

Life-time integration using Monte Carlo Methods when optimizing the design of concentrated solar power plants

Olivier Farges, Jean Jacques Bézian, Hélène Bru, Mouna El-Hafi, Richard Fournier, Christophe Spiesser

► **To cite this version:**

Olivier Farges, Jean Jacques Bézian, Hélène Bru, Mouna El-Hafi, Richard Fournier, et al.. Life-time integration using Monte Carlo Methods when optimizing the design of concentrated solar power plants. Solar Energy, Elsevier, 2015, 113, pp.57-62. 10.1016/j.solener.2014.12.027 . hal-01141476

HAL Id: hal-01141476

<https://hal-mines-albi.archives-ouvertes.fr/hal-01141476>

Submitted on 14 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Life-time integration using Monte Carlo Methods when optimizing the design of concentrated solar power plants

O. Farges^{a,b,*}, J.J. Bézian^a, H. Bru^b, M. El Hafi^a, R. Fournier^c, C. Spiesser^a

^a *Université de Toulouse, École des Mines d'Albi, UMR CNRS 5302, RAPSODEE Research Center, F-81013 Albi, France*

^b *Total Énergies Nouvelles, R&D - Concentrated Solar Technologies, Tour Michelet, 24 cours Michelet, 92069 Paris La Défense Cedex, France*

^c *Université Paul Sabatier, UMR 5213 - Laboratoire Plasma et Conversion d'Énergie (LAPLACE), Bt. 3R1, 118 route de Narbonne, F-31062 Toulouse cedex 9, France*

Abstract

Rapidity and accuracy of algorithms evaluating yearly collected energy are an important issue in the context of optimizing concentrated solar power plants (CSP). These last ten years, several research groups have concentrated their efforts on the development of such sophisticated tools: approximations are required to decrease the CPU time, closely checking that the corresponding loss in accuracy remains acceptable. Here we present an alternative approach using the Monte Carlo Methods (MCM). The approximation effort is replaced by an integral formulation work leading to an algorithm providing the exact yearly-integrated solution, with computation requirements similar to that of a single date simulation. The corresponding theoretical framework is fully presented and is then applied to the simulation of PS10.

Keywords: Concentrated solar power, Monte Carlo methods, yearly energy, integral formulation

1. Introduction

Concentrated solar plants are commonly designed to have the best energy-collection efficiency in nominal conditions on March 21st. However, several codes such as HFLCAL (Schwarzbözl et al., 2009), System Advisor Model (Gilman et al., 2008), UHC or DELSOL already assess the annual performances of large-size heliostat fields. These codes are fast but retain approximations in their resolution methods. In order to decrease the CPU costs, some authors make use of simplified heliostat-flux convolutions (Garcia et al., 2008), reduce the number of heliostats (choosing a representative number of heliostats)(Sanchez

*Principal Corresponding Author

Email address: olivier-farges@mines-albi.fr (O. Farges)

Nomenclature

B	Blocking performance (%)	$\%RSD$	Relative standard deviation (%)
b	Blinn parameter for reflection imperfections	Sh	Shadowing performance (%)
\mathcal{D}_X	Domain of definition of the random variable X	Sp	Spillage performance (%)
DNI	Direct normal irradiance (W m^{-2})	$S_{\mathcal{H}^+}$	Area of mirror (m^2)
E	Yearly average energy (kW h)	\mathcal{T}	Target (the exponent + indicates the active side)
\mathcal{H}	Heliostats surface (the exponent + indicate the active side)	t	Time (h)
$\mathbf{H}()$	The Heaviside step function	\hat{w}	Monte Carlo weight
\mathbf{n}_1	Ideal normal at \mathbf{r}_1	\bar{x}	Average of Monte Carlo weights
\mathbf{n}_h	Effective normal at \mathbf{r}_1 around the ideal normal \mathbf{n}_1	Ω_S	Solar cone (sr)
N_r	Number of rays sampled for a date	ω_1	Direction after reflexion (rad)
P	Power (kW)	ω_S	Direction inside the solar cone (rad)
\mathbf{r}_i	Location	ρ	Mirror reflectivity
		σ	Standard deviation of Monte Carlo weights

and Romero, 2006) or account for blocking and shadowing in simplified manners (Collado, 2008). In all cases, the first question is the accurate prediction of fluxes and temperatures within the receiver, at each date (collected thermal power, hot spots on the receiver-wall), which requires an accurate enough sun-spot model. The second issue is then the integration of these predictions over CSP-lifetime. In the present paper, we describe a Monte Carlo approach allowing to perform this integration with short CPU times, therefore allowing to avoid the step of simplifying the sun-spot model. The reason why CPU times are short is that we handle the multiple combination of integrals by statistical means. Adding a new integral over time (to the already quite complex integral over optical-paths and geometry) does not change the overall complexity level and similar numbers of statistical samples are required: addressing lifetime integrated quantities require CPU times similar to those required to predict the same quantity at a single date. In other terms, the CPU time required to handle multiple integrals is imposed by the integral that is the highest source of variance, and here the leading integral is not the time-integral. The well-known code MIRVAL includes a mode called “Energy Run”, based on a similar approach (Falcone, 1986). The corresponding integral formulation is first presented for a single date (Sec. 2.2), and is only extended to lifetime in Sec. 2.1. The last paragraph (Sec. 3) is

dedicated to the implementation on the PS10 testcase. Even if a straightforward implementation of the method already leads to attractive CPU times, CSP-design optimization implies iterative processes in which all further CPU-time reductions are significant. We therefore show in appendix how such further reductions can be achieved using advanced computer-graphics techniques.

2. The algorithm and its associated integral formulation

2.1. The starting point

To take advantage of fast intersection calculations and parallel computing, we implement our MCM algorithm within the EDStar framework (De La Torre et al., 2014). We start with an algorithm referred as Monte Carlo Fixed Date, or MCFD. It is meant for the design of heliostat fields of Central Receiver Systems (CRS) considering a single date. It predicts the solar power P incident on the receiver. It is very similar to the first example presented at section 4.1 in (De La Torre et al. (2014)): rays are sampled from the heliostat field and are followed until they reach or miss the central receiver. A Monte Carlo weight is associated to each sampled ray and P is evaluated as the average value of a large number of such weights. The details of the ray-sampling procedure are given hereafter.

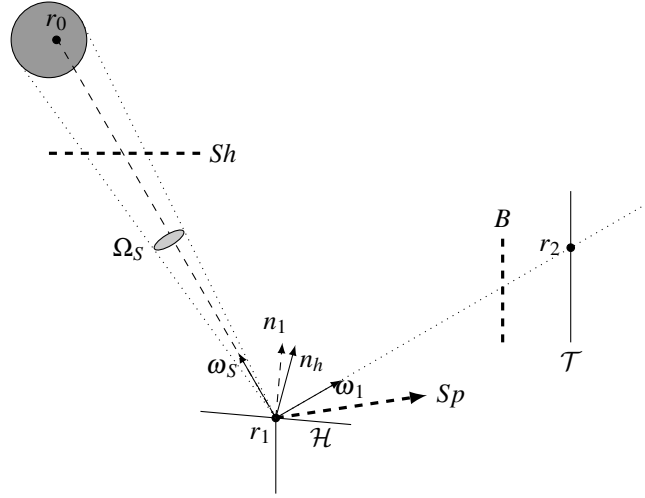


Figure 1: Schematic representation of the Monte Carlo Fixed Date (MCFD) algorithm

- (1) A location r_1 is uniformly sampled on the reflective surface of the whole heliostat field \mathcal{H}^+ of surface $S_{\mathcal{H}^+}$

- (2) A direction $\boldsymbol{\omega}_S$ is uniformly sampled within the solar cone $\boldsymbol{\Omega}_S$ of angular radius θ_S .
- (3) An effective normal vector \mathbf{n}_h is sampled around the ideal normal vector \mathbf{n}_1 at \mathbf{r}_1 representing reflection and pointing imperfections. $\boldsymbol{\omega}_1$ corresponds to the specular reflection of $-\boldsymbol{\omega}_S$ by a surface normal to \mathbf{n}_h
- (4) \mathbf{r}_0 is defined as the first intersection with a solid surface of the ray starting at \mathbf{r}_1 in the direction $\boldsymbol{\omega}_S$
 - (a) If \mathbf{r}_0 belongs to an heliostat surface \mathcal{H} or to the receiver \mathcal{T} , a shadowing effect appears and the Monte Carlo weight is $\hat{w} = 0$;
 - (b) If \mathbf{r}_0 doesn't exist (or is at the sun), the location \mathbf{r}_2 is defined as the first intersection with a solid surface of the ray starting at \mathbf{r}_1 in the direction $\boldsymbol{\omega}_1$
 - (i) If \mathbf{r}_2 belongs to something else than the receiver \mathcal{T} , there is a blocking effect and the Monte Carlo weight is $\hat{w} = 0$;
 - (ii) If \mathbf{r}_2 doesn't exist there is a spillage effect and the Monte Carlo weight is $\hat{w} = 0$;
 - (iii) If \mathbf{r}_2 belongs to the receiver \mathcal{T} , the Monte Carlo weight is $\hat{w} = DNI \times \rho \times (\boldsymbol{\omega}_S \cdot \mathbf{n}_h) \times S_{\mathcal{H}^+}$

This algorithm is equivalent to the integral formulation of Eq. (1) (see Fig. 1).

$$P = \int_{\mathcal{D}_{\mathcal{H}^+}} p_{\mathbf{R}_1}(\mathbf{r}_1) d\mathbf{r} \int_{\mathcal{D}_{\boldsymbol{\Omega}_S}} p_{\boldsymbol{\Omega}_S}(\boldsymbol{\omega}_S) d\boldsymbol{\omega} \int_{\mathcal{D}_{\mathcal{N}_h}} p_{\mathcal{N}_h}(\mathbf{n}_h|\boldsymbol{\omega}_S; b) d\mathbf{n} \hat{w} \quad (1)$$

with the Monte Carlo weight :

$$\hat{w} = \left\{ \begin{array}{l} \mathbf{H}(\mathbf{r}_0 \in \mathcal{H} \cup \mathcal{T}) \times 0 \\ +\mathbf{H}(\mathbf{r}_0 \notin \mathcal{H} \cup \mathcal{T}) \times \left\{ \begin{array}{l} \mathbf{H}(\mathbf{r}_2 \notin \mathcal{T}) \times 0 \\ +\mathbf{H}(\mathbf{r}_2 \in \mathcal{T}) \times DNI \times \rho \times (\boldsymbol{\omega}_S \cdot \mathbf{n}_h) \times S_{\mathcal{H}^+} \end{array} \right\} \end{array} \right\} \quad (2)$$

and probability density functions :

$$p_{\mathbf{R}_1} = \frac{1}{S_{\mathcal{H}^+}} \quad (3)$$

$$p_{\boldsymbol{\Omega}_S} = \frac{1}{\int_{\boldsymbol{\Omega}_S} d\boldsymbol{\omega}_S} = \frac{1}{2\pi(1 - \cos \theta_S)} \quad (4)$$

$$p_{\mathcal{N}_h}(\mathbf{n}_h|\boldsymbol{\omega}_S; b) = \frac{\left(1 + \frac{1}{b}\right) \times (\mathbf{n}_h \cdot \mathbf{n}_1)^{1+\frac{1}{b}}}{2\pi \times \left(1 - \cos^2 + \frac{1}{b} \left(\frac{\pi}{4} - \frac{1}{2} \times \arccos(\boldsymbol{\omega}_S \cdot \mathbf{n}_1)\right)\right)} \quad (5)$$

As presented by De La Torre et al. (2014) in section 4.1, reflection and pointing imperfections are modelled with the Blinn's model (Pharr and Humphreys, 2010), of parameter b . This microfacet model consists in an overall angular spreading obtained by a distribution of effective normal vectors \mathbf{n}_h around the ideal normal \mathbf{n}_1 . This model introduces an exponential falloff to approximate the distribution of microfacet normals¹ and appears in the probability density function $p_{\mathbf{N}_h}(\mathbf{n}_h|\boldsymbol{\omega}_S; b)$.

2.2. Formulation of the lifetime-collected energy

Strong benefits are expected from the establishment of a rigorous equivalence between the algorithm and the integral formulation: a modification of one leads to a modification of the other and vice versa. This allows the choice of either working on the very intuitive nature of the photon-tracking algorithm, or on the mathematical features of the formulation (De La Torre et al., 2014). When modifying the algorithm directly, the formulation helps to check that the modification is rigorous. When working on the formulation, there is no question about rigor and the Monte Carlo algorithm is obtained by translating the successive integrals into successive sampling events. This is the case here: instead of discretizing lifetime and performing a computation at each date, which would require a considerable amount of time, we simply modify Eq. (1), integrating it over time and introducing a probability density p_τ , which leads to an algorithm in which each ray-sampling is preceded by the sampling of a date along lifetime. Eq. (1) becomes

$$\begin{aligned} E &= \int_{Lifetime} P(t) dt \\ &= \int_{Lifetime} p_\tau(t) dt \int_{\mathcal{D}_{\mathcal{H}^+}} p_{\mathbf{R}_1}(\mathbf{r}_1) d\mathbf{r} \int_{\mathcal{D}_{\boldsymbol{\Omega}_S(t)}} p_{\boldsymbol{\Omega}_S(t)}(\boldsymbol{\omega}_S(t)) d\boldsymbol{\omega} \\ &\quad \int_{\mathcal{D}_{\mathbf{N}_h}} p_{\mathbf{N}_h}(\mathbf{n}_h|\boldsymbol{\omega}_S(t); b) d\mathbf{n} \hat{w}(t) \end{aligned} \quad (6)$$

with time-dependent probability density functions

$$p_{\boldsymbol{\Omega}_S(t)} = \frac{1}{\int_{\boldsymbol{\Omega}_S(t)} d\boldsymbol{\omega}_S(t)} = \frac{1}{2\pi(1 - \cos\theta_S(t))} \quad (7)$$

$$p_{\mathbf{N}_h}(\mathbf{n}_h|\boldsymbol{\omega}_S(t); b) = \frac{\left(1 + \frac{1}{b}\right) \times (\mathbf{n}_h \cdot \mathbf{n}_1)^{1+\frac{1}{b}}}{2\pi \times \left(1 - \cos^2 + \frac{1}{b} \left(\frac{\pi}{4} - \frac{1}{2} \times \arccos(\boldsymbol{\omega}_S(t) \cdot \mathbf{n}_1)\right)\right)} \quad (8)$$

¹The distribution of \mathbf{n}_h is truncated to avoid the occurrence of reflected directions towards the surface for quasi-tangent incidences. Thus, the angular spreading is dependent on the incident direction $\boldsymbol{\omega}_S$.

and the Monte Carlo weight that now is

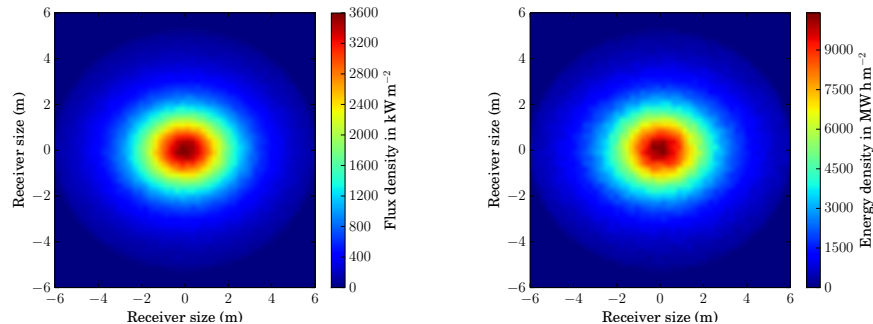
$$\hat{w}(t) = \left\{ \begin{array}{l} \mathbf{H}(\mathbf{r}_0 \in \mathcal{H} \cup \mathcal{T}) \times 0 \\ + \mathbf{H}(\mathbf{r}_0 \notin \mathcal{H} \cup \mathcal{T}) \times \left\{ \begin{array}{l} \mathbf{H}(\mathbf{r}_2 \notin \mathcal{T}) \times 0 \\ + \mathbf{H}(\mathbf{r}_2 \in \mathcal{T}) \times \frac{DNI(t) \times (\boldsymbol{\omega}_S(t) \cdot \mathbf{n}_h) \times S_{\mathcal{H}^+}}{p_{\tau}(t)} \end{array} \right\} \end{array} \right\} \quad (9)$$

In this formulation, time is continuous but the climatic data that we used took the form of a succession of DNI values every hour. We therefore needed an interpolation assumption: as a first order approach we assumed that DNI was a one hour piecewise constant function of time. As far as the sampling probability $p_{\tau}(t)$ is concerned, we used an importance sampling approach: $p_{\tau}(t)$ is simply a normalised form of DNI,

$$p_{\tau}(t) = \frac{DNI(t)}{\int_{Lifetime} DNI(t) dt} \quad (10)$$

This means that the dates with the larger DNI values are sampled more frequently than the low DNI ones. This introduces no bias because $p_{\tau}(t)$ appears both in the integral and in the weight. Practically speaking, this simply means that DNI data are pre-treated: only the non zero DNI values are kept, they are classified in ascending order, cumulated and normalised. Then, when we need to sample time according to $p_{\tau}(t)$, a value of this cumulative function is sampled uniformly in the unit interval and the corresponding one-hour time-interval is retained. Hereafter, the corresponding algorithm is referred as Monte Carlo Solar Tracker, or MCST. The ray-sampling procedure becomes:

- (1) A DNI is uniformly sampled over the lifetime period according to $p_{\tau}(t)$ and the corresponding one-hour time-integral is retained
- (2) A location \mathbf{r}_1 is uniformly sampled on the reflective surface of the whole heliostat field \mathcal{H}^+ of surface $S_{\mathcal{H}^+}$
- (3) A direction $\boldsymbol{\omega}_S$ is uniformly sampled within the solar cone $\boldsymbol{\Omega}_S$ of angular radius θ_S .
- (4) An effective normal vector \mathbf{n}_h is sampled around the ideal normal vector \mathbf{n}_1 at \mathbf{r}_1 representing reflection and pointing imperfections. $\boldsymbol{\omega}_1$ corresponds to the specular reflection of $-\boldsymbol{\omega}_S$ by a surface normal to \mathbf{n}_h
- (5) \mathbf{r}_0 is defined as the first intersection with a solid surface of the ray starting at \mathbf{r}_1 in the direction $\boldsymbol{\omega}_S$
 - (a) If \mathbf{r}_0 belongs to an heliostat surface \mathcal{H} or to the receiver \mathcal{T} , a shadowing effect appears and the Monte Carlo weight is $\hat{w} = 0$;



(a) Flux density map at PS10 receiver on March 21st with $DNI = 1000 \text{ W m}^{-2}$ in kW m^{-2}

(b) Energy density map at PS10 receiver during a year in MW h m^{-2}

Figure 2: Flux density and energy density maps

- (b) If \mathbf{r}_0 doesn't exist (or is at the sun), the location \mathbf{r}_2 is defined as the first intersection with a solid surface of the ray starting at \mathbf{r}_1 in the direction $\boldsymbol{\omega}_1$
- (i) If \mathbf{r}_2 belongs to something else than the receiver \mathcal{T} , there is a blocking effect and the Monte Carlo weight is $\hat{w} = 0$;
 - (ii) If \mathbf{r}_2 doesn't exist there is a spillage effect and the Monte Carlo weight is $\hat{w} = 0$;
 - (iii) If \mathbf{r}_2 belongs to the receiver \mathcal{T} , the Monte Carlo weight is $\hat{w} = \frac{DNI(t) \times \rho \times (\boldsymbol{\omega}_S(t) \cdot \mathbf{n}_h) \times S_{\mathcal{H}^+}}{p_{\tau}(t)}$

3. Implementation

In order to test the performance of the MCST algorithm, as well as for validation purposes, it is run with the characteristics of a reference Central Receiver System. PS10, a 11 MW_e power plant, has been the subject of several studies from which data are available. It is located near Seville, in Spain, and is the world's first commercial concentrating solar power plant. Its heliostat field consists in 624 heliostats following a radial staggered layout (Siala and Elayeb, 2001). Each heliostat has a surface of 121 m^2 and concentrates sun rays to a receiver, located at the top of a 115 m high tower, that feeds a steam turbine. Each sun position is translated into a DNI value coming from Typical Meteorological Year (TMY) data that correspond to a typical year on the considered site. As we have no better information, when addressing a lifetime prediction, we make the assumption that the climate will be identical every year and the lifetime simulation becomes strictly identical to a yearly simulation. Therefore only yearly simulations will be presented. According to the design, this power

plant should annually produce 23 GWh_e (electrical energy) and about 95 GWh_{th} (thermal energy) (Osuna et al., 2006). With MCST algorithm we have estimated 97 GWh_{th}. This result indicates a very good agreement with the literature data and we retain it as a first MCST-validation exercise. For internal validation, we also performed time discretization using a deterministic approach for shorter time-intervals. This required that a separate Monte Carlo run was performed for each date with MCFD algorithm and that each Monte Carlo reached a high enough accuracy level so that we only tested the time-integration part of the algorithm. Adequation was perfect, which fully validated our statistical integration procedure.

Table 1 presents a comparison of MCST (yearly integration) and MCFD algorithms in terms of computational times and number of required realizations for a relative standard deviation (*%RSD*) of 0.1%, where *%RSD* is defined as

$$\%RSD = \frac{\sigma}{\bar{x}} \times 100 \quad (11)$$

where σ is the standard deviation of Monte Carlo weights and \bar{x} is the average of Monte Carlo weights.

MCFD algorithm performances are presented for March 21st at noon and March 21st at 6PM. Figures [2a; 2b] display sun spots : one for March 21st at noon and one on a yearly integrated basis. The main point is that time integration with MCST requires 26 times more realizations than MCFD for March 21st at 12PM but 11 times less realizations than MCFD for March 21st at 6PM. This simply reflects the fact that these two particular dates are close to the extreme cases. For March 21st at noon, there is nearly no optical losses (blocking, shading and spillage phenomena) and the optical integration is quite simple. On the contrary, at 6PM, optical losses require an accurate optical integration which translates into more Monte Carlo realizations. In any case, this confirms that time integration is not the main source of statistical difficulty in the CSP-design context. We claimed in introduction that annual integration could be performed with computational time comparable to that of a single date simulation. This is not true when comparing with favorable dates but is even an understatement when we consider disadvantageous dates. In appendix, the yearly simulation time of 28 s is reduced to 2.8s thanks to further computer-graphics optimization.

Table 1: Comparison of the computation time for Monte Carlo Fixed Date and Monte Carlo Sun Tracker algorithm for a relative standard deviation *%RSD* = 0.1%

Algorithms	Computation time in second	Number of required realizations ³	Energy in GW h
MCFD			
March 21 12PM	1.3	2407	...
March 21 6PM	33	700 000	...
MCST	28	62 100	97

4. Conclusion

Our main message is that statistical approaches make it possible to address lifetime integrated quantities with the same ease as temporal quantities. There is therefore no need to simplify the physical representation of the concentration process itself. We have illustrated this potential with the PS10 example, but up to now we have not entered into the details of what is now possible using this integration technique. The next step will be to consider phenomena such as :

- optical ageing of CSP components along lifetime extended up to fifty years or longer
- accurate representation of climatic fluctuations (annual DNI change)
- and even representation of predicted climate changes (long-term DNI change)

Acknowledgments

The authors acknowledge financial support from Total New Energies.

Appendix A. Optimization of the computation time

For optimization purposes, two ways are explored in order to reduce the MCST CPU time. To differentiate each code version, the following notation is introduced :

MCST_a the original algorithm (see 2.2)

MCST_b using the first optimization

MCST_c using the second optimization

Appendix A.1. First optimization

Considering that re-orientation (therefore at each date) of heliostats at each realization is time consuming, the aim is here to reduce the number of sample dates. As it is necessary to preserve estimation accuracy, the idea is here to sample several rays per date. This is the concept of systematic sampling (Dunn and Shultis, 2011): N_r rays are sampled and followed onto the field for each sampled date. The corresponding integral formulation is modified and Eq. (6) becomes Eq. (A.1) and MCST_a becomes MCST_b algorithm.

$$E = \int_{Lifetime} p_{\tau}(t) dt \sum_{n=1}^{N_r} \frac{1}{N_r} \int_{\mathcal{D}_{\mathcal{H}^+}} p_{R_1}(r_1) dr \int_{\mathcal{D}_{\Omega_S(t)}} p_{\Omega_S(t)}(\omega_S(t)) d\omega \int_{\mathcal{D}_{N_h}} p_{N_h}(n_h | \omega_S(t); p) dn \hat{w}^{th} \quad (A.1)$$

³A Monte Carlo realization means a single ray

Appendix A.2. Second optimization

The second optimization has nothing to do with the Monte Carlo algorithm (the integral formulation) itself. The integral formulation remains the same as equation Eq(A.1). Here we optimize the photon tracking algorithm using computer graphics techniques. The sun tracking performed during MCST realizations implies that the geometry of the solar installation is updated at each solar position. The re-positioning of the heliostat field and of the bounding boxes associated to each heliostat, for a sun position consists in performing matrix multiplications, each heliostat position being described with a 4×4 matrix (Pharr and Humphreys (2010)). The computation time devoted to this step becomes huge when simulating a large scale central receiver system with hundreds of heliostats. We developed heliostat-reorientation algorithm that reduces this computation time. Each heliostat is enclosed into an enlarged cubic bounding box. This bounding box includes all the positions that an heliostat can take when tracking the sun.

Appendix A.3. Results

Table 2 displays the computation time associated to these two optimizations. The first optimization reduces the variance (the number of required realizations) but increases the computation time. It is therefore rejected. For the second optimization the number of realizations is strictly identical, the integral formulation is indeed unchanged but the computation time is reduced by approximately a factor ten ² for MCST algorithm.

Table 2: Comparison of the computation time of the several versions of Monte Carlo Sun Tracker algorithms for a relative standard deviation $\%RSD = 0.1\%$

Algorithms	Computation time in second	Number of required realizations ³
MCST _a	28	62 100
MCST _b	33	48 200
MCST _c	2.8	62 100

References

P. Schwarzbözl, R. Pitz-Paal, M. Schmitz, Visual HFLCAL - A software tool for layout and optimization of heliostat fields, in: Proceedings of 15th SolarPACES Conference, 1–8, 2009.

²The computation time is given for a desktop PC with AMD Phenom II X6 1055T 2.8 GHz and 12 Go RAM.

³A Monte Carlo realization means a single ray

- P. Gilman, N. Blair, M. Mehos, C. Christensen, S. Janzou, C. Cameron, Solar advisor model user guide for version 2.0, National Renewable Energy Laboratory, 2008.
- P. Garcia, A. Ferrière, J. J. Bézian, Codes for solar flux calculation dedicated to central receiver system applications: A comparative review, *Solar Energy* 82 (3) (2008) 189–197.
- M. Sanchez, M. Romero, Methodology for generation of heliostat field layout in central receiver systems based on yearly normalized energy surfaces, *Solar Energy* 80 (2006) 861–874.
- F. Collado, Quick evaluation of the annual heliostatfield efficiency, *Solar Energy* 82 (2008) 379–384.
- P. K. Falcone, A handbook for solar central receiver design, Tech. Rep., Sandia National Labs., Livermore, CA (USA), 1986.
- J. De La Torre, G. Baud, J. Bézian, S. Blanco, C. Caliot, J. Cornet, C. Coustet, J. Dauchet, M. El Hafi, V. Eymet, R. Fournier, J. Gautrais, O. Gourmel, D. Joseph, N. Meilhac, A. Pajot, M. Paulin, P. Perez, B. Piaud, M. Roger, J. Rolland, F. Veynandt, S. Weitz, Monte Carlo advances and concentrated solar applications, *Solar Energy* 103 (2014) 653–681.
- M. Pharr, G. Humphreys, *Physically Based Rendering, second edition : from theory to implementation*, Morgan Kaufmann Publishers, 2010.
- F. M. F. Siala, M. E. Elayeb, Mathematical formulation of a graphical method for a no-blocking heliostat field layout, *Renewable energy* 23 (1) (2001) 77–92.
- R. Osuna, R. Olavarría, R. Rafael Morillo, Construction of a 11MW Solar Thermal Tower Plant in Seville, Spain, in: *Proceeding of 13 th Solar PACES Symposium*, Seville, Spain, 2006.
- W. L. Dunn, J. K. Shultis, *Exploring Monte Carlo Methods*, Elsevier, 2011.