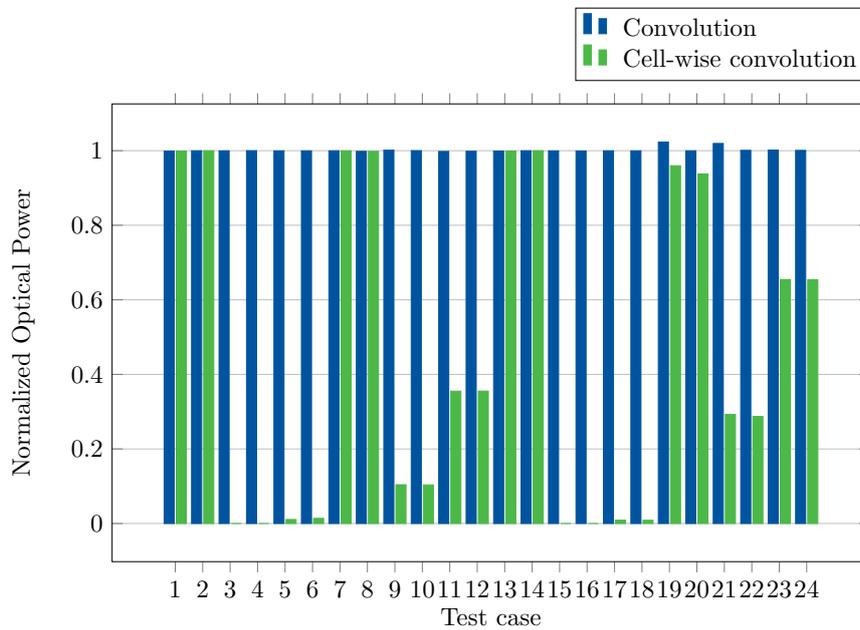


Figure 7.5: Results of analytical flux computation on validation test cases.

8. Bitboard resolution and preselection

In this section an approach to reduce the computationally expensive task of the ray tracing is presented. In order to achieve this a subset of heliostats is used when calculating shading and blocking effects in contrast to all heliostats. The subset contains potentially shading or blocking heliostats which then need to be checked with ray tracing. To determine this subset, a data structure is needed, which is fast in nearest-neighbour search and in range-search[47].

For the calculation of the subset a two-dimensional index structure, called bitboard, is used. The index structure is designed to map onto an equidistant grid, such that the two-dimensional x - y space is covered. The grid cells then store the information if they contain a part of a heliostat, see Figure 8.1.

To calculate the subset of potential shading or blocking heliostats we distinct in a two-dimensional preselection, a three-dimensional preselection as well as no preselection at all. The last case results in all rays being traced against all heliostats to see if they are shaded or blocked. The two-dimensional preselection works only on the bitboard, considering each heliostat on the path of the ray to be potentially shading or blocking. The rays get widened to build a channel with width equal to the heliostats expansion, evaluating all cells which are in this channel. The three-dimensional preselection checks whether the distance of the ray to the heliostat is smaller than the expansion of the heliostat plus the expansion of the represented area of the ray, which could lead to a shaded or blocked ray.

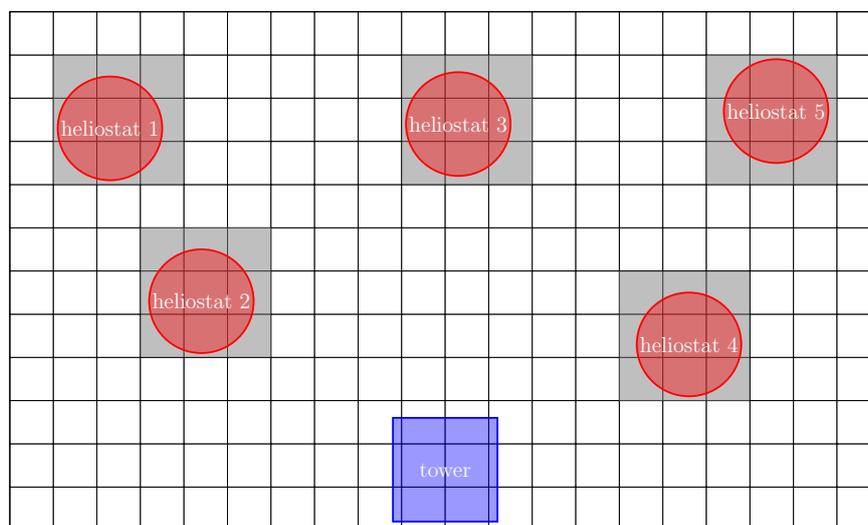


Figure 8.1: Bitboard index structure: Each cell stores the information whether part of a heliostat is in this cell. [47]

A study was made over simulating 16 moments on different resolutions of the bitboard, e.g. on different cell sizes. In Figure 8.2 the average run time on all 16 moments is depicted for different bitboard resolutions. The fluctuations without any preselection

are caused by the initialization of the bitboard. The 2D preselection offers a lower run time peak whilst the 2D and 3D preselection together increase less with bigger cells, e.g more heliostats which may be flagged for potential shading and blocking. Overall just the 3D preselection takes 17 times longer than the other combinations while no preselection leads to the lowest run times.

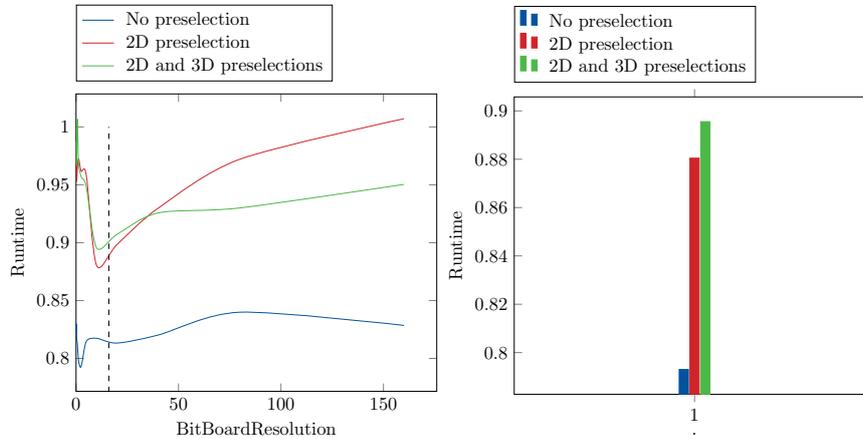


Figure 8.2: The runtime of SunFlower over the resolution of the BitBoard.

8.1. Only preselection

Another approach to speed up the shading and blocking calculations is to rely on the preselection without tracing any ray at all to calculate the shading and blocking effects.

The requirement for this to work is that the number of blocked or shaded rays is equal to the number of potentially blocked or shaded rays. When using both preselections 2D and 3D preselection we made a study which should give an estimation on how many preselected rays finally get shaded or blocked. The study showed that only 10% of the potentially shaded or blocked rays are finally shaded or blocked. Therefore this method would create an enormous error on the outcome and is not suitable for speeding up the simulation process.

9. Parallelization in hierarchical ray tracing

As stated in 1.6 we use a hierarchical ray tracing approach. This can be parallelized in two different ways, parallelization of the single stages during the ray tracing or the whole ray tracing. Therefore the first approach was to parallelize the creation of rays, the shading calculation, the blocking calculation and the tracing against the receiver mesh. Whilst the second was a parallelization of the whole process.

We assumed that the second approach should lead to faster run times since we have a better data locality in the cache of the cores. When studying the run time of these

two different approaches, we noticed that the first approach led to higher run times than the second, see Figure 9.1.

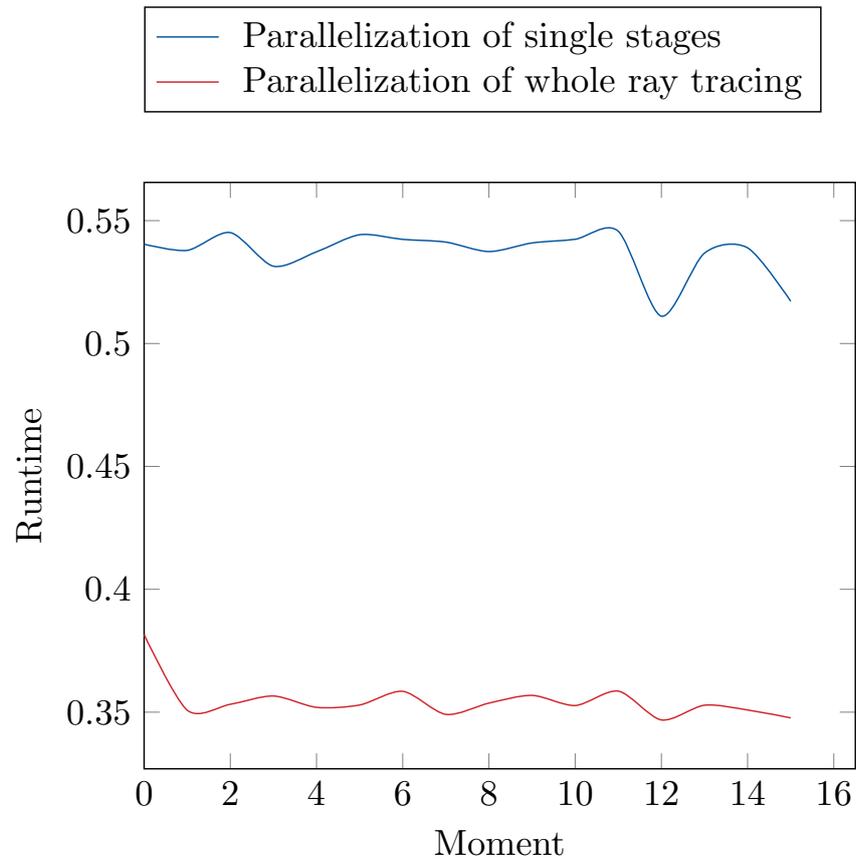


Figure 9.1: The runtime of SunFlower when parallelizing single stages of the ray tracing or the whole process.

Thus, it can be said that the second approach is the better one and is used up from now on in SunFlower .

Part III.

Optimization

The annual energy production of a solar tower power plant underlies multiple effects as presented in Sections 1 to 5. In order to minimize losses during the energy production we optimize the positioning of the heliostats. The research results of Richter et. al. [47, 48, 49] are the foundation of the development of this part.

Heliostat field layout optimization is a continues optimization problem. This introduces complexity to the already given restrictions on positioning the heliostats as presented in Section 1.1. In the following each algorithm has to cope with the complexity.

With our findings from Part II we use $0.2 \frac{\text{rays}}{\text{m}^2}$ and the Monte-Carlo ray tracing method. Furthermore we do not use a pre-selection for the shading and blocking calculations.

In the following the optimization problem is described. Furthermore each used algorithm is described and a new multi-step optimizer is presented. Finally we optimize a large-scale solar tower power plant.

10. Algorithms

For the optimization of the heliostat field layout we derived several algorithms from existing research as well as new ones. In the following several patterns, which mark the state of the art approach for designing a heliostat field, will be explained. Further a genetic algorithm and a local search heuristic are presented.

10.1. Patterns

Patterns are the state of the art approach for building solar tower power plants. To our knowledge, all commercial solar tower power plants use patterns to position the heliostats. Furthermore state of the art research discusses new patterns which shall improve the outcome of a plant. In [57] Schramek et. al. claim that the ground coverage is more important than the actual pattern. This will lead to a closer packing of the heliostats on the field.

In general, patterns can be described in terms of their free variables, which define the actual positioning of the heliostats. The positioning of the heliostats can then be optimized by setting the free variables with a modern solver like the simplex or a simple method which discretizes the range of the free variables and tests all combinations to find the best suitable one. We have chosen the *Downhill Simplex* method from [37]. In the following some of these pattern approaches will be discussed.

10.1.1. North-South Cornfield

The north-south cornfield [57] pattern places the heliostats on straight lines. Where every second row is staggered to achieve a closer packing. This pattern is defined by the row and column distances (see Figure 10.1). To achieve the staggering the distances can be parametrized by:

$$d_i^{\text{row}} = a_{\text{row}} \cdot r_i + b_{\text{row}}, \quad (10.1)$$

$$d_i^{\text{col}} = a_{\text{col}} \cdot c_i + b_{\text{col}}. \quad (10.2)$$

Where r_i denotes the y -value of row i and c_i denotes the x -value of column i . Therefore yielding the following free variables for optimization:

a_{row}	linear factor for the row distance
b_{row}	constant factor for the row distance
a_{col}	linear factor for the column distance
b_{col}	constant factor for the column distance

10.1.2. Radial staggered

The radial staggered pattern is based on the intuitive idea of arranging the heliostats in rings around the tower, such that the heliostats in every second ring are staggered in order to achieve a closer packing. This pattern is currently used in various commercial solar tower power plants, see for example the Khi One in Figure 10.2. An advantage of this pattern is, that it naturally includes paths for cleaning trucks. This should not be underestimated in practice [42].

Figure 10.3 shows schematically how the pattern is constructed. The heliostats are placed in rings around the tower in a staggered manner. Several rings are grouped into zones. The zones are enumerated from the inside to the outside by Z_i and the rings within zone Z_i by $R_{i,j}$. The rings within one zone have the same number of heliostats, which is determined by the requested distance d^{sep} of the innermost ring of each zone. The distance between two zones is larger than the distance between rings, since at this point the heliostats are not properly staggered.

The pattern offers several degrees of freedom:

- when to start a new zone
- distance in-between rings and zones

The number of heliostats per ring can be computed by:

$$N_i = \left\lfloor \frac{2\pi}{2 \cdot \text{asin}\left(\frac{d^{\text{sep}}}{2r_{i,1}}\right)} \right\rfloor. \quad (10.3)$$

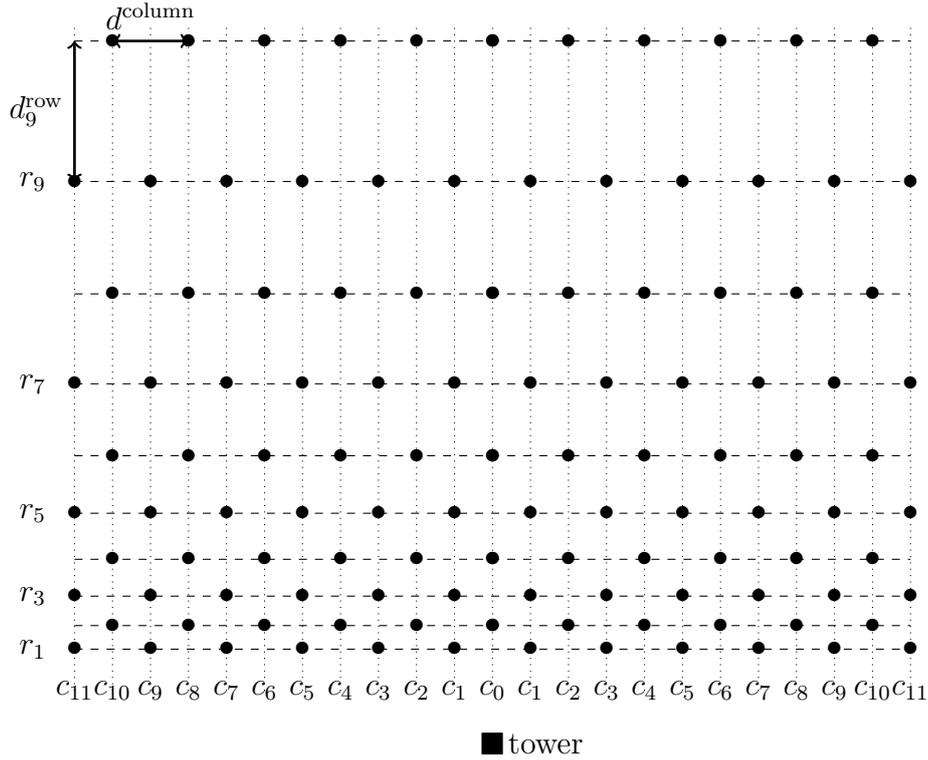


Figure 10.1: This sketch illustrates the general structure of the north-south cornfield layout. Heliostats are placed on rows in staggered columns. The rows are enumerated from the front to the back, the columns symmetrically from the center to the outside.

We reduce the dimension of the corresponding search space as follows: For the starting of a new zone we merely try two different heuristics, resulting in two distinct patterns. The first option is to start a new zone whenever the number of potential heliostats per ring doubles. This heuristic was proposed by Collado et. al. [13]. The second option compares the number of heliostats per space used, in the case a new ring in the previous zone is added and when a new zone is started. It then picks the better of the two values. Formally, we compute Equation (10.4), where N_1 is the number of heliostats per ring in the current zone and N_2 the number in a hypothetical new zone. This heuristic was proposed by Siala and Elayeb [59].

$$\frac{N_2}{R_2^2\pi - R_0^2\pi} \geq \frac{N_1}{R_1^2\pi - R_0^2\pi} \quad (10.4)$$

Furthermore, we do not optimize each ring distance individually, but parameterize it as a linear function of the current radius, i.e. $d_{i,j}^{\text{Ring}} = a_{\text{ring}} \cdot r_{i,j} + b_{\text{ring}}$. The distance between zones is directly coupled to the ring distance. Lastly, the distance between heliostats on a ring is also modeled as a linear term of the current radius, i.e. $d_i^{\text{sep}} = a_{\text{sep}} \cdot r_{i,1} + b_{\text{sep}}$. This linear parametrization enables us to generate a field which is



Figure 10.2: This is a picture from the khi solar power plant in South Africa [21].

denser in the center and less dense on the outside, while still having a manageable number of free variables. This results in the following variables for optimization:

a_{ring}	linear factor for ring distance
b_{ring}	constant factor for ring distance
a_{sep}	linear factor for heliostat distance
b_{sep}	constant factor for heliostat distance

10.1.3. Hexagon

The radial staggered pattern introduced in the previous section has several shortcomings. It wastes some space due to the fact that the number of heliostats per ring is constant within one zone while the rings continuously increase. It would be desirable to also increase the number of heliostats continuously. Additionally, radial patterns are more complex to build than straight ones and they are more difficult to maintain, due to the bended paths which the cleaning truck has to use. This all adds up to the costs of the plant, which makes it less profitable [42]. A possible improvement is to approximate the rings by hexagons. This has been suggested in patent by Pham et. al. [42] and the resulting hexagon pattern is also mentioned in other publications [3].

Figure 10.4 shows the structure of this pattern. The heliostats are placed on concentric hexagons H_i around the tower. Each hexagon has one more heliostat per edge than the previous one, resulting in $N_i = N_0 + 6i$ heliostats on H_i , were N_0 is the number of heliostats on the first ring. The heliostats are distributed regularly on the sides of the hexagon, such that the distance from the corner to the first heliostat is half the distance between two heliostats. This results automatically in a staggered layout with no two heliostats in adjacent rows forming a line with the tower. Note, that the side of each hexagon equals the distance of its corners to the tower, and the ratio between this

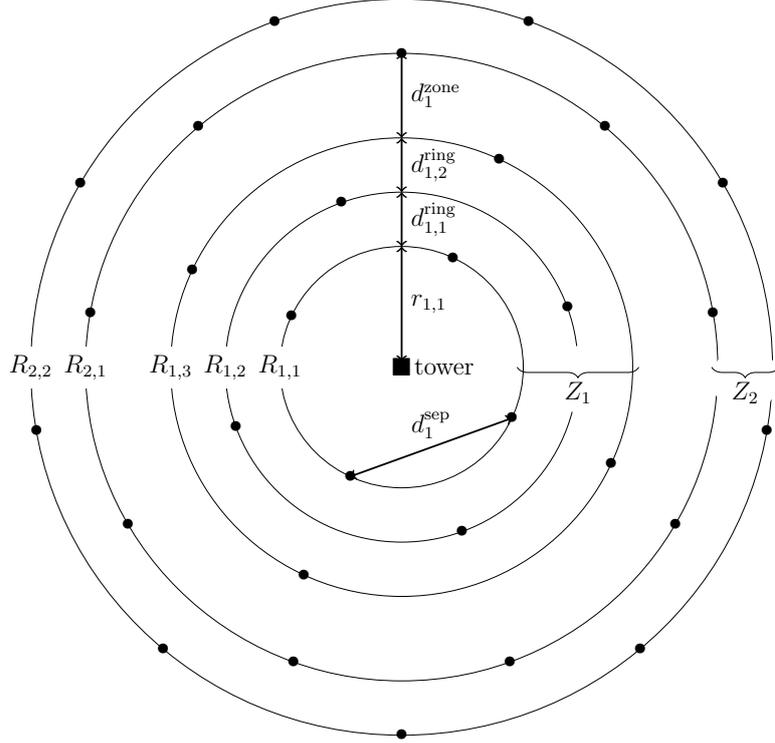


Figure 10.3: This sketch illustrates the general structure of the radial staggered layout. The heliostats are arranged on rings $R_{1,1} - R_{2,2}$ around the tower. The radius of a ring $R_{i,j}$ is denoted by $r_{i,j}$. The distance between rings $R_{i,j}$ and $R_{i,j+1}$ is denoted by $d_{i,j}^{\text{Ring}}$. The rings are subdivided in zones Z_1 and Z_2 . The distance between these zones is denoted by d_1^{Zone} . The distance between two heliostats on the same ring is d_i^{sep} and corresponds directly to the number of heliostats placed on this ring.

distance and the radius r_i of a hexagon is $\frac{k_i}{r_i} = \frac{2}{\sqrt{3}}$. From this we can compute, that the distance between heliostats on ring H_i is $c_i = \frac{2r_i}{\sqrt{3}(N_0+i)}$. Furthermore, the distance d^{sep} between heliostats on adjacent hexagons can be calculated as $d^{\text{sep}} = \sqrt{(\frac{1}{2}c_j)^2 + d_i^2}$ where j is either i or $i + 1$.

The free variables in this pattern are the number N_0 of heliostats on the first hexagon and the distances d_i between the individual hexagons.

In our implementation, we start with a small hexagon with only one heliostat per edge. This hexagon might be within the circle around the tower which is blocked by the tower, the powerblock and other buildings. Therefore it may be removed by the post processing working on these constraints. The distances between hexagons are parametrized analogously to the radial staggered setting by $d_i = a \cdot r_i + b$. This yields the following variables for optimization:

- a linear factor for the hexagon distance
- b constant factor for the hexagon distance

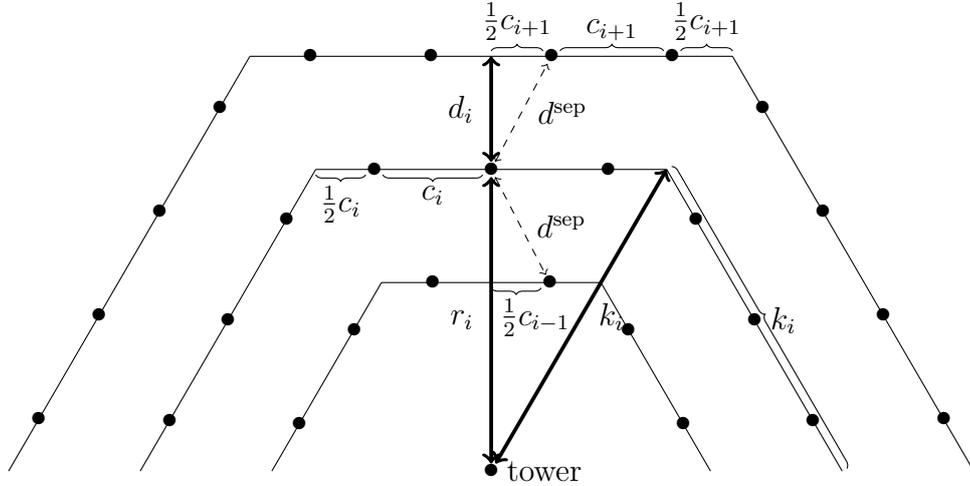


Figure 10.4: This sketch shows the general structure of the hexagon pattern. All heliostats are placed on the edges of the concentric hexagons. We define r_i by the distance from the center to the midpoints of its edges and k_i by the distance to its corners. Since hexagons are composed of regular triangles, its edges also have length k_i . We denote the distance between H_i and H_{i+1} measured at the midpoints of the edges by d_i . The distance between two heliostats on the hexagon H_i is c_i and the distance of a heliostat on H_i to the next corner is $\frac{1}{2}c_i$. The distance to a heliostat on the next hexagon is d_{sep} (which is the distance of the heliostats closest to the center of the edge).

10.1.4. Spiral

A disadvantage of all previous patterns is, that their regularities still waste space while at the same time not avoiding blocking and shadowing. By studying sunflowers, it turned out that their spiral arrangement is much better in this task. Therefore, Noone et. al. [39] suggest transferring this pattern to the heliostat field and reports good results.

In this pattern, the position of each heliostats is determined by its angle and radius. For the angle of the k -th heliostat we use $\alpha_k = 2\pi\varphi^{-2}k$ where $\varphi = \frac{1+\sqrt{5}}{2}$ is the golden ratio. And for its radius we use $r_k = ak^b$. The resulting spiral as shown in Figure 10.5 is called the golden spiral.

This pattern has naturally two free parameters for optimization:

- a linear factor ("zoom")
- b exponential factor ("density")

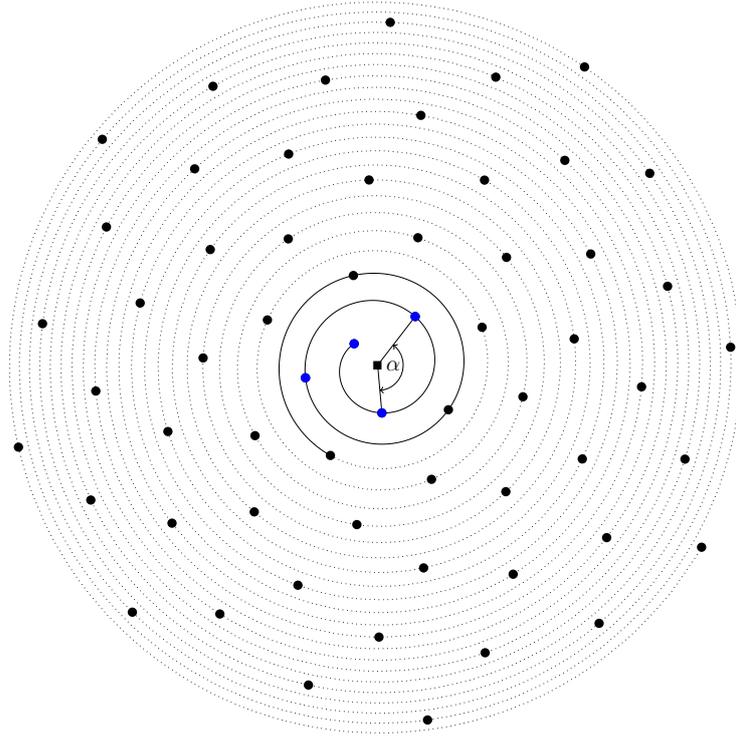


Figure 10.5: This sketch illustrates the general structure of the biomimetic pattern. All heliostats are placed on the spiral around the tower (denoted by ■). The angle distance between adjacent heliostats is α .

10.1.5. Contracted honeycombs

The following pattern has, to our knowledge, not been suggested for heliostat fields, but originates from logistics [12]. There, the setting consists of a center and a number of local distribution points. The task is to minimize both the distance of all distribution points to the center and the distance of all points in the area to the closest distribution point. It is known that contracted honeycombs are a good solution for this scenario, which is rather intuitive if we consider that regular hexagons (often referred to as honeycombs) form a lattice that equally divides the plane.

For heliostat fields we also value heliostats which are closer to the tower, and the area which is not covered by heliostats should be minimized. Although the logistics scenario does not consider effects like blocking and shading, it might nevertheless be a good candidate.

The pattern is based on a regular hexagon grid, but shifts all points using

$$z' = z^2 \quad \forall z \in \mathbb{C} \quad \iff \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x^2 - y^2 \\ 2xy \end{pmatrix} \quad \forall \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2, \quad (10.5)$$

resulting in Figure 10.6.

The only free parameter in this pattern is the zoom, i.e. the dimension of the

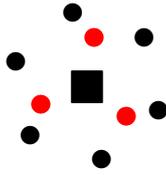


Figure 10.6: This sketch shows how the points in the original hexagon grid (depicted in black) are shifted by the transformation z^2 (resulting points shown in red). Since this quadratic function is symmetric, it results in only three points.

underlying hexagon grid. We also considered using a more general form $z' = z^p$ with p as a free parameter, but experiments showed this would yield rather strange results.

10.1.6. JSON file

The JSON File for a pattern contains a node named by the pattern. For all patterns only the settings for the downhill simplex [37] have to be chosen, see Table 25.

Parameter name	Unit	Data Type	Range
tol	factor	double	[0, 1]
max_iterations	number	double	[0, ∞)

Table 25: JSON file for the settings of the downhill simplex.

10.2. Genetic algorithm

The genetic algorithm which we use in this work is a deviate from the genetic algorithm in Netz [38]. We use the discrete version of the algorithm which bases on the extensible genetic algorithms library called GeneiAL [19]. In this section we will use the expressions gene for a position, chromosome for a position set and population or generation for a set of positions sets. The genetic algorithm uses eight components that are presented in order of execution. In Figure 10.7 the pipeline for the genetic algorithm is shown.

1. Chromosome Factory
2. Stopping Criteria
3. Selection Operation
4. Coupling Operation
5. Crossover Operation

6. Choosing Operation
7. Mutation Operation
8. Replacement Operation

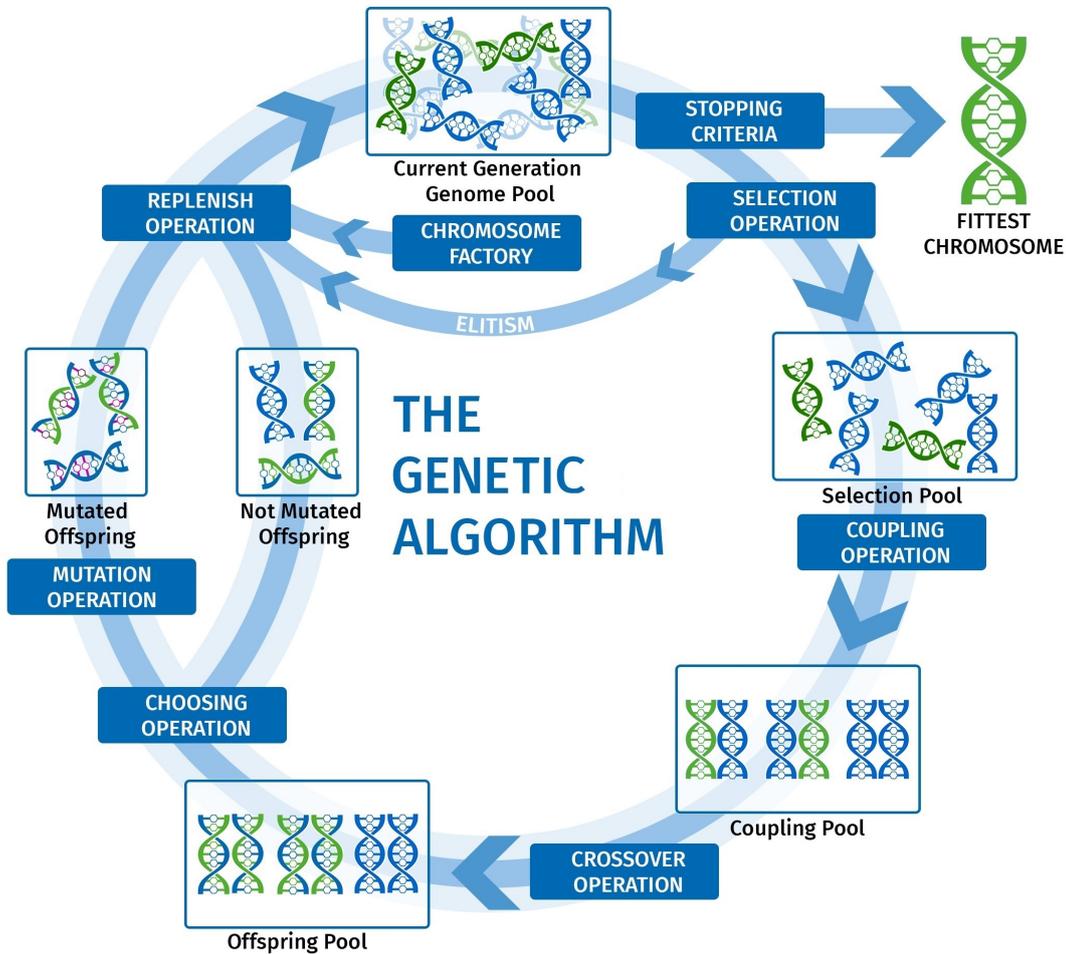


Figure 10.7: This sketch revised from Netz [38] shows how the components in the genetic algorithm work together.

10.2.1. Discrete Value Mapping

In order to connect the genetic algorithm to our problem we need to define a chromosome type which allows for comparing two different chromosomes. Netz [38] defined a discrete value mapping, which allows for comparing. To achieve this we discretize

the available area into ρ^2 many cells, where ρ defined by the resolution. The available positions can then be described by:

$$x^d = a \cdot \frac{x_{\max} - x_{\min}}{\rho} + \frac{x_{\max} - x_{\min}}{2 \cdot \rho} \quad \text{with } a \in [0, \rho - 1] \wedge a, \rho \in \mathbb{N}, \quad (10.6)$$

$$y^d = b \cdot \frac{y_{\max} - y_{\min}}{\rho} + \frac{y_{\max} - y_{\min}}{2 \cdot \rho} \quad \text{with } b \in [0, \rho - 1] \wedge b, \rho \in \mathbb{N}. \quad (10.7)$$

With these x^d and y^d we can build a chromosome γ^d :

$$\gamma^d = \{\{x_1^d, y_1^d\}, \{x_2^d, y_2^d\}, \dots, \{x_n^d, y_n^d\}\} \quad (10.8)$$

where n corresponds to the number of heliostats to be placed. With this we can build different chromosomes to a population of size N_{pop} .

10.2.2. Chromosome Factory

The Chromosome Factory is used to create random, but sound chromosomes. In the very first step it provides the genetic algorithm with a current population (Current Generation Genome Pool, see Figure 10.7).

Soundness is provided by the chromosome factory by an iterative process which adds heliostats in each iteration. The first step of an iteration is the generation of a random position in the field boundaries. This position is then added to the chromosome and the chromosome is checked against the model. If the chromosome only contains valid positions the currently generated position is kept. Otherwise it gets removed from the chromosome. This behavior is shown in Listing 1.

```

1 Chromosome ChromosomeFactory :: generateChromosome ()
2   Chromosome chrom;
3   while chrom.size () < N_genes do
4     Position pos = generateRandomPos ()
5     chrom.add (pos)
6     if chrom.valid () then
7       noop
8     else
9       chrom.remove (pos)
10    end if
11  end while

```

Listing 1: Chromosome Factory Pseudo-Code

The chromosome factory also consists of a fallback mode if too many attempts fail to generate one new position. This fallback mode then iterates all positions in a random direction. If a valid position is found along the way it moves back to its normal behavior, else it assumes the field to be too dense and will terminate.

10.2.3. Stopping Criteria

There are multiple ways to define stopping criteria for the genetic algorithm. For this genetic algorithm we allow a combination of two common criteria which are a maximum number of iterations (e.g. generations) and an optimal fitness value. Other more complex stopping criteria are stagnation recognition or a possible future progress detection.

10.2.4. Selection Operation

Each generation is ordered by its fitness. To determine the fitness of a gene (e.g. a position) we use the optical efficiencies. The fitness of a chromosome is then the average fitness of its genes. The selection operator determines chromosomes which are used for the crossover operation (see Section 10.2.6).

The selection operation provides a set of elite chromosomes which are passed to the replacement operation (see Section 10.2.9) without modification and passes a selection pool to the coupling operation (see Section 10.2.5). The first set provides elitism whilst the second shall create diversity. We use a "roulette wheel selection" [38] where each chromosome is picked by probability $p_{s,i}$ depending on their fitness. For given fitness f_i the probability of chromosome i to be selected is

$$p_{s,i} = \frac{f_i}{\sum_{j=1}^{N_{\text{pop}}} f_j}. \quad (10.9)$$

10.2.5. Coupling Operation

We use a random coupling operation to find disjunct pairs within the selection pool. The resulting pairs are then forwarded to the crossover operation.

10.2.6. Crossover Operation

The crossover operation or recombination is a difficult task in an optimization problem with interactive restriction between genes, namely the proximity restriction of two positions. Therefore classic crossover operations like one-point crossover, N-point crossover or uniform crossover (see Figure 10.8) are not applicable on our problem.

We use a one-step crossover from [49] to solve our proximity restrictions and get sound offspring. The one-step crossover first merges both chromosomes into a combined set. All proximity restrictions are then solved by removing those of the second parent that are in conflict. After this the set is reduced to the n best positions/genes where n is the amount of heliostats to be placed. Figure 10.9 shows the one-step crossover method.

10.2.7. Choosing Operation

The introduction of new aspects through mutation to existing chromosomes can lead to worsen chromosomes with an overall good fitness. Therefore only some chromosomes

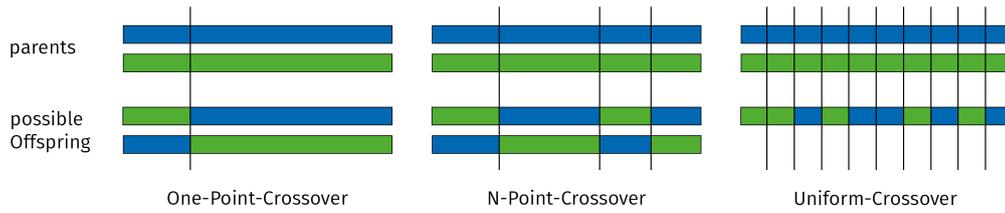


Figure 10.8: Classical crossover operations: one-point crossover ('flips' both chromosomes at one point), N-point crossover ('flips' chromosomes at n points) and uniform crossover (picks random genes on a uniform distributions from both parents) [38]

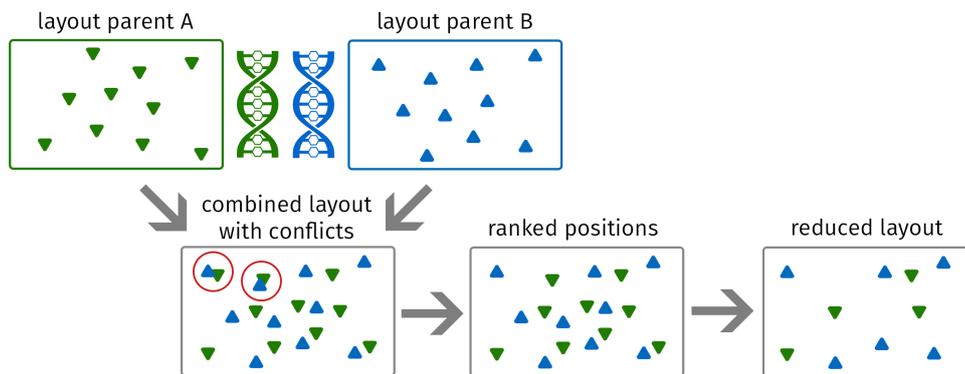


Figure 10.9: One-step crossover as defined in Netz [38]

of the offspring pool are selected. The choosing operation divides the offspring pool into a set of chromosomes which get mutated and one set which does not get mutated. Each chromosome is chosen at random with possibility p_{choosing} .

10.2.8. Mutation Operation

Mutation is used to amplify diversity by changing the chosen chromosomes. Ideally only a small portion of each chromosome is changed, so that the majority of the genetic information stays the same.

For each chromosome in the chosen set we choose a n_{mut} at random such that $n_{\text{mut}}^{\min} \leq n_{\text{mut}} \leq n_{\text{mut}}^{\max}$. Then n_{mut} positions/genes are chosen at random to be mutated, where positions/genes with lower fitness have a higher chance to be chosen because they have a higher chance to be improved by the mutation. Therefore we increase the likelihood of the chromosomes fitness to raise during mutation.

For position $p_i = (x, y)$ a minimal value δ_{\min} is calculated in x- and y-direction to ensure that the mutated position p_{mut} is indeed in a different cell. The computation

for the minimal values is

$$\delta_{\min_x} = \frac{x_{\max} - x_{\min}}{2\rho}, \quad (10.10)$$

$$\delta_{\min_y} = \frac{y_{\max} - y_{\min}}{2\rho}. \quad (10.11)$$

Then random variables r_x and r_y are picked which decide in which directions the position is altered. We calculate a random position (x_r, y_r) with

$$x_r = \begin{cases} \text{Random}(x + \delta_{\min_x}, x_{\max}), & \text{if } r_x > 0 \\ \text{Random}(x_{\min}, x - \delta_{\min_x}), & \text{if } r_x \leq 0 \end{cases} \quad (10.12)$$

$$y_r = \begin{cases} \text{Random}(y + \delta_{\min_y}, y_{\max}), & \text{if } r_y > 0 \\ \text{Random}(y_{\min}, y - \delta_{\min_y}), & \text{if } r_y \leq 0 \end{cases} \quad (10.13)$$

With this we can calculate the mutated position $p_{\text{mut}} = (x_{\text{mut}}, y_{\text{mut}})$ with

$$x_{\text{mut}} = (x + (r_x \cdot \delta_{\min_x})) \cdot (1 - \mu) + (x_r \cdot \mu) \quad (10.14)$$

$$y_{\text{mut}} = (y + (r_y \cdot \delta_{\min_y})) \cdot (1 - \mu) + (y_r \cdot \mu) \quad (10.15)$$

where μ is the mutation rate.

Now the mutated position p_{mut} has to be transformed into a discretized mutated position p_{mut}^d . For this we calculate a cell index c with

$$c = (c_{\text{row}} \cdot \rho) + c_{\text{column}} \quad (10.16)$$

where

$$c_{\text{row}} = \left\lfloor \frac{y_{\text{mut}}}{\frac{|y_{\max} - y_{\min}|}{\rho}} \right\rfloor \quad (10.17)$$

$$c_{\text{column}} = \left\lfloor \frac{x_{\text{mut}}}{\frac{|x_{\max} - x_{\min}|}{\rho}} \right\rfloor. \quad (10.18)$$

With the cell index c we can calculate the discretized mutated position $p_{\text{mut}}^d = (x_{\text{mut}}^d, y_{\text{mut}}^d)$ with

$$x_{\text{mut}}^d = \frac{c - (c \bmod \rho)}{\rho} \cdot \frac{|x_{\max} - x_{\min}|}{\rho} + \frac{|x_{\max} - x_{\min}|}{2 \cdot \rho} \quad (10.19)$$

$$y_{\text{mut}}^d = c \bmod \rho \cdot \frac{|y_{\max} - y_{\min}|}{\rho} + \frac{|y_{\max} - y_{\min}|}{2 \cdot \rho} \quad (10.20)$$

During the mutation operation we ensure that all restrictions are satisfied and no cell contains more than one position.

10.2.9. Replacement and Replenish Operation

The population Γ_{next} for the next generation is assembled by the replacement and replenish operation. Let Γ_{current} be the population after the mutation operation from the previous section. We delete the worst chromosome from Γ_{current} until

$$|\Gamma_{\text{current}}| = N_{\text{pop}} - N_{\text{rand}}, \quad (10.21)$$

where N_{rand} is the number of randomly added genes to the population.

Since the fitness of randomly generated chromosomes after a few generations is worse than most chromosomes, they would be removed when reducing the set to a size of N_{pop} chromosomes. To keep the diversity of randomly generated chromosomes the replenishing is done after the removing of the worst chromosomes. The population of the next generation is then

$$\Gamma_{\text{next}} = \Gamma_{\text{current}} \cup \{\gamma_1, \gamma_2, \dots, \gamma_{N_{\text{rand}}}\}, \quad (10.22)$$

where $\gamma_1, \gamma_2, \dots, \gamma_{N_{\text{rand}}}$ are randomly generated chromosomes from the chromosome factory (see Section 10.2.2). The diversity is then moved to next generations by the coupling and crossover operations, where the randomly generated chromosomes could have an impact.

10.2.10. JSON file

In Table 26 the parameters of the JSON File specifying the settings for the genetic algorithm are shown. The settings lie within a node called "ga".

Parameter name	Unit	Data Type	Range
resolution	number	int	$[N_{\text{pop}}, \infty)$
amount_chromosomes_per_generation	number	int	$[1, \infty)$
mating_pool_size	number	int	$[1, N_{\text{pop}}]$
offspring_pool_size	number	int	$[1, N_{\text{pop}}]$
choosing_probability	propability	double	$[0,1]$
min_mutation_genes	number	int	$[1, \text{max_mutation_genes}]$
max_mutation_genes	number	int	$[\text{min_mutation_genes}, N_{\text{pop}}]$
mutation_rate	factor	double	$[0,1]$
elitism	number	int	$[1, N_{\text{pop}}]$
amount_random_genes	number	int	$[0, N_{\text{pop}}]$
max_iteration	number	int	$[1, \infty)$
max_fitness		double	$[0,1]$
chromosome_factory_max_creation_attempts	number	int	$[1, \infty)$

Table 26: JSON file for the settings of the genetic algorithm.

10.3. Variable neighborhood descent

Variable neighborhood descent (VND) is a variant of the variable neighborhood search (VNS) proposed by Hansen and Mladenović [22, 23] which utilizes defined neighborhoods to search a local optimum in respect to all neighborhoods. For a defined order of neighborhoods VND starts with the first neighborhood and searches a better solution in this neighborhood. If a better solution was found the algorithm starts again with the first neighborhood. Otherwise it progresses to the next neighborhood. Those steps are repeated until no better solution can be found and the last neighborhood is reached. A pseudo-code implementing this principle is presented in Listing 2.

```
1 // Input: initial solution  $s^0$ 
2 set  $k := 1$  (neighborhood counter),  $s := s^0$ 
3 repeat
4   repeat
5     search neighborhood  $N_k(s)$  for improving neighbor
6      $s' \in N_k(s)$  with  $eval(s') > eval(s)$ 
7     if improving neighbor  $s'$  found then
8       set  $s := s'$ 
9       set  $k := 1$ 
10    until no improvement found
11    set  $k := k + 1$ 
12 until  $k > k_{max}$ 
13 // Output: local optima  $s$  regarding all neighborhoods
```

Listing 2: VND Pseudo-Code

10.3.1. Neighborhoods

We define a neighborhood as the all sets of heliostat positions which can be reached when moving each heliostat by a certain distance. Since this leaves infinite many possibilities for each heliostat we discretize the direction in which the heliostats can move. We implemented the possibility to search in 4, 8, 12 or 16 directions. These directions are equidistant, e.g. {N, E, S, W} for four directions.

We regard more distances close to the heliostat position and few far away. The calculation for the distances is

$$e^{\frac{\log(8*d_{helio})}{N-1} \cdot i}, i \in \{1, \dots, N-1\}, \quad (10.23)$$

where N is the number of neighborhoods. The ordering of the neighborhoods is reverse to the ordering of the distances. Therefore leading to neighborhoods regarding bigger distances first.

10.3.2. JSON file

In Table 27 the parameters of the JSON File specifying the settings for the variable neighborhood descent algorithm are shown. The settings lie within a node called "vnd".

Parameter name	Unit	Data Type	Range
max_iterations	number	int	[0, ∞)
min_improvement	factor	double	[1.0,2.0]
num_neighborhoods	number	int	[1,20]
move_directions	number	int	{4,8,12,16}

Table 27: JSON file for the settings of the variable neighborhood descent algorithm.

11. Multi-Step optimizer

The idea behind a multi-step optimizer is to combine the advantages of different algorithms to shorten the overall runtime whilst getting the best possible outcome. The multi-step optimizer chains different algorithms passing results in between algorithms. This leads to increasing the overall runtime, therefore we adjust the algorithms in such a way that they stop earlier or assume a better initial solution.

11.1. Settings

The settings of the multi-step optimizer decide the overall runtime and the outcome by defining which optimization algorithms run in which order. Furthermore the settings allow for defining the algorithm behavior as defined in Section 10 for each algorithm specifically.

Optimization algorithm pipelines The pipeline defines which and in which order the optimization algorithms should be executed. Thereby passing the outcome of the first algorithm as initial solution to the second algorithm and so forth. There are two exceptions for passing the outcome. First the pattern algorithms do not need an initial solution, therefore they don't get passed the last result. And second the genetic algorithm can handle more than one initial solution, therefore it gets past all outcomes of earlier algorithms in the pipeline.

11.1.1. JSON file

In Table 28 the parameters of the JSON File specifying the settings for the multi-step optimizer are shown.

Parameter name	Unit	Data Type	Range
chain	-	enum []	{ns-cornfield, radial-staggered, hexagon, spiral, contracted-honeycombs, vnd, genetic-algorithm}

Table 28: JSON file for the settings of the variable neighborhood descent algorithm.

12. Optimizing large-scale solar tower power plants

12.1. Used algorithms

We use all of the patterns described in Section 10.1, the genetic algorithm of Section 10.2 and the variable neighborhood descent of Section 10.3. As stated in Section 11.1 the pattern algorithms do not need initial solution. Therefore the outcome of each pattern will be discussed individually.

The genetic algorithm was run with the patterns as initial solutions as well as no initial solution. The local search was run on the outcome of the genetic algorithm initialized with patterns.

In Table 29 the settings for the complete pipeline from patterns to the genetic algorithm to the variable neighborhood descent can be found. We chose the same stopping criteria for all patterns. The settings for the genetic algorithm without an initial solution are the same but the pipeline only consists of the genetic algorithm.

Parameter	Value
pipeline	[spiral, radial-staggered, ns-cornfield, contracted-honeycombs, hexagon, vnd, genetic-algorithm]
tol	0.0001
max_iterations (patterns)	600
resolution	8000
amount_chromosomes_per_generation	100
mating_pool_size	10
offspring_pool_size	10
choosing_probability	0.1
min_mutation_genes	3
max_mutation_genes	15
mutation_rate	0.75
elitism	10
amount_random_genes	10
max_iterations (genetic_algorithm)	100
max_fitness	1.0
chromosome_factory_max_creation_attempts	100
max_iterations (local_search)	10
min_improvement	1.0001
num_neighborhoods	5
move_directions	8

Table 29: Settings for the multi step optimizer.

12.2. Test case PS10

The optimization of the heliostat positions for the PS10 solar tower power plant regards 624 heliostats, each of which has a mirror area of about 120 square meters. The area is

bounded by $[-400, 400]$ in \mathbf{x} -direction and $[-10, 800]$ in \mathbf{y} -direction. We use 16 optical moments to calculate an annual integration. The original position layout achieves an average optical efficiency of 80.04%.

12.2.1. North-South Cornfield

The north-south cornfield pattern achieved an average optical efficiency of 77.58% in roughly five hours. The resulting positions are displayed in Figure 12.1a.

12.2.2. Radial Staggered

The radial staggered pattern achieved an average optical efficiency of 79.3% in roughly three and a half hours. The resulting positions are displayed in Figure 12.1b.

12.2.3. Hexagon

The hexagon pattern achieved an average optical efficiency of 79.66% in roughly two hours. The resulting positions are displayed in Figure 12.1c.

12.2.4. Spiral

The spiral pattern achieved an average optical efficiency of 78.93% in roughly three and a half hours. The resulting positions are displayed in Figure 12.1d.

12.2.5. Contracted Honeycombs

The contracted honeycombs pattern achieved an average optical efficiency of 78.88% in roughly two hours. The resulting positions are displayed in Figure 12.1e.

12.2.6. Genetic algorithm

The genetic algorithm did not manage to improve the result when initialized with the patterns. Therefore the best solution after 100 generation was the positioning of the heliostats in the hexagon pattern (see Figure 12.1c).

Without initial solutions the genetic algorithm achieved an average optical efficiency of 71.65% after 100 generations. The resulting position set is depicted in Figure 12.2a and the fitness plotted over the generations is depicted in Figure 12.2b.

The genetic algorithm ran for about two days to finish 100 generations in both cases.

12.2.7. Variable neighborhood descent

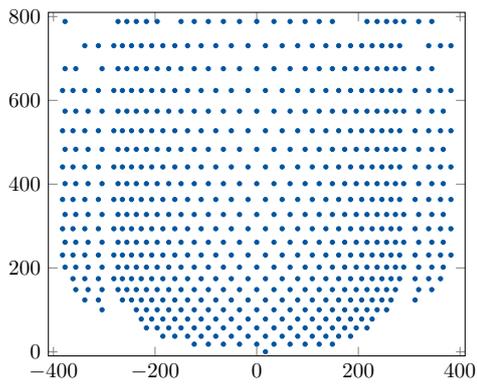
The variable neighborhood descent terminated after three iterations with an average optical efficiency of 79.55%. Figure 12.3 shows the position set produced by the variable neighborhood descent with the hexagon solution as initial solution. The variable neighborhood ran for roughly one day per iteration.

12.3. Economical evaluation

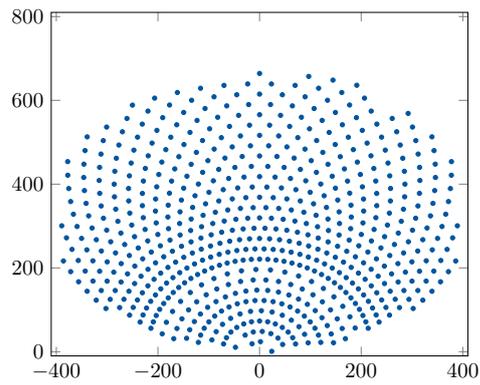
Table 30 shows the final calculated values for the PS10 original layout and our optimized layout. In both cases the investment cost equals 61.65M€.

Layout	AEP	IRR	LCOE	NPV	Payback period
Original	33.77GW	0.0559	18.73¢/(kWh _{el})	32.11M€	12.69a
Optimized	33.48GW	0.0543	18.89¢/(kWh _{el})	30.87M€	12.88a

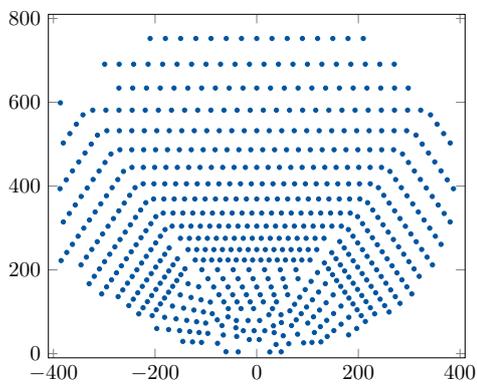
Table 30: Settings for the multi step optimizer.



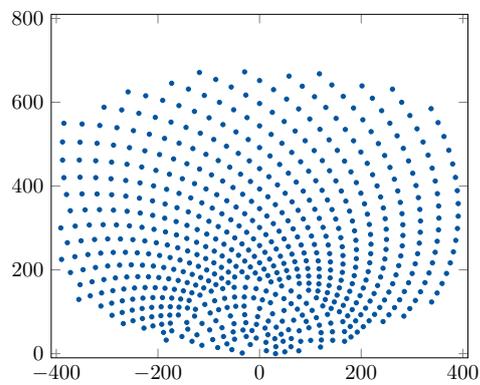
(a) The resulting positions of the north-south cornfield pattern.



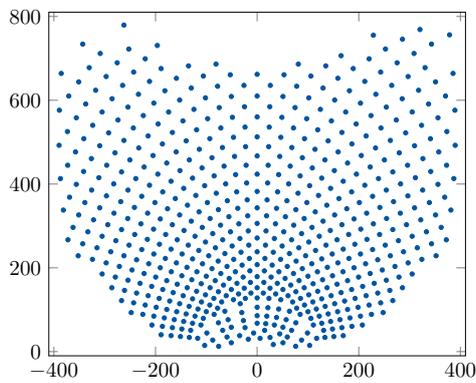
(b) The resulting positions of the radial staggered pattern.



(c) The resulting positions of the hexagon pattern.

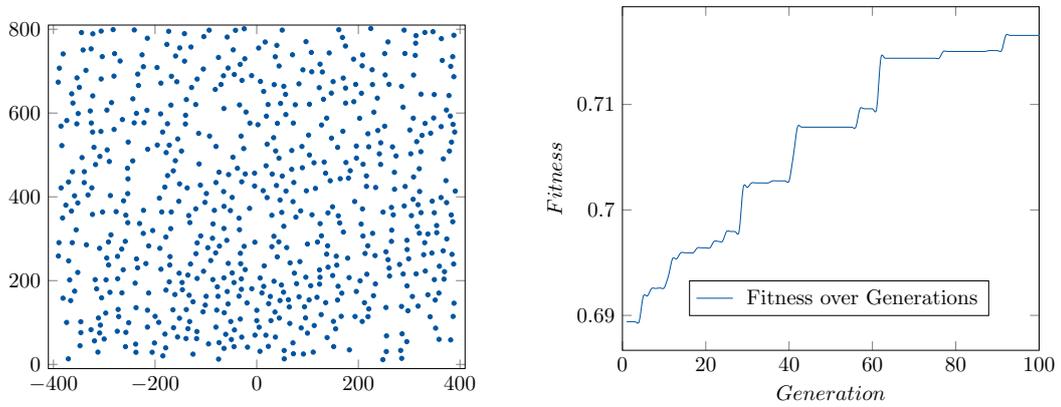


(d) The resulting positions of the spiral pattern.



(e) The resulting positions of the contracted honeycombs pattern.

Figure 12.1: Resulting heliostat positions from patterns



(a) The resulting positions of the genetic algorithm without initial position sets. (b) The fitness of the best chromosome over all generations.

Figure 12.2: Genetic algorithm performance

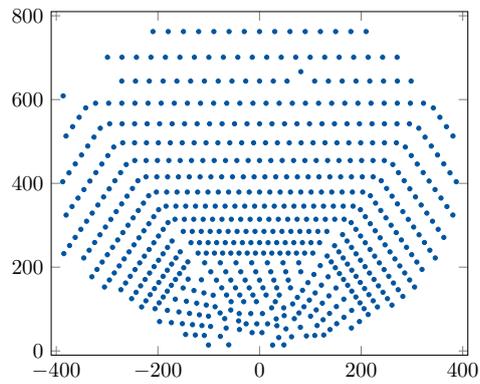


Figure 12.3: Resulting heliostat positions from variable neighborhood descent

Part IV.

Summary

In Part I we presented mathematical the model. In Part II we validated our optical model against state of the art software SolTrace [65] and showed that the our results are correct. Furthermore we evaluated ray resolutions, preselections and data locality to achieve a lower runtime whilst minimizing the error. With this low runtime high accuracy model we showed optimization processes for the heliostat field layout in Part III.

13. Conclusion

A new model for evaluating solar tower power plants was presented. The advantages of this new model lies in its extensive evaluation to achieve a low cost high gain policy that make it suitable for optimization problems.

An accurate description of each aspect of the model was given in Sections 1-5. The presentation included a mathematical description of the scene as well as a description of how we trace a ray. The thermal model was explained with all loses and heat propagation. A bilinear interpolation is used to simulate the power block.

The cross validation against the state of the art ray-tracer SolTrace showed that we achieve more accurate and reliable results. In 17 of 20 test cases our results were between the SolTrace maximum and minimum over 10 runs. The calculated gap for each test case was smaller in our tool than the one in SolTrace for the same amount of runs. Summarized our tool is faster and more accurate for each test case. Unfortunately the cell-wise convolution ray tracing method seems not applicable on canted heliostats. However the convolution method yields reasonable results and is a good candidate for further investigation and analytical optical power calculations.

It was shown that preselections for shading and blocking calculations are not efficient with the developed ray tracing approach. Furthermore, it was shown that data locality during hierarchical ray tracing as well as a reduction in the number of rays improve the runtime of the model. We found out that roughly 1 ray per facet ($\approx 5m^2$) suffices to optimize a solar tower power plant with an error of less than 0.05% and a more precise evaluation with an error of less than 0.001% is achieved when tracing $945 \frac{\text{rays}}{m^2}$.

The optimization of solar tower power plants with state of the art pattern-based optimization produces good results. For an optimization on the Plant Solar 10 we achieved scores of 77.58% for a north-south cornfield, 78.88% for contracted honeycombs, 78.93% for a spiral, 79.3% for a radial staggered and 79.66% for a hexagon layout without post processing. These results are almost equal to the original layout of the Plant Solar 10 in Sevilla which achieved 80.04% efficiency.

The genetic algorithm did not produce good results. The parameters chosen for the genetic algorithm may not be optimal for the given problem. When considering

a genetic algorithm with its high runtime a suitable initial solution set with the right parameters has to be found.

The variable neighborhood descent achieved an average efficiency of 79.55%. Right now it is unclear how our implementation was able to worsen the result from the 79.66% of the hexagon solution.

14. Outlook

Despite the accomplishments of our model we can define new goals for the future. In the following for most parts of our tool an outlook is given what can or will be implemented in the near future.

The Sun can make use of Buie's sunshape model [10, 11]. This would allow us to compute the effective size of the image on the receiver produced by the incoming sunlight.

The ray-tracer is a complex technology which can be further optimized and extended. We can introduce a two axis slope error or even a map of different slope errors to calculate different optical errors depending on the position of the traced ray. Furthermore a GPU accelerated ray tracing and the usage of new technologies like NVIDIA RTX [14] have to be studied in order to determine their usefulness in this field.

An annual integration should consider each hour of the year. Since this is not suitable for optimization a study has to be carried out which integration domain and resolution to choose to best approximate the summation of each hour of the year.

Pattern-based optimization is currently the state of the art in layout optimization for heliostat fields. Our patterns only depend on their parameters, whilst some of them could be removed and implicitly handled in the pattern generation. This could not only yield to better results, but also achieve lower run times during an optimization through a solver since it has less free variables to optimize.

Despite the good result of our downhill simplex solver more solvers can be implemented in order to find the best one suitable for this problem, e.g. the one with a low run time and possible best outcome. Furthermore, a parameter study has to be carried out to find out which stopping criteria fits our problem best, e.g. which *tol* and *max_iteration*, see Section 10.1.

The genetic algorithm may not be suitable for this problem, since the calculation is just too complex and costly for the amount of generations and population needed. This may be due to the fact that uniformly random generated positions are the most far from the actual optima. This in mind the chromosome factory from Section 10.2.2

could consider choosing random parameters for random patterns. Thereby introducing diversity in patterns and in solution space.

Diversity in optimization runs for different CSP may need more complex stopping criteria like the stagnation recognition or possible future progress. The simple max iterations or optimal fitness value stopping criteria may run too long for less positions or terminate too early for more positions to optimize.

The coupling operation chooses at random but maybe it is more suitable to choose via a metric which finds the most opposing chromosomes and pair them since they can create the most diversity.

A parameter study has to be carried out to find a suitable set of parameters for our problem. If there is no such parameter set, the genetic algorithm may be too costly for this optimization problem and should be considered bad practice when compared to pattern-base optimization.

Variable neighborhood descent can profit from a smarter approach where a first fit instead of a best fit method is used in order to reduce the run-time. Furthermore we have to check against the implementation how worse results can be accepted.

Less evaluation will result in less run-time. Therefore if only one or a few local positions are evaluated it may suffice to only reevaluate a subset of the current solution. A method can be developed to find a suitable subset for less than a defined number of heliostats changed. Then a study has to find out, if this results in less run time, e.g. if the method is faster than the traded run time. Also the validity of said method has to be proven.

Furthermore, an evaluation always carries out all calculation despite the requested value. A feature could be implemented, only calculating the value needed and dropping unnecessary calculations, thereby minimizing the run time.

The economic evaluation calculates investment costs which are too high compared to the ones given by Osunaa et al. [41]. In order to solve this the annual integration has to be validated. Afterwards the settings of the economic model, see Table 19, need suitable values.

References

- [1] IRENA (2018). *Renewable Power Generation Costs in 2017*. International Renewable Energy Agency, Abu Dhabi, 2018.
- [2] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, ninth dover printing, tenth gpo printing edition, 1964.
- [3] Florian Arbes, Markus Wöhrbach, Daniel Gebreiter, and Gerhard Weinrebe. Towards high efficiency heliostat fields. *AIP Conference Proceedings*, 1850(1):030001, 2017. doi: 10.1063/1.4984344. URL <https://aip.scitation.org/doi/abs/10.1063/1.4984344>.
- [4] Germain Augsburger. *Thermo-economic optimisation of large solar tower power plants*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2013.
- [5] National Solar Radiation Data Base. Typical meteorological year 3. Website, 2015. Available online at https://rredc.nrel.gov/solar/old_data/nsrdb/1991-2005/tmy3/, visited on July 25st 2018.
- [6] Boris Belhomme, Robert Pitz-Paal, Peter Schwarzbözl, and Steffen Ulmer. A new fast ray tracing tool for high-precision simulation of heliostat fields. *Journal of Solar Energy Engineering*, 131(3):031002, 2009.
- [7] M Blanco, Juana M Amieva, and Azael Mancilla. The tonatiuh software development project: An open source approach to the simulation of solar concentrating systems. *Proceedings of the IMEC2005, Orlando, FL*, pages 157–164, 2005.
- [8] Sebastian-James Bode and Paul Gauché. Review of optical software for use in concentrating solar power systems. In *South African Solar Energy Conference, Stellenbosch*, 2012.
- [9] R. Buck. Heliostat field layout using non-restricted optimization. In *SolarPACES Conference*, 2012.
- [10] D. Buie, C.J. Dey, and S. Bosi. The effective size of the solar cone for solar concentrating systems. *Solar Energy*, 74(5):417 – 427, 2003. ISSN 0038-092X. doi: [https://doi.org/10.1016/S0038-092X\(03\)00156-7](https://doi.org/10.1016/S0038-092X(03)00156-7). URL <http://www.sciencedirect.com/science/article/pii/S0038092X03001567>.
- [11] Damien Buie, A Monger, and Christopher Dey. Sunshape distributions for terrestrial solar simulations. *Solar Energy*, 74:113–122, 02 2003. doi: 10.1016/S0038-092X(03)00125-7.
- [12] John Gunnar Carlsson and Fan Jia. Euclidean hub-and-spoke networks. *Operations research*, 61(6):1360–1382, 2013.

- [13] Francisco J Collado and Jesús Guallar. Campo: Generation of regular heliostat fields. *Renewable energy*, 46:49–59, 2012.
- [14] NVIDIA Developer. Nvidia rtx™. Website, 2018. Available online at <https://developer.nvidia.com/rtx>, visited on November 1st 2018.
- [15] John A. Duffie and William A. Beckman. *Solar Engineering of Thermal Processes*. John Wiley & Sons, fourth edition, 2013. ISBN 978-0-470-87366-3.
- [16] Michael Epstein, D. Liebermann, M. Rosh, and Arthur J. Shor. Solar testing of 2 mwth water/steam receiver at the weizmann institute solar tower. *Solar Energy Materials*, 24(1):265 – 278, 1991. ISSN 0165-1633. doi: [https://doi.org/10.1016/0165-1633\(91\)90067-U](https://doi.org/10.1016/0165-1633(91)90067-U). URL <http://www.sciencedirect.com/science/article/pii/016516339190067U>.
- [17] Patricia Kuntz Falcone. A handbook for solar central receiver design. Technical Report SAND-86-8009, Sandia National Labs., Livermore, CA (USA), 12 1986.
- [18] Pierre Garcia, Alain Ferriere, and Jean-Jacques Beziau. Codes for solar flux calculation dedicated to central receiver system applications: a comparative review. *Solar Energy*, 82(3):189–197, 2008.
- [19] Geneial. Genetic algorithm library geneial. Available online at <http://www.geneial.org>, visited on September 1st 2018.
- [20] M. Geyer and W.B. Stine. Power from the sun. Website, 2001. Available online at www.powerfromthesun.net/book.html, visited on July 25st 2018.
- [21] CMI Group. Cmi energy. Website. Available online at <http://www.cmigroupe.com/thermal-solar-receivers>, visited on September 25th 2018.
- [22] P. Hansen and N. Mladenović. Variable neighborhood search for the p-median. *Location Science*, 5(4):207 – 226, 1997. ISSN 0966-8349. doi: [https://doi.org/10.1016/S0966-8349\(98\)00030-8](https://doi.org/10.1016/S0966-8349(98)00030-8). URL <http://www.sciencedirect.com/science/article/pii/S0966834998000308>.
- [23] Pierre Hansen and Nenad Mladenović. *Variable Neighborhood Search*, pages 211–238. Springer US, Boston, MA, 2005. ISBN 978-0-387-28356-2. doi: 10.1007/0-387-28356-0_8. URL https://doi.org/10.1007/0-387-28356-0_8.
- [24] Gregor Heimig. Development of a techno-economic model for solar tower power plants. Master’s thesis, MathCCES - RWTH Aachen, gregor.heimig@rwth-aachen.de, 11 2017.
- [25] K. Hennecke, P. Schwarzbözl, G. Koll, M. Beuter, B. Hoffschmidt, J. Götsche, and T. Hartz. The solar power tower jülich — a solar thermal power plant for test and demonstration of air receiver technology. In D. Yogi Goswami and Yuwen Zhao, editors, *Proceedings of ISES World Congress 2007 (Vol. I – Vol. V)*, pages

- 1749–1753, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-540-75997-3.
- [26] M Kiera. Description of the computing code system hfcac. *I. Report, GASTIAS-BT-200000-075, Interatom Report*, 1986.
- [27] 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.
- [28] Reinhold Kneer. Wärme- und Stoffübertragung I/II, Vorlesungsskript. Lecture notes, Lehrstuhl für Wärme- und Stoffübertragung, RWTH Aachen University, 2015. Ausgabe vom 2. September 2015.
- [29] Gregory J Kolb. Economic evaluation of solar-only and hybrid power towers using molten-salt technology. *Solar Energy*, 62(1):51 – 61, 1998. ISSN 0038-092X. doi: [https://doi.org/10.1016/S0038-092X\(97\)00075-3](https://doi.org/10.1016/S0038-092X(97)00075-3). URL <http://www.sciencedirect.com/science/article/pii/S0038092X97000753>.
- [30] P.L. Leary and J.D. Hankins. User’s guide for MIRVAL: a computer code for comparing designs of heliostat-receiver optics for central receiver solar power plants. Technical report, Sandia Laboratories, 1979.
- [31] Erminia Leonardi and Bruno D’Aguanno. Crs4-2: A numerical code for the calculation of the solar power collected in a central receiver system. *Energy*, 36(8): 4828–4837, 2011.
- [32] FW Lipps and LL Vant-Hull. A cellwise method for the optimization of large central receiver systems. *Solar Energy*, 20(6):505–516, 1978.
- [33] Matti Lubkoll, Theodor W. von Backström, and Thomas M. Harms. Performance outlook of the scrap receiver. *AIP Conference Proceedings*, 1734(1):030024, 2016. doi: 10.1063/1.4949076. URL <https://aip.scitation.org/doi/abs/10.1063/1.4949076>.
- [34] S.L. Lutchman, A.A. Groenwold, P. Gauché, and S. Bode. On using a gradient-based method for heliostat field layout optimization. In *Proceedings of the SolarPACES 2013 International Conference*, volume 49 of *Energy Procedia*, pages 1429–1438. Elsevier, 2014.
- [35] Gabriel Morin. *Techno-Economic Design Optimization of Solar Thermal Power Plants*. PhD thesis, Technische Universität Braunschweig, 2011.
- [36] National Renewable Energy Laboratory (NREL). SSC (SAM Simulation Core). Open Source Software Repository, 2017. URL <https://github.com/NREL/SSC>. Last modified October 20, 2017.

- [37] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965. doi: 10.1093/comjnl/7.4.308. URL <http://dx.doi.org/10.1093/comjnl/7.4.308>.
- [38] Lukas Netz. Analysis on discretization of gene parameters in evolutionary algorithms. Master’s thesis, i2 - RWTH Aachen, lukas.netz@rwth-aachen.de, 12 2017.
- [39] C.J. Noone, M. Torrilhon, and A. Mitsos. Heliostat field optimization: A new computationally efficient model and biomimetic layout. *Solar Energy*, 86(2):792–803, 2012.
- [40] U.S. Department of Energy’s (DOE) Building Technologies Office (BTO). Energyplus. Website, 2018. Available online at <https://energyplus.net/> managed by the National Renewable Energy Laboratory (NREL), visited on July 25st 2018.
- [41] Rafael Osunaa, Rafael Olavarriáaa, Rafael Morilloa, Marcelino Sáncheza, Felipe Canteroa, Valerio Fernández-Queroa, Pedro Roblesb, Teodoro López del Cerrob, Antonio Estebanb, Francisco Cerónb, Juan Talegónb, Manuel Romerod, Felix Téllezd, MaJesus Marcosc, Diego Martíneze, Antonio Valverdee, Rafael Monterreale, Robert Pitz-Paalf, George Brakmann, and Manuel Silva. Ps10, construction of a 11mw solar thermal tower plant in seville, spain. In *Proceedings of the SolarPACES 2006*, 2006.
- [42] Q. Pham, C. Gregory, M. Slack, B. Gross, D. Reznik, and P. Arbogast. Heliostat array layouts for multi-tower central receiver solar power plants, May 28 2009. URL <https://www.google.com/patents/US20090133685>. US Patent App. 12/324,734.
- [43] C.L. Pitman and L.L. Vant-Hull. Atmospheric transmittance model for a solar beam propagating between a heliostat and a receiver. *ASES Progress in Solar Energy*, pages 1247–1251, 1982.
- [44] CL Pitman and LL Vant-Hull. The university of houston solar central receiver code system: concepts, updates, and start-up kits. Technical Report SAND-88-7029, Houston Univ., TX (USA). Energy Lab., 3 1989.
- [45] R. Pitz-Paal, N.B. Botero, and A. Steinfeld. Heliostat field layout optimization for high-temperature solar thermochemical processing. *Solar Energy*, 85(2):334–343, 2011.
- [46] A. Rabl. *Active solar collectors and their applications*. Oxford University Press, 1985.
- [47] Pascal Richter. *Simulation and optimization of solar thermal power plants*. Dissertation, RWTH Aachen University, Aachen, 2017. URL <http://publications.rwth-aachen.de/record/690762>. Veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, RWTH Aachen University, 2017.

- [48] Pascal Richter, Erika Ábrahám, and Martin Frank. Multi-objective optimization of solar tower power plants. 06 2014.
- [49] Pascal Richter, David Laukamp, Levin Gerdes, Martin Frank, and Erika Ábrahám. Heliostat field layout optimization with evolutionary algorithms. In *GCAI*, pages 240–252, 2016.
- [50] Pascal Richter, Gregor Heiming, Nils Lukas, and Martin Frank. Sunflower: A new solar tower simulation method for use in field layout optimization, 09 2017.
- [51] David Riveros-Rosas, Marcelino Sánchez-González, and Claudio A Estrada. Three-dimensional analysis of a concentrated solar flux. *Journal of Solar Energy Engineering*, 130(1):014503, 2008.
- [52] JP Roccia, B Piaud, C Coustet, C Caliot, E Guillot, G Flamant, and J Delatorre. Solfast, a ray-tracing monte-carlo software for solar concentrating facilities. In *Journal of Physics: Conference Series*, volume 369, page 012029. IOP Publishing, 2012.
- [53] Marc Röger, Lars Amsbeck, Birgit Gobereit, and Reiner Buck. Face-down solid particle receiver using recirculation. *Journal of Solar Energy Engineering*, 133: 031009–1, 08 2011.
- [54] Marcelino Sanchez and Manuel Romero. Methodology for generation of heliostat field layout in central receiver systems based on yearly normalized energy surfaces. *Solar Energy*, 80(7):861–874, 2006.
- [55] Sargent & Lundy LLC Consulting Group. Assessment of parabolic trough and power tower solar technology cost and performance forecasts. Subcontractor Report NREL/SR-550-34440, National Renewable Energy Laboratory, Golden, CO (USA), October 2003.
- [56] M. Schmitz, P. Schwarzbözl, R. Buck, and R. Pitz-Paal. Assessment of the potential improvement due to multiple apertures in central receiver systems with secondary concentrators. *Solar energy*, 80(1):111–120, 2006.
- [57] Philipp Schramek, David R Mills, Wes Stein, and Peter Le Lièvre. Design of the heliostat field of the csiro solar tower. *Journal of Solar Energy Engineering*, 131(2):024505, 2009.
- [58] Peter Schwarzbözl, Robert Pitz-Paal, and Mark Schmitz. Visual hfical - a software tool for layout and optimisation of heliostat fields. In *SolarPACES Conference*, 2009.
- [59] F.M.F Siala and M.E Elayeb. Mathematical formulation of a graphical method for a no-blocking heliostat field layout. *Renewable Energy*, 23(1):77–92, may 2001. doi: 10.1016/s0960-1481(00)00159-2. URL <http://www.sciencedirect.com/science/article/pii/S0960148100001592>.

- [60] Dennis L. Siebers and John S. Kraabel. Estimating convective energy losses from solar central receivers. Technical Report SAND84-8717, Sandia National Laboratories, Livermore, CA (USA), 1984.
- [61] D.L. Siebers, R.F. Moffatt, and R.G. Schwind. Experimental, variable properties natural convection from a large, vertical, flat surface. *Journal of Heat Transfer*, 107(1):124–132, February 1985.
- [62] Janna Tinnes. Acceleration of ray-tracer models for the simulation of solar tower power plants with real weather data. Bachelor’s thesis, MathCCES - RWTH Aachen, janna.tinnes@rwth-aachen.de, 10 2017.
- [63] M.J. Wagner. *Simulation and predictive performance modeling of utility-scale central receiver system power plants*. University of Wisconsin–Madison, 2008. URL <https://books.google.de/books?id=LW40AQAAMAAJ>.
- [64] Xiudong Wei, Zhenwu Lu, Zhifeng Wang, Weixing Yu, Hongxing Zhang, and Zhihao Yao. A new method for the design of the heliostat field layout for solar tower power plant. *Renewable Energy*, 35(9):1970–1975, 2010.
- [65] Tim Wendelin. 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, 2003.