HAL
open science

# Simulation des performances annuelles d'une installation à énergie solaire concentrée

Olivier Farges, Jean-Jacques Bézian, Mouna El-Hafi, Hélène Bru

## To cite this version:

Olivier Farges, Jean-Jacques Bézian, Mouna El-Hafi, Hélène Bru. Simulation des performances annuelles d'une installation à énergie solaire concentrée. Congrès Français de Thermique, Société Française de Thermique, May 2013, Gerardmer, France. hal-01164584

**HAL Id: hal-01164584**

**https://imt-mines-albi.hal.science/hal-01164584v1**

Submitted on 17 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.

# Simulation des performances annuelles d'une installation à énergie solaire concentrée

**Olivier FARGES[1*], Jean-Jacques BÉZIAN[2], Mouna EL HAFI[2] et Hélène BRU[3]**

[1] Total énergies Nouvelles et Centre RAPSODEE, UMR CNRS 5302,
Campus Jarlard, Route de Teillet, 81013 Albi Cedex 09, France
[2] Centre RAPSODEE, UMR CNRS 5302, école des Mines d'Albi-Carmaux, 81013 Albi Cedex 09, France
[3] Total énergies Nouvelles, R&D - Concentrated Solar Technologies, Tour Michelet, 24 cours Michelet, 92069 Paris La Défense Cedex, France
*(auteur correspondant : olivier.farges@mines-albi.fr)

**Résumé -** Nous avons développé un code de calcul basé sur les méthodes de Monte Carlo permettant de simuler l'énergie moyenne annuelle reçue à l'entrée du récepteur d'une installation à énergie solaire concentrée. Ce code permet, lors du processus de conception d'une telle installation, de prendre en considération les positions du soleil sur une année. Implémenté dans l'environnement de développement EDStaR, ce code rapide et précis vise à être intégré dans un schéma d'optimisation pour permettre le design de systèmes solaires à récepteur central ayant des performances annuelles maximisées.

**Nomenclature**

*Symboles latins*

$A$ énergie annuelle moyenne, $\mathrm{Wh}$
$B$ Blockage, $\%$
$\mathcal{C}$ Cible (l'indice $+$ désigne le coté actif)
$CSP$ Énergie Solaire Concentrée (Concentrated Solar Power)
$DNI$ Densité de flux normale incidente, $\mathrm{W \cdot m^{-2}}$
$f_r$ Fonction de distribution bidirectionnelle de réflectance (BRDF)
$h$ Angle horaire, $\mathrm{rad}$
$\mathcal{H}$ Surface d'héliostats (l'indice $+$ désigne le coté réfléchissant)
$Sh$ Ombrage, $\%$
$pdf$ Fonction de densité de probabilité
$Sp$ Pertes par débordement, $\%$
$\boldsymbol{n_1}$ Normale idéale en $\boldsymbol{x_1}$
$\boldsymbol{n_h}$ Normale effective en $\boldsymbol{x_1}$ autour de la normale idéale $\boldsymbol{n_1}$
$t$ Temps, $\mathrm{s}$
$\hat{w}_i$ Poids de Monte Carlo, $\mathrm{Wh}$
$\boldsymbol{x_i}$ Point sur la géométrie

*Symboles grecs*

$\gamma$ Jour de l'année
$\eta$ Heure, $\mathrm{h:min:s}$
$\Omega_S$ Cône solaire, $\mathrm{sr}$
$\boldsymbol{\omega_S}$ Direction dans le cône solaire, $\mathrm{rad}$
$\boldsymbol{\omega_1}$ Direction après réflexion, $\mathrm{rad}$

## 1. Introduction

Du fait de l'investissement considérable demandé par la construction d'une centrale solaire à récepteur central, les performances de ce type d'installations doivent être maximisées. Dans cette optique, la phase préliminaire de design revêt un caractère fondamental. De nos jours, la simulation numérique est abondamment mise à contribution lors de cette étape cruciale. Plusieurs outils de simulations, largement utilisés et bien connus dans la communauté « solaire concentré », se sont développés avec la filière. Nous pouvons répartir ces logiciels en deux catégories :

- Les logiciels de lancer de rayons (Ex : Tonatiuh, Soltrace, SolFast, ...) : ils ne prennent en compte qu'une position du soleil, ce qui rend, dans la pratique, impossible l'évaluation des performances annuelles ;
- Les logiciels d'analyse technico-économique (Ex : SAM, Greenius, ...) : ils calculent un productible annuel par l'utilisation de rendements optiques globaux ;

Nous proposons ici une nouvelle approche basée sur les méthodes de Monte Carlo permettant d'intégrer annuellement la puissance instantanée reçue à l'entrée du récepteur d'une centrale solaire. Pour cela nous prenons en compte les positions que prend le soleil tout au long d'une année et la puissance associée. Nous obtenons ainsi, avec un temps de simulation acceptable, une estimation précise de l'énergie thermique collectée annuellement. De part sa vitesse d'exécution, ce code peut s'intègrer dans une boucle d'optimisation de type méthodes d'optimisation stochastique (optimisation par essaim de particules, algorithmes génétiques, ...).

## 2. Outil de simulation

EDStaR (Environnement de Développement en STAtistiques Radiatives) [1] est un environnement de développement maintenu par le groupe StarWest [2]. Ce groupe est constitué de physiciens spécialisés notamment dans les transferts radiatifs. Les domaines d'applications concernés sont essentiellement :

- L'optimisation radiative des centrales solaires à concentration ;
- La paramétrisation du rayonnement au sein de l'atmosphère terrestre dans le cadre de l'étude du changement climatique ;
- Tout autre champ d'applications en rapport avec des phénomènes de transport (imagerie médicale, morphogénèse auto-organisée et intelligence collective en biologie, transferts thermiques à micro-échelle en écoulements diphasiques, ...)

Utilisant les méthodes de Monte Carlo [3] [4], il tire avantage des techniques avancées de rendu issues de la communauté de synthèse d'images : les géométries complexes sont traitées grâce à la bibliothèque C++ conçue dans le cadre du projet « Physically Based Rendering Techniques » (PBRT) [5]. Cette bibliothèque PBRT est combinée à la bibliothèque C++ *mcm* qui se charge de la programmation d'algorithmes de Monte Carlo, incluant notamment l'estimation de sensibilités et le calcul de barres d'erreur. Enfin, la bibliothèque *GNU Scientific Library (GSL)* [6] est utilisée pour la génération de nombres aléatoires. Elle contient plusieurs générateurs aléatoires assurant le même niveau d'uniformité et d'indépendance. Ces trois bibliothèques sont combinées dans l'environnement de développement EDStaR. De plus, ce dernier tire avantage des possibilités informatiques actuelles telles que la parallélisation massive et l'accélération de lancer de rayons en géométrie complexe [7].

## 3. Algorithme de Monte Carlo et suivi solaire

Comme indiqué dans la section 1. nous avons bâti un code suivant la course du soleil durant l'année. L'environnement EDStaR nous permet de construire un algorithme de Monte Carlo spécifique intégrant les positions du soleil appliqué à la simulation d'installations solaires à récepteur central.

### 3.1. Positions du soleil et valeurs de DNI

La prise en compte des positions du soleil dans un algorithme de Monte Carlo entraîne la nécessité de mettre à jour la géométrie de l'installation solaire à chaque position. En effet, l'orientation du champ d'héliostats est dépendante de ce paramètre. Nous devons de plus trouver une représentation du soleil capable de rendre compte de son déplacement annuel. Pour un lieu sur le globe, c'est à dire une couple *longitude-latitude*, la localisation du soleil peut être défini par un couple d'angles (azimut et élévation, azimut et angle zénithal, déclinaison et angle horaire, ...) ou par une « date » (jour, heure, minute, seconde) [8]. Ces représentations étant équivalentes, nous avons fait le choix de travailler avec des dates où $\gamma$ représente un jour entre $[1; 365]$ et $\eta$

une heure entre $[6AM; 18PM]$ [a]. Pour une localisation donnée et une période d'étude, chaque position du soleil correspond à une valeur de DNI suivant un fichier de données météo associé. Nous avons besoin de données de DNI pour chaque instant alors que la plupart des fichiers météo utilisés en simulation solaire ne fournissent que des moyennes horaires. Une interpolation de ces données est par conséquent nécessaire et constitue une approximation suffisamment précise.

## 3.2. Algorithme « Monte Carlo Sun Tracking »

### *3.2.1. Formulation mathématique*

L'algorithme *Monte Carlo Sun Tracking* (« MCST ») implémenté dans ce code prend en compte les positions du soleil. Nous introduisons une intégration sur le temps afin d'obtenir une valeur d'énergie (en $\mathrm{Wh}$) au lieu d'une puissance (en $\mathrm{W}$). Le but de cette simulation est d'évaluer l'énergie annuelle collectée par le récepteur de la centrale. Cette énergie $A$ est exprimée par l'équation (1). L'intégrale sur le temps est représentée par deux intégrales sur le jour et l'heure.

$$\begin{aligned} A = \int_{1}^{365} \mathrm{d}\gamma \int_{6\ A.M.}^{6\ P.M.} \mathrm{d}\eta \int_{H^+} \mathrm{d}x \int_{\Omega_{sun}} \mathrm{d}\omega_S \times [H(\boldsymbol{x_0} \notin (\mathcal{H} \cup \mathcal{C})) \\ \times \int_{2\pi} \mathrm{d}\boldsymbol{\omega_1} f_r(\boldsymbol{\omega_1}|\boldsymbol{\omega_S}, \boldsymbol{x_1}, \nu)\, |\boldsymbol{n}(\boldsymbol{x_1}) \cdot \boldsymbol{\omega_1}| \times \Big[H(\boldsymbol{x_2} \in \mathcal{C}^+)\Big]\Big] \times DNI(t) \end{aligned} \tag{1}$$

Le modèle « MCST » évalue également les différentes pertes optiques se produisant dans un système CSP à récepteur central. Elles sont illustrées sur le schéma de principe d'une centrale solaire à récepteur central de la figure (1).

**L'ombrage :** une surface se trouve entre le soleil et un héliostat ;

**Le blocage :** une surface bloque le rayon réfléchi par l'héliostat avant son impact avec la cible ;

**Les pertes par débordement :** les rayons réfléchis venant des héliostats franchissent le plan contenant la cible sans impacter cette dernière ;

### *3.2.2. Formulation algorithmique*

L'évaluation de ces variables par les méthodes de Monte Carlo nécessite l'introduction de fonctions de densité de probabilité (pdf). Ces dernières, lorsqu'elles sont bien choisies en fonction de la physique du problème [9], permettent d'améliorer la convergence de l'algorithme tout en réduisant la variance de chaque variable. En introduisant le poids de Monte Carlo et les pdfs, nous obtenons l'équation (2) à partir de l'équation (1).

$$\begin{aligned} A = \int_{\Delta} p_{\Delta}(\delta)\mathrm{d}\delta \int_{H} p_H(\eta)\mathrm{d}\eta \int_{\mathcal{H}^+} p_{X_1}(\boldsymbol{x_1})\mathrm{d}\boldsymbol{x} \int_{\Omega_S} p_{\Omega_S}(\boldsymbol{\omega_S})\mathrm{d}\boldsymbol{\omega_S} \times \Bigg\{ H(\boldsymbol{x_0} \notin (\mathcal{H} \cup \mathcal{C}) \\ \times \int_{D_{N_h}} p_{N_h}(\boldsymbol{n_h}|\boldsymbol{\omega_S}; p)\mathrm{d}\boldsymbol{n_h} \times \{H(\boldsymbol{x_2} \in \mathcal{C})\hat{w}_A\} \Bigg\} \end{aligned} \tag{2}$$

a. Échantillonner sur 24 heures n'a pas de sens donc nous nous concentrons sur les heures de fonctionnement d'une installation solaire, ici pour un lieu de latitude $0°$. Pour un lieu ne se trouvant pas sur l'équateur, la plage d'échantillonnage sera modifiée pour couvrir le jour le plus long de l'année

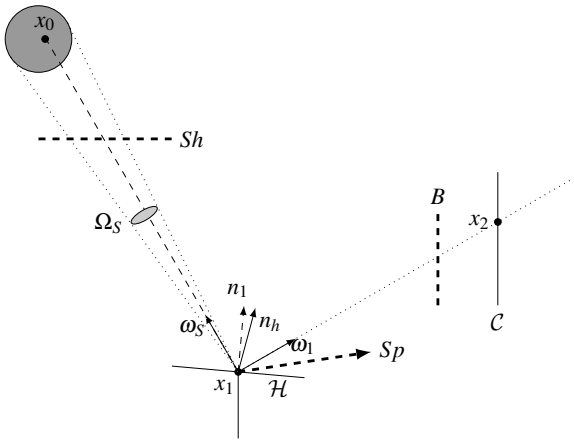


Figure 1 *Schéma de principe d'une installation solaire à récepteur central*

Les pdfs sont définies par les équations (3), (4), (5), (6), (7). La fonction de densité de probabilité $p_{\boldsymbol{n}_h}$ (Eq.7) implémente le modèle de Blinn [5] utilisé en synthèse d'image. Son rôle est de définir une normale apparente autour de la normale théorique, permettant ainsi de prendre en compte les erreurs de réflexion générées par l'ensemble des erreurs optiques (erreur de position, erreur de courbure, erreur de spécularité). Lorsqu'un rayon n'est pas perturbé et qu'il impacte le récepteur, ce rayon contribue à la grandeur d'intérêt $A$. Cette contribution prend la forme d'un poids de Monte Carlo qui, dans notre cas est présentée dans l'équation (8).

$$p_{\Delta}(\delta) = \frac{1}{365} \tag{3}$$

$$p_H(\eta) = \frac{1}{12} \tag{4}$$

$$p_{X_1}(\boldsymbol{x_1}) = \frac{1}{S_{\mathcal{H}^+}} \tag{5}$$

$$p_{\Omega_S}(\boldsymbol{\omega_S}) = \frac{1}{2\pi(1-\cos\theta_S)} \tag{6}$$

$$p_{\boldsymbol{n}_h}(\boldsymbol{n}_h|\boldsymbol{\omega}_S;p) = \frac{2+\dfrac{1}{p}}{2\pi\left(1-\mu(\boldsymbol{\omega}_S)^{2+\frac{1}{p}}\right)} \tag{7}$$

$$\hat{w}_A = \frac{I_{sun}(\boldsymbol{\omega_S}\cdot\boldsymbol{n_h})}{p_{\Omega_S}(\boldsymbol{\omega_S})p_{X_1}(\boldsymbol{x_1})} = DNI(t)(\boldsymbol{\omega_S}\cdot\boldsymbol{n_h})S_{\mathcal{H}^+} \tag{8}$$

La formulation intégrale présentée dans les équations précédentes (Éqs. (2) à (8)) est directement interprétée comme l'algorithme (1). Cet algorithme présente également la version *Multi Rays* de notre code présentée à la section 4.3. (désigné par l'acronyme « MRMCST » dans la suite de ce texte).

```
Energie A = 0
pour chaque réalisation faire
    Échantillonnage uniforme de δ dans [0, 365]
    Échantillonnage uniforme de η dans [6A.M., 6P.M.]
    pour i = 1 a i = N faire [b]
        Échantillonnage uniforme de x_1 sur H^+
        Échantillonnage uniforme de ω_0 dans le disque solaire Ω_S
        Génération de n_h selon le modèle de Blinn
        si Pas d'ombrage entre le soleil et x_1 alors
            ŵ_A = DNI(t)|n_h · ω_S|S_{H^+}
        sinon
            ŵ_A = 0
            arrêt
        fin
        Réflexion spéculaire ω_1 = ω_S + 2|n_h · ω_S|n_h
        x_2 = intersection entre Rayon(x_1, ω_1) et un élément de la géométrie
        si x_2 existe alors
            si x_2 ∈ C^+ alors
                A = A + ŵ_A
            sinon
                ŵ_A = 0
                arrêt
            fin
        sinon
            ŵ_A = 0
            arrêt
        fin
        i = i + 1
    fin
fin
```

**Algorithme 1:** Algorithme Monte Carlo Sun Tracking (MCST)

# 4. Validation

## 4.1. Comparaison avec Tonatiuh à une date fixée

Pour valider notre approche, nous avons réalisé une série de simulations avec Tonatiuh [10], considéré comme un outil de simulation de référence dans la communauté « solaire concentré ». Nous avons établi un cas test représenté par une centrale à tour composée de 146 héliostats. Le champ d'héliostats est construit suivant la méthode MUEEN [11] pour obtenir un agencement radial étagé. Chaque héliostat comprend 9 miroirs carrés de 1,6 $\mathrm{m}$ de coté. Nous avons également posé certaines hypothèses simplificatrices :

- Les réflexions sont supposées spéculaires ;
- L'installation est située sur l'équateur ;
- La cible est un carré de 4 $\mathrm{m}$ de coté ;

La première étape de validation consiste en la simulation de ce cas test pour des dates fixes, sans introduire le suivi du soleil et l'intégration annuelle. Cela doit nous assurer que Tonatiuh

b. Pour MCST $N = 1$, pour MRMCST $N = 100$

et EDStaR donnent des résultats similaires. Nous avons choisi 4 dates avec un DNI de 1000 $\mathrm{Wm}^{-2}$ et un tirage de 1 000 000 de rayons :

- L'équinoxe de printemps : le 21 Mars à midi [c] ;
- Le solstice d'été : le 21 Juin à midi [c] ;
- L'équinoxe d'automne : le 22 Septembre à midi [c] ;
- Le solstice d'hiver : le 21 Décembre à midi [c] ;

Les résultats de ces simulations sont présentés dans le tableau (1). Nous pouvons voir qu'ils sont égaux jusqu'au troisième chiffre significatif, ce qui permet de conclure à l'excellente concordance entre Tonatiuh et EDStaR.

| | Tonatiuh | EDStar |
|---|---|---|
| | MW | MW |
| Équinoxe de printemps | $2,97$ | $2,97$ |
| Solstice d'été | 3.19 | $3,19$ |
| Équinoxe d'automne | $2,98$ | $2,98$ |
| Solstice d'hiver | $3,19$ | $3,19$ |

Tableau 1 *Puissance reçue au récepteur par dates*

### 4.2. Comparaison pour 50 dates

Nous cherchons ici à valider le résultat obtenu avec le code *MCST*, prenant en compte l'intégration temporelle. Nous générons aléatoirement 50 dates. Nous réalisons une simulation de la configuration pour chaque date avec Tonatiuh. Afin d'avoir une bonne précision, nous utilisons 1 000 000 de rayons. Nous obtenons ainsi 50 valeurs de puissance reçue par le récepteur. Nous supposons que la puissance moyenne de ces 50 dates est représentative de l'énergie moyenne reçue à chaque instant de l'année. Nous réalisons la simulation des mêmes dates avec EDStaR en un seul calcul. Nous avons donc un rayon pour chaque date, soit 50 rayons. Les différents résultats sont répertoriés dans le tableau (2). Nous pouvons voir que la valeur estimée en une seule simulation par EDStaR est très proche de la moyenne des 50 simulations réalisées avec Tonatiuh. Nous pouvons remarquer que la barre d'erreur fournie par l'algorithme de Monte Carlo est assez large. Cependant, avec seulement 50 valeurs, nous nous attendions à cette tendance. Nous verrons au 4.3. que la barre d'erreur se resserre drastiquement avec un nombre de dates supérieur. Nous pouvons conclure ici que EDStaR donne une estimation de l'énergie annuelle reçue par le récepteur d'une centrale solaire en accord avec les simulations réalisées date par date avec Tonatiuh.

| | Tonatiuh | EDStar |
|---|---|---|
| | $\mathrm{GWh}_{th}$ | $\mathrm{GWh}_{th}$ |
| 50 dates | $6,329$ | $6,323 \pm 0,623$ |

Tableau 2 *Comparaison Tonatiuh - EDStaR pour 50 dates*

c. Heure du soleil

### 4.3. Simulation annuelle et version « Multi Rays »

Dans le but d'obtenir un résultat plus précis, c'est à dire avec des barres d'erreur plus étroites, et ainsi avoir une confiance plus grande dans nos simulations, nous allons devoir considérer un plus grand nombre de dates. En parallèle, nous avons établi une version « Multi Rays » de notre algorithme *MCST*. En effet, alors que la réorientation du champ d'héliostats entre chaque date se trouve être la phase de simulation la plus gourmande en tant de calcul, nous utilisons plusieurs rayons pour une date au lieu d'un seul. L'algorithme (1) présente cette variante pour $N = 100$ rayons. Nous réalisons des simulations de notre cas test pour 1 rayon, 100 rayons et 1000 rayons. Les résultats sont présentés dans le tableau (3). Il apparaît que les barres d'erreur sont très proches pour un nombre de réalisations donné, indépendamment du nombre de rayons utilisés par date. Ceci s'explique par le fait que la contribution de chaque rayon impactant le récepteur pour une date donnée varie très peu. De plus, la configuration du champ obtenue par la méthode MUEEN réduit fortement les phénomènes d'ombrage et de blockage du champ. Le nombre de rayons ne contribuant pas à la grandeur estimée est de ce fait très faible (moins de $5\%$).

Le code « MRMCST » trouve tout son intérêt pour la réalisation de cartes de flux *spatio-temporelles*. Il offre une meilleure précision par comparaison avec le code « MCST » pour chaque pixel d'une carte de flux en raison du plus grand nombre de rayons impactant le récepteur pour une date donnée. Ces cartes de flux présentent la répartition de l'énergie reçue par le récepteur, permettant de tirer des enseignements sur le viellissement des récepteurs de centrale solaire.

| | 1 rayon | 100 rayons | 1000 rayons |
|---|---|---|---|
| | $\mathrm{GWh}_{th}$ | $\mathrm{GWh}_{th}$ | $\mathrm{GWh}_{th}$ |
| 50 000 | $6.39 \pm 4,64 \cdot 10^{-2}$ | $6.38 \pm 4.61 \cdot 10^{-2}$ | $6.37 \pm 4.59 \cdot 10^{-2}$ |
| 500 000 | $6.37 \pm 9,65 \cdot 10^{-3}$ | $6.36 \pm 9,58 \cdot 10^{-3}$ | $6.36 \pm 9,51 \cdot 10^{-3}$ |
| 5 000 000 | $6.36 \pm 4,66 \cdot 10^{-3}$ | $6.36 \pm 4.6 \cdot 10^{-3}$ | $6.36 \pm 4.58 \cdot 10^{-3}$ |

Tableau 3 *Comparaison entre les algorithmes MCST et MRMCST*

### 4.4. Temps de calcul

Comme précisé dans la partie 1., nous souhaitons utiliser ce code dans le cadre d'une optimisation. Cela implique qu'un grand intérêt doit être porté au temps de calcul. Nous avons donc comparé les temps de simulation [d] sur la base d'un nombre de réalisations équivalent. Nous rappellons ici que le logiciel Tonatiuh réalise des simulations pour une date donnée alors que le code « MCST » et sa version « Multi Rays » permettent l'évaluation de grandeurs annuelles. Pour Tonatiuh, une réalisation correspond à un rayon. Pour les codes « MCST » et « MRMCST », une réalisation correspond à une date, soit 1 rayon pour « MCST » et 100 rayons pour « MRMCST 100 ». Le tableau (4) rassemble les différents temps de calcul correspondant à ces 3 codes.

d. Simulations réalisées sur un ordinateur de bureau équipé d'un CPU AMD Phenom II X6 1055T 2.8GHz et de 4Mo RAM

| | Tonatiuh | MCST | MRMCST 100 |
|---|---|---|---|
| | s | s | s |
| 50 000 | $\simeq 3$ | 1 | 22 |
| 500 000 | $\simeq 24$ | 5 | 173 |
| 5 000 000 | $\simeq 156$ | 43 | 1211 |

Tableau 4 *Comparaison des temps de calcul des différents codes*

## 5. Conclusion

Nous avons développé un code de simulation pour centrale solaire à récepteur central dans l'environnement de développement EDStaR. Ce code présente la particularité de prendre en compte les positions du soleil tout au long d'une année et ainsi d'intégrer les puissances instantanées reçues au récepteur pour estimer une énergie moyenne annuelle. L'utilisation des méthodes de Monte Carlo nous permet d'obtenir une solution avec un intervalle de confiance. Au vu de la rapidité du code, comme envisagé, nous pouvons l'intégrer dans une boucle d'optimisation lors du design préliminaire d'une telle installation.

### Références


[1] J. De La Torre and al., Monte Carlo advances and concentrated solar applications, *Solar Energy*, 2013

[2] STARWest (Statistiques Radiatives du Sud-Ouest), *http ://www.starwest.ups-tlse.fr/*.

[3] N. Metropolis and S. Ulam, The monte carlo method, *Journal of the American Statistical Association*, 1949,44, 335–341.

[4] J. M. Hammersley and D. C. Handscomb, Monte Carlo Methods, *Chapman and Hall*, 1969.

[5] M. Pharr and G. Humphreys, Physically Based Rendering, second edition : from theory to implementation, *Morgan Kaufmann Publishers*, 2010.

[6] GSL - GNU Scientific Library, *http ://www.gnu.org/software/gsl*.

[7] The Message Passing Interface (MPI) standard, *http ://www.mcs.anl.gov/research/projects/mpi*.

[8] I. Reda and A. Andreas, Solar position algorithm for solar radiation applications, *Solar energy*, Elsevier, 2004, 76, 577-589.

[9] A. De Lataillade and S. Blanco and Y. Clergent and J. L. Dufresne and M. El Hafi and R. Fournier, Monte Carlo method and sensitivity estimations, *Journal of Quantitative Spectroscopy and Radiative Transfer*, 2002, 75,529–538.

[10] M. J. Blanco and J. M. Amieva and A. Mancilla, The Tonatiuh Software Development Project : An open source approach to the simulation of solar concentrating systems *Proc. of the ASME Computers and Information in Engineering Division*, 2005, 157–164.

[11] F. M. F. Siala and M. E. Elayeb,Mathematical formulation of a graphical method for a no-blocking heliostat field layout, *Renewable energy*, 2001, 23, 77–92.