Available online at www.sciencedirect.com

ScienceDirect

Solar Energy 113 (2015) 57–62

SOLAR ENERGY

www.elsevier.com/locate/solener

Brief Note

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

Received 27 June 2014; received in revised form 15 December 2014; accepted 19 December 2014
Available online 20 January 2015

Communicated by: Associate Editor Robert Pitz-Paal

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 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

* Corresponding author at: Université de Toulouse, École des Mines d'Albi, UMR CNRS 5302, RAPSODEE Research Center, F-81013 Albi, France.
E-mail address: olivier-farges@mines-albi.fr (O. Farges).

http://dx.doi.org/10.1016/j.solener.2014.12.027
0038-092X/

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)
		$\hat{w}$	Monte Carlo weight
$\mathbf{H}()$	the Heaviside step function	$\bar{x}$	average of Monte Carlo weights
$\boldsymbol{n}_1$	ideal normal at $\boldsymbol{r}_1$	Ω_S	solar cone (sr)
$\boldsymbol{n}_h$	effective normal at $\boldsymbol{r}_1$ around the ideal normal $\boldsymbol{n}_1$	$\boldsymbol{\omega}_1$	direction after reflexion (rad)
N_r	number of rays sampled for a date	$\boldsymbol{\omega}_S$	direction inside the solar cone (rad)
P	power (kW)	ρ	mirror reflectivity
$\boldsymbol{r}_i$	location	σ	standard deviation of Monte Carlo weights

quite complex integral over optical-paths and geometry) does not change the overall complexity level and similar numbers of statistical samples are required: addressing life-time 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 (Section 2.2), and is only extended to lifetime in Section 2.1. The last paragraph (Section 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.

(1) A location $\boldsymbol{r}_1$ 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 $\boldsymbol{n}_h$ is sampled around the ideal normal vector $\boldsymbol{n}_1$ at $\boldsymbol{r}_1$ representing reflection and pointing imperfections. $\boldsymbol{\omega}_1$ corresponds to the specular reflection of $-\boldsymbol{\omega}_S$ by a surface normal to $\boldsymbol{n}_h$.
(4) $\boldsymbol{r}_0$ is defined as the first intersection with a solid surface of the ray starting at $\boldsymbol{r}_1$ in the direction $\boldsymbol{\omega}_S$.
   (a) If $\boldsymbol{r}_0$ 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 $\boldsymbol{r}_0$ does not exist (or is at the sun), the location $\boldsymbol{r}_2$ is defined as the first intersection with a solid surface of the ray starting at $\boldsymbol{r}_1$ in the direction $\boldsymbol{\omega}_1$.
      (i) If $\boldsymbol{r}_2$ 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 $\boldsymbol{r}_2$ does not exist there is a spillage effect and the Monte Carlo weight is $\hat{w} = 0$.
      (iii) If $\boldsymbol{r}_2$ belongs to the receiver $\mathcal{T}$, the Monte Carlo weight is $\hat{w} = \mathrm{DNI} \times \rho \times (\boldsymbol{\omega}_S \cdot \boldsymbol{n}_h) \times S_{\mathcal{H}^+}$.

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

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