

# SOLFAST, outil de simulation du rayonnement dans les installations solaire

C. Coustet, B. Piaud, R. Lecussan



## RayBOOSTER© (lancer de rayons)

- simulation de la performance énergétique des bâtiments (ArchiWIZARD, HPC-SA)
- synthèse d'image (ARTLANTIS, ABVENT)
- simulation optique et électromagnétique (ONERA)

**HPC/SA** RAYTRACING  
SOLUTIONS

RayBOOSTER© (lancer de rayons)

- simulation de la performance énergétique des bâtiments (ArchiWIZARD, HPC-SA)
- synthèse d'image (ARTLANTIS, ABVENT)
- simulation optique et électromagnétique (ONERA)

Transfert Radiatif  
Méthode MC



Partenaire de  
valorisation

PROMES-CNRS  
LAPLACE (Univ. de Toulouse)  
RAPSODEE-EMAC



## RayBOOSTER© (lancer de rayons)

- simulation de la performance énergétique des bâtiments (ArchiWIZARD, HPC-SA)
- synthèse d'image (ARTLANTIS, ABVENT)
- simulation optique et électromagnétique (ONERA)

Transfert Radiatif  
Méthode MC



Partenaire de  
valorisation

PROMES-CNRS  
LAPLACE (Univ. de Toulouse)  
RAPSODEE-EMAC

Parteneriat

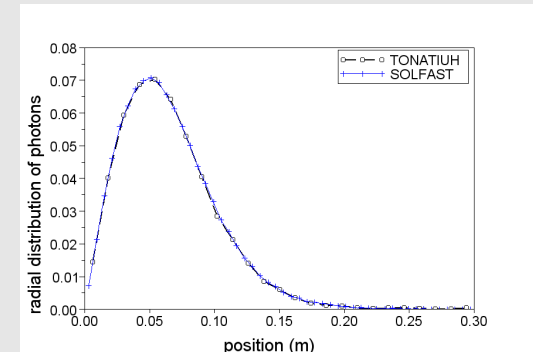
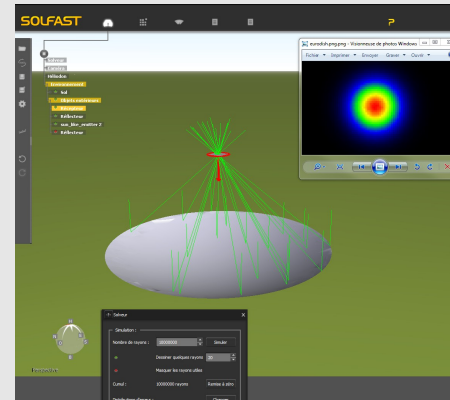


Développement SOLFAST  
(SOLar FACility Simulation Tools)

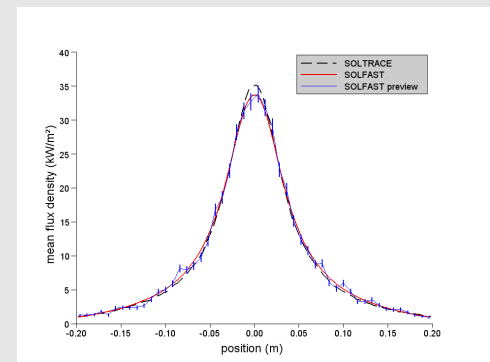
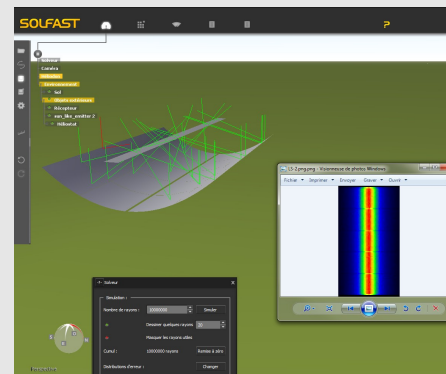
# SOLar FAcility Simulation Tools

SOLFAST est un logiciel de simulation du flux solaire concentré basé sur une méthode Monte-Carlo/Ray-Tracing

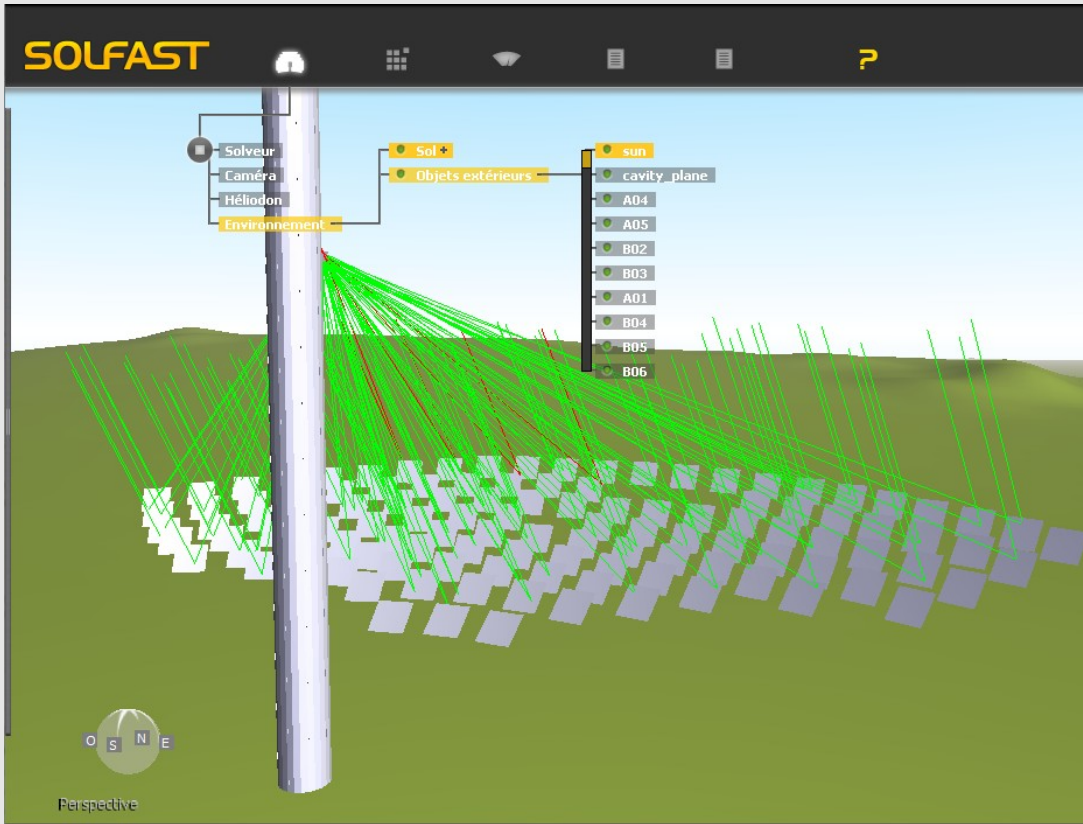
→ Développer un logiciel intégrant les techniques Monte-Carlo sous formulation intégrale



Eurodish (SOLFAST/TONATIUH)

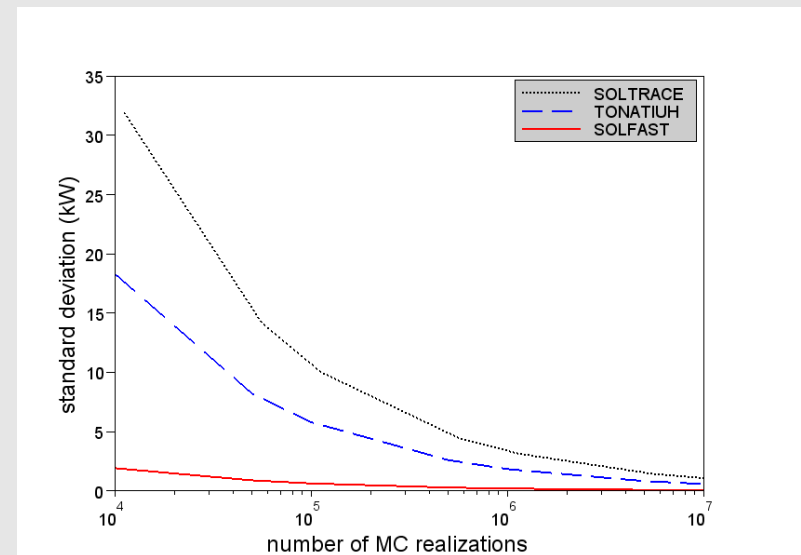
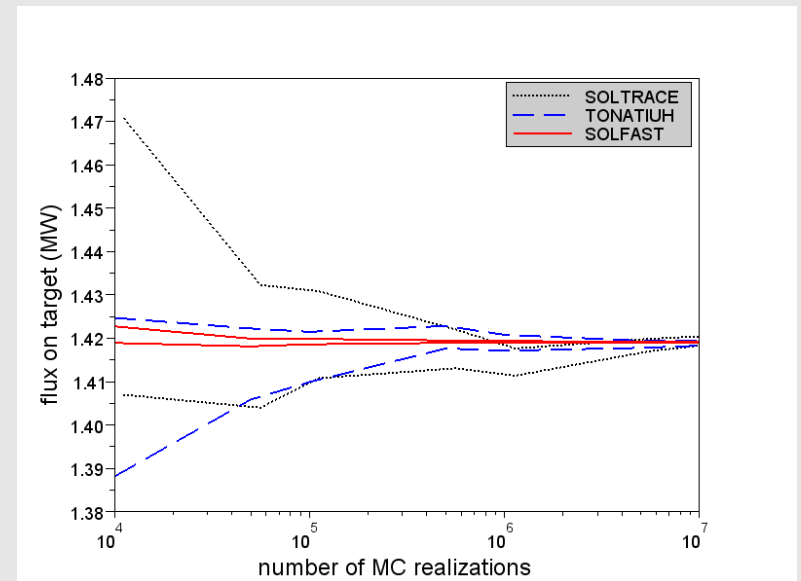


Cylindro-parabolique LS-2 (SOLFAST/SOLTRACE)

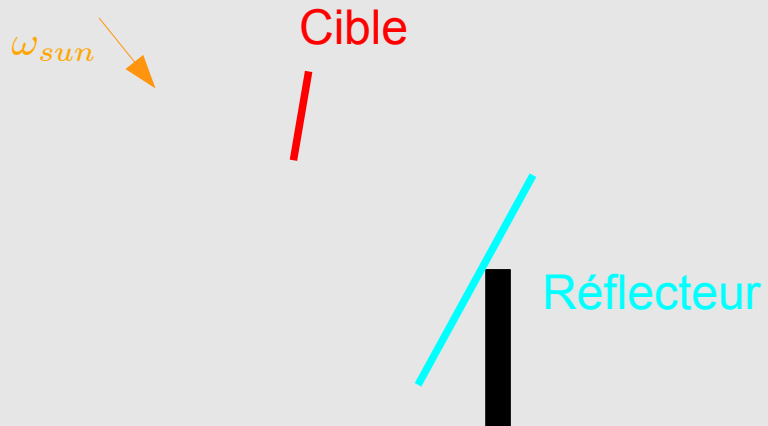
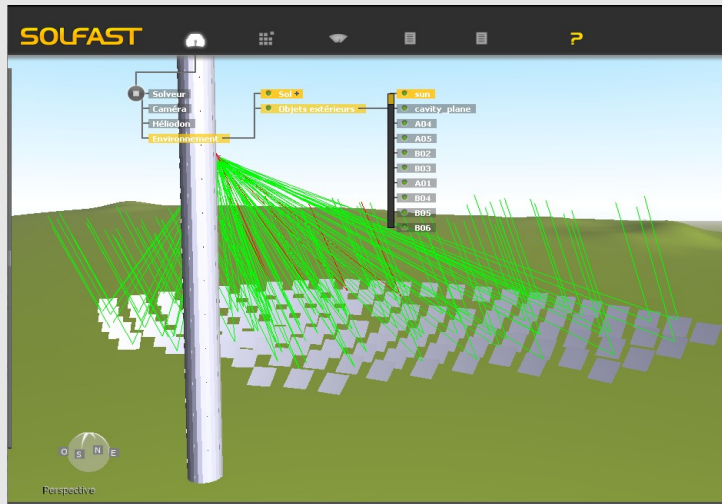


Solfast	Tonatiuh	Soltrace
1.2	241	497

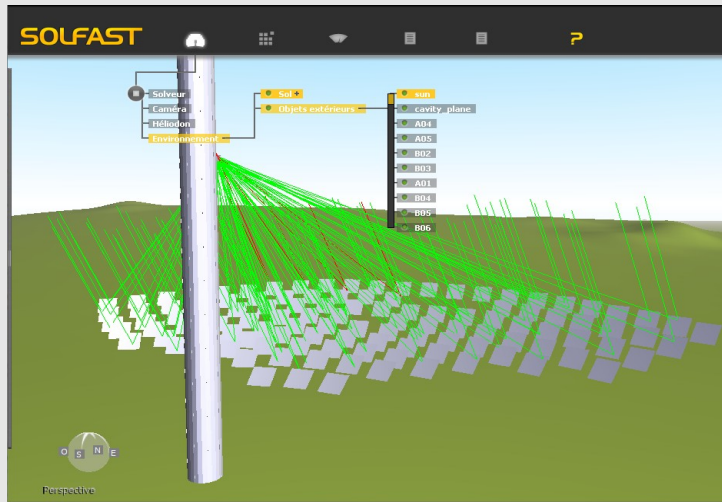
Temps de calcul pour obtenir une barre d'erreur de 500 W



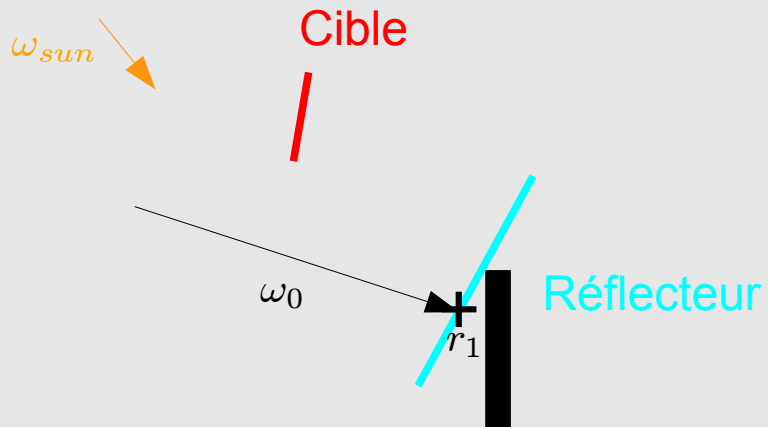
# Algorithme Monte-Carlo



# Algorithme Monte-Carlo

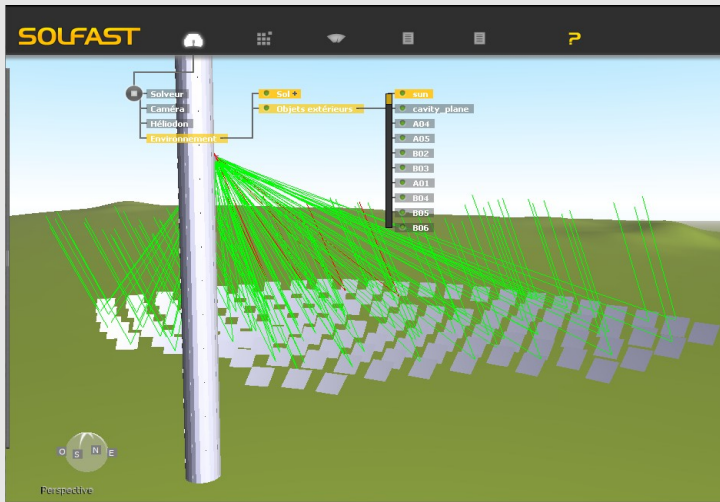


1. Génération de  $r_1$  et d'une direction  $\omega_0$  dans le cône solaire. Initialisation du poids:  $w_i = \rho DNI S_R |\omega_0 \cdot n_1|$

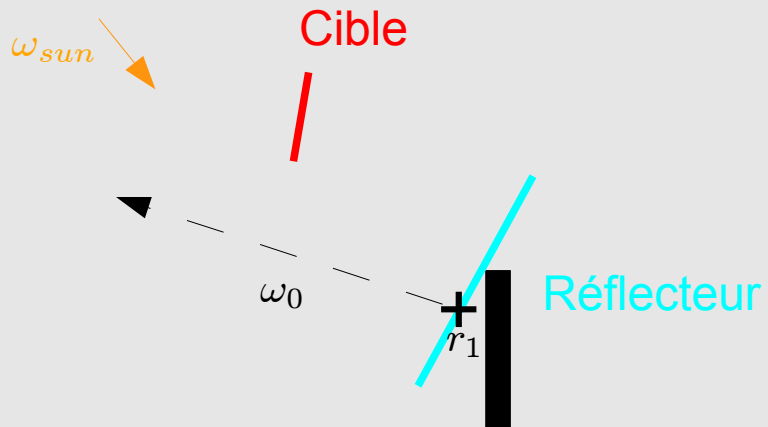




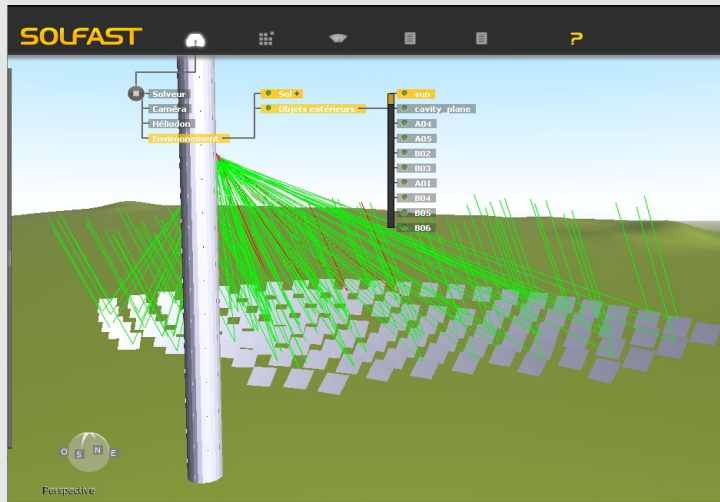
# Algorithme Monte-Carlo



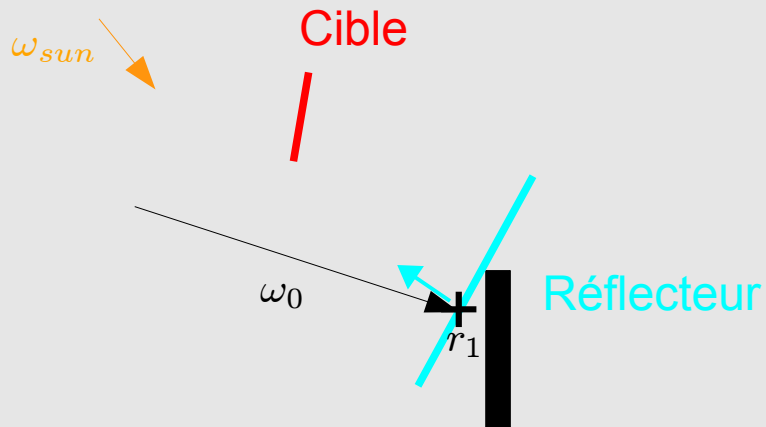
1. Génération de  $r_1$  et d'une direction  $\omega_0$  dans le cône solaire. Initialisation du poids:  $w_i = \rho DNI S_R |\omega_0 \cdot n_1|$
2. Si  $r_1$  n'appartient pas à un réflecteur (test d'ombrage) :  $w_i = 0$  et **FIN**.



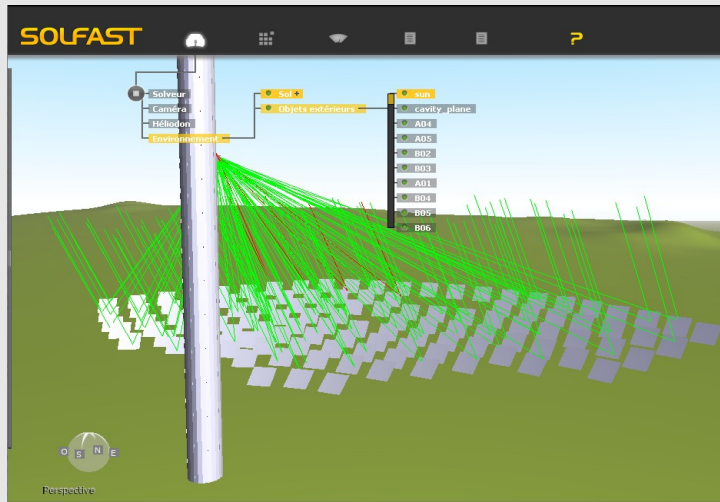
# Algorithme Monte-Carlo



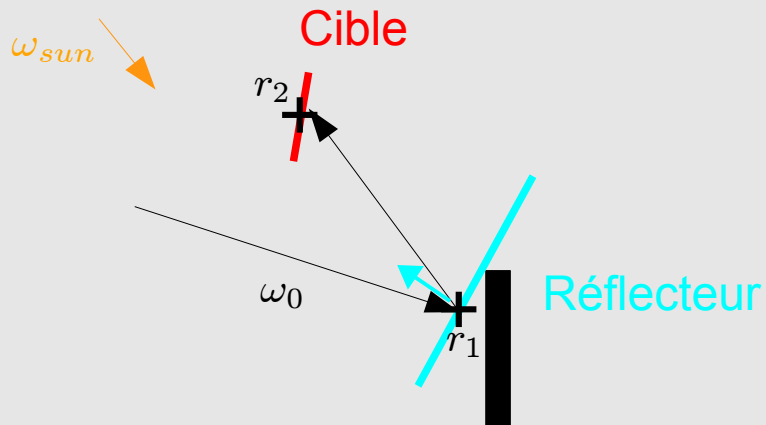
1. Génération de  $r_1$  et d'une direction  $\omega_0$  dans le cône solaire. Initialisation du poids:  $w_i = \rho DNI S_R |\omega_0 \cdot n_1|$
2. Si  $r_1$  n'appartient pas à un réflecteur (test d'ombrage) :  $w_i = 0$  et **FIN**.
3. Tirage d'une normale (modèle de microfacettes). Réflexion spéculaire.



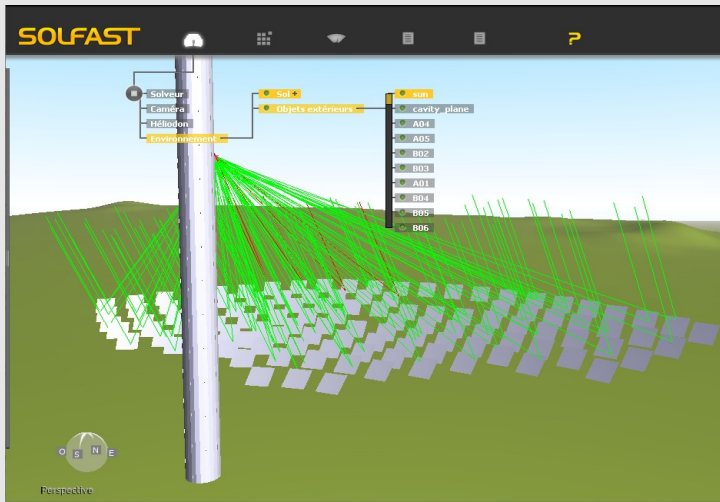
# Algorithme Monte-Carlo



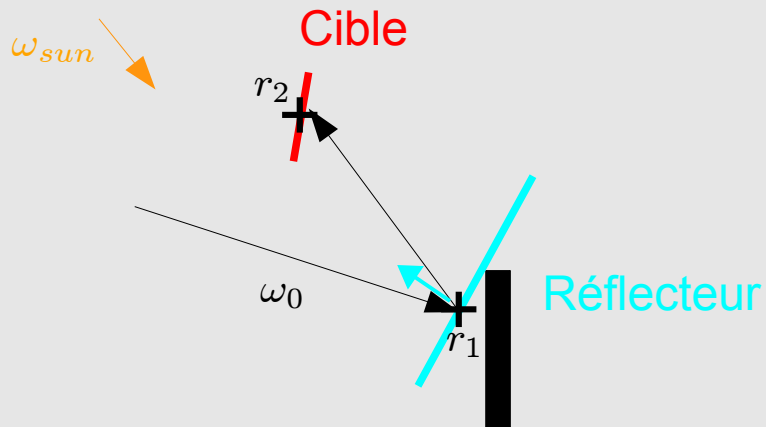
1. Génération de  $r_1$  et d'une direction  $\omega_0$  dans le cône solaire. Initialisation du poids:  $w_i = \rho DNI S_R |\omega_0 \cdot n_1|$
2. Si  $r_1$  n'appartient pas à un réflecteur (test d'ombrage) :  $w_i = 0$  et **FIN**.
3. Tirage d'une normale (modèle de microfacettes).  
Réflexion spéculaire.
4. Si  $r_2$  n'appartient pas à la cible (blocage ou débordement) :  $w_i = 0$  et **FIN**.



# Algorithme Monte-Carlo



1. Génération de  $r_1$  et d'une direction  $\omega_0$  dans le cône solaire. Initialisation du poids:  $w_i = \rho DNI S_R |\omega_0 \cdot n_1|$
2. Si  $r_1$  n'appartient pas à un réflecteur (test d'ombrage) :  $w_i = 0$  et **FIN**.
3. Tirage d'une normale (modèle de microfacettes). Réflexion spéculaire.
4. Si  $r_2$  n'appartient pas à la cible (blocage ou débordement) :  $w_i = 0$  et **FIN**.



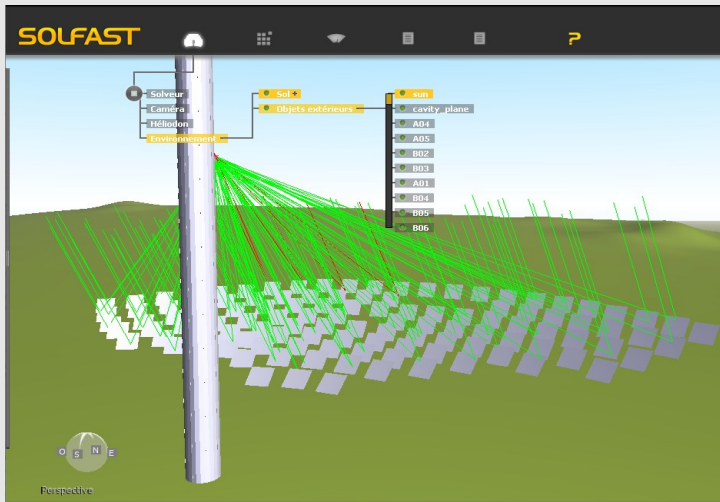
Estimation MC de la puissance reçue par la cible:

$$P \approx \langle w_i \rangle = \frac{1}{N} \sum_i w_i$$

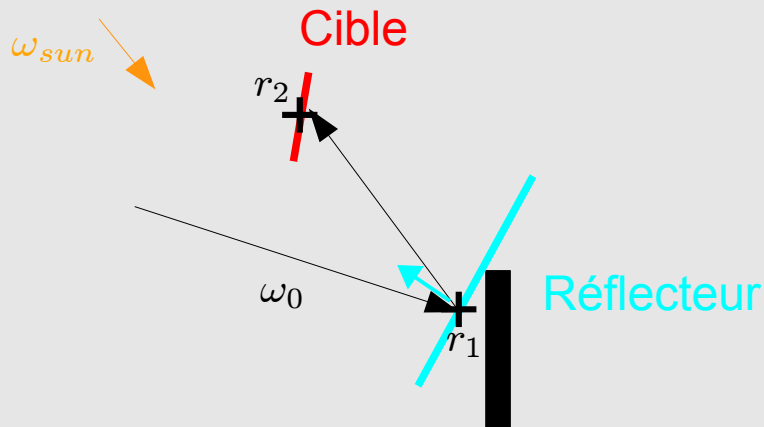
Estimation d'un intervalle de confiance:

$$\sigma_P \approx \frac{1}{\sqrt{N}} \sigma_w \quad \sigma_w = \langle w_i^2 \rangle - \langle w_i \rangle^2$$

# Algorithme Monte-Carlo



1. Génération de  $r_1$  et d'une direction  $\omega_0$  dans le cône solaire. Initialisation du poids:  $w_i = \rho DNI S_R |\omega_0 \cdot n_1|$
2. Si  $r_1$  n'appartient pas à un réflecteur (test d'ombrage) :  $w_i = 0$  et **FIN**.
3. Tirage d'une normale (modèle de microfacettes). Réflexion spéculaire.
4. Si  $r_2$  n'appartient pas à la cible (blocage ou débordement) :  $w_i = 0$  et **FIN**.



Estimation MC de la puissance reçue par la cible:

$$P \approx \langle w_i \rangle = \frac{1}{N} \sum_i w_i$$

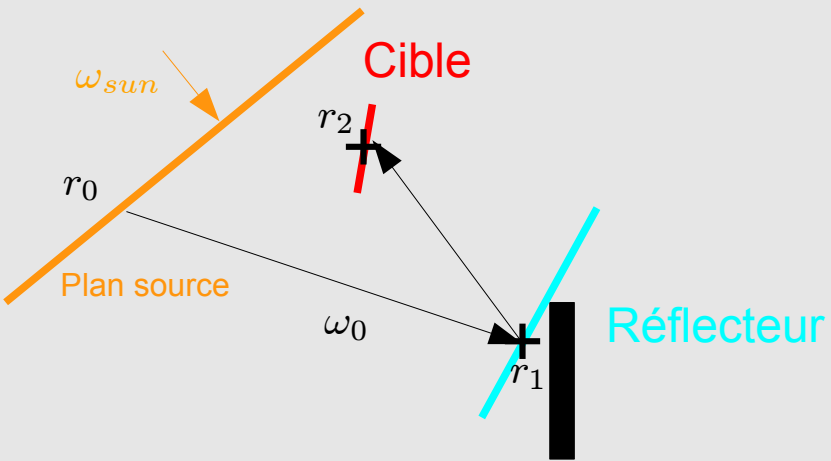
Estimation d'un intervalle de confiance:

$$\sigma_P \approx \frac{1}{\sqrt{N}} \sigma_w \quad \sigma_w = \langle w_i^2 \rangle - \langle w_i \rangle^2$$

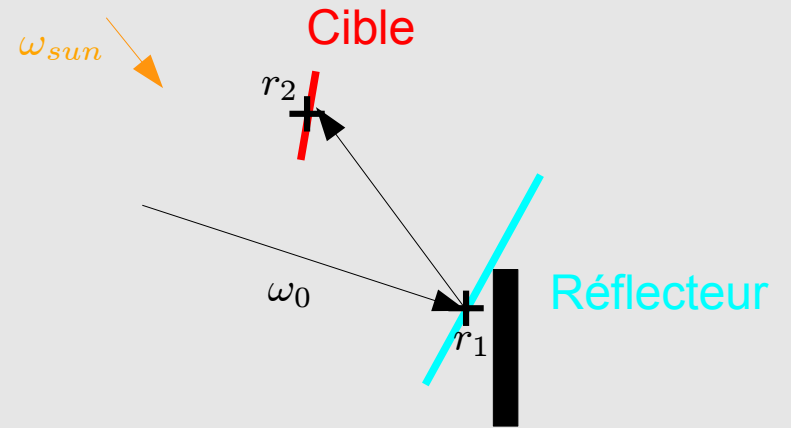
$$P = \int_{S_R} p_S ds(r_1) \int_{\Omega_{sun}} p_\Omega d\omega_0 H(r_0 \in sun) \int_{\Omega_{max}} p_{N_h} dn_h H(r_2 \in Cible) \rho DNI S_R |\omega_0 \cdot n_1|$$

# Analyse des algo MC

Algorithme Analogue (AA)

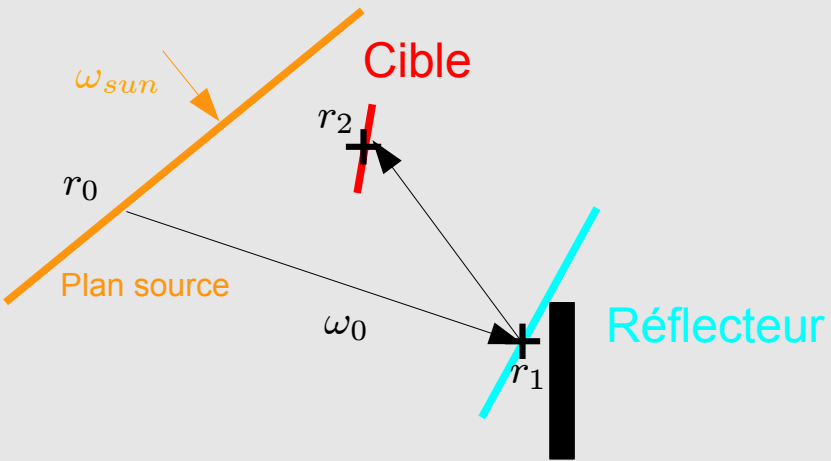


Algorithme Solfast (SF)

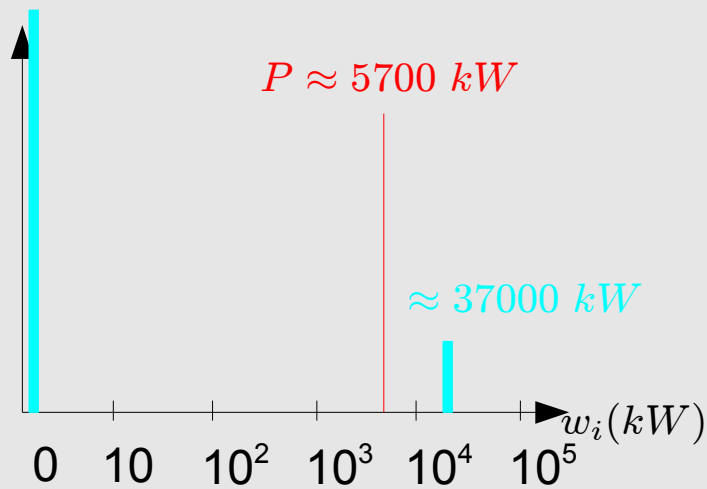


# Analyse des algo MC

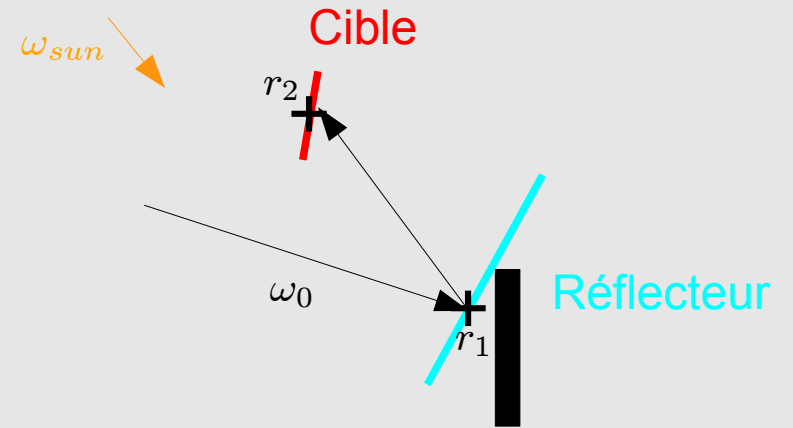
**Algorithme Analogue (AA)**



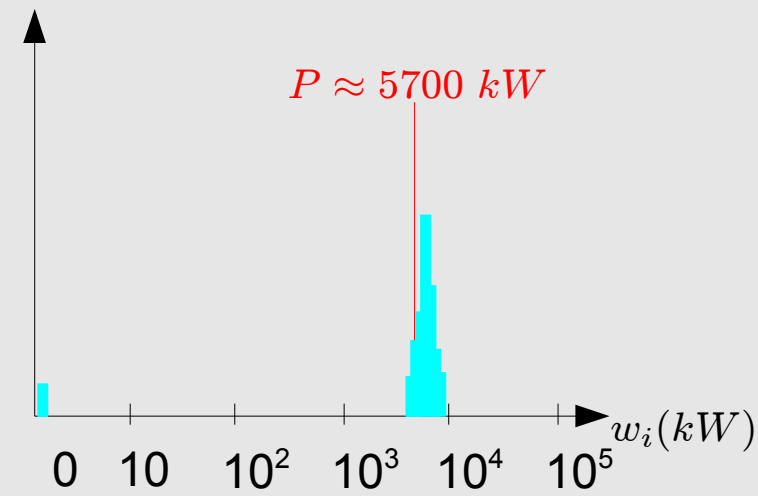
$\sigma_w \approx 13000 \text{ kW}$



**Algorithme Solfast (SF)**

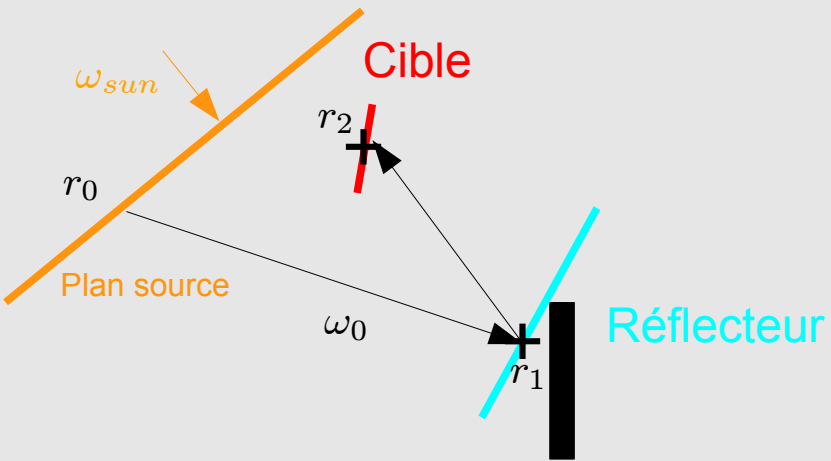


$\sigma_w \approx 800 \text{ kW}$

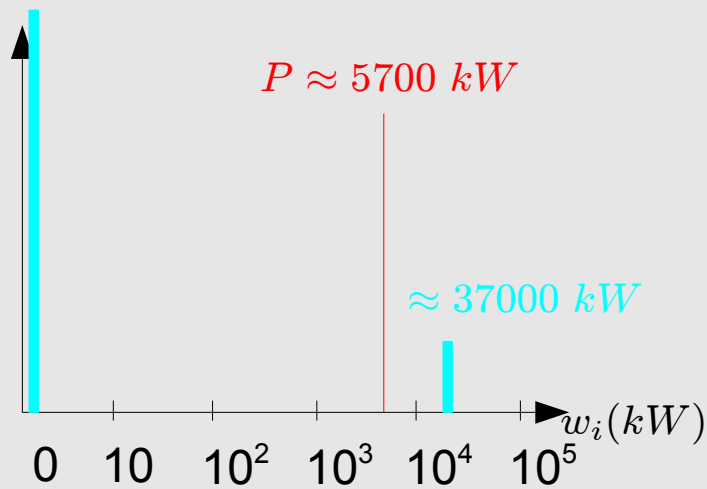


# Analyse des algo MC

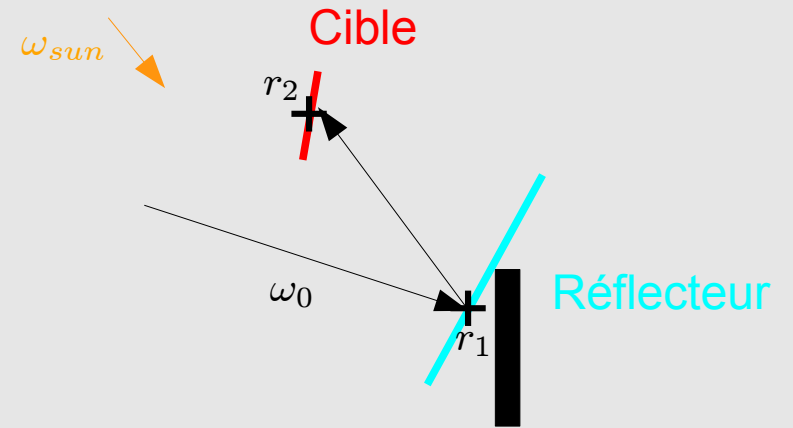
**Algorithme Analogue (AA)**



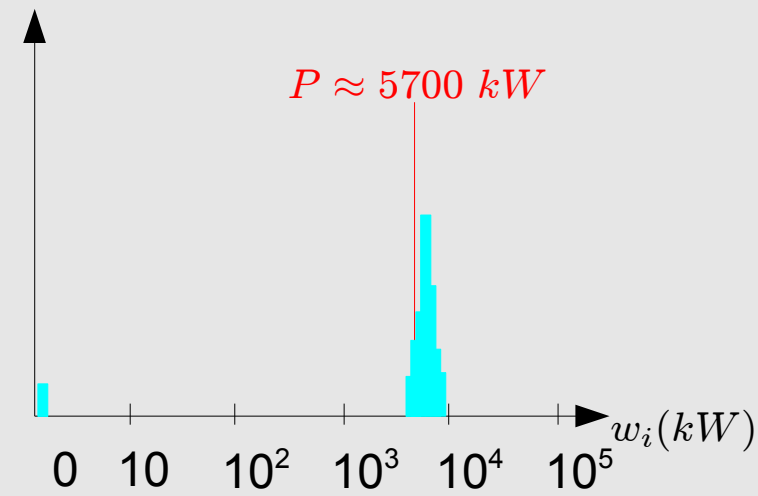
$$\sigma_w \approx 13000 \text{ kW}$$



**Algorithme Solfast (SF)**



$$\sigma_w \approx 800 \text{ kW}$$

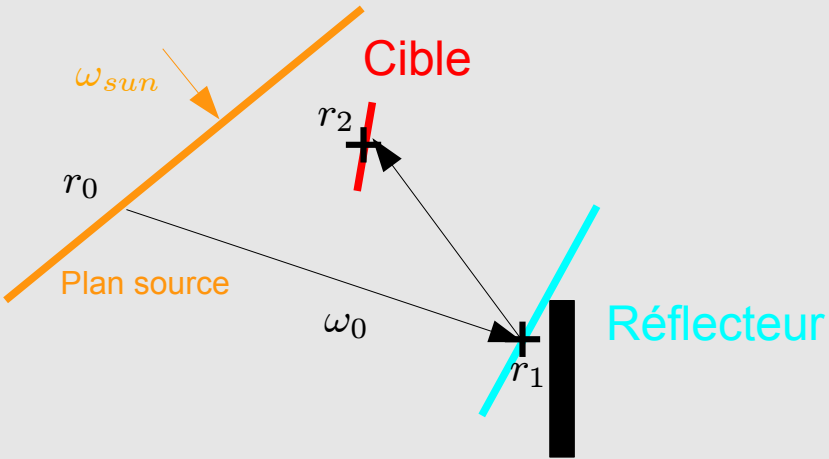


$$\sigma_P \approx \frac{1}{\sqrt{N}} \sigma_w$$

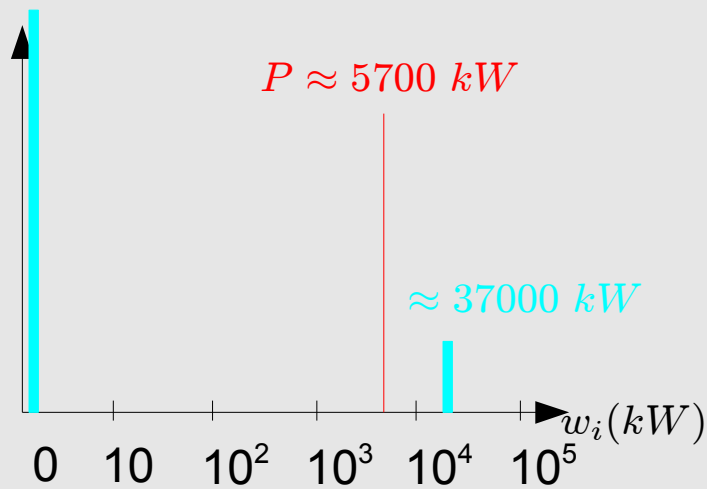


# Analyse des algo MC

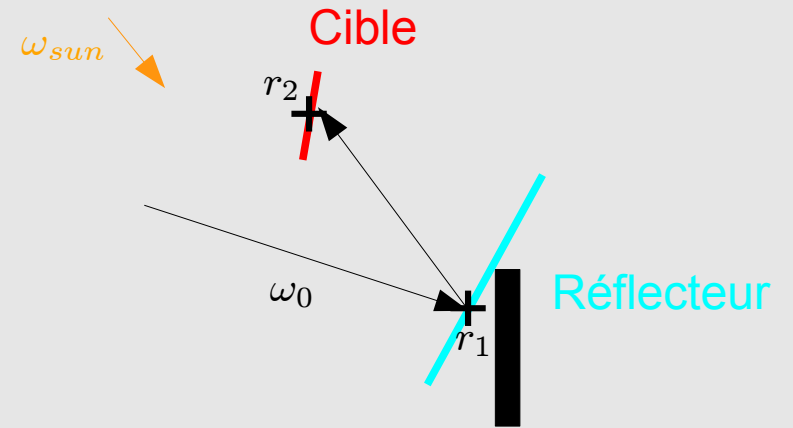
**Algorithme Analogue (AA)**



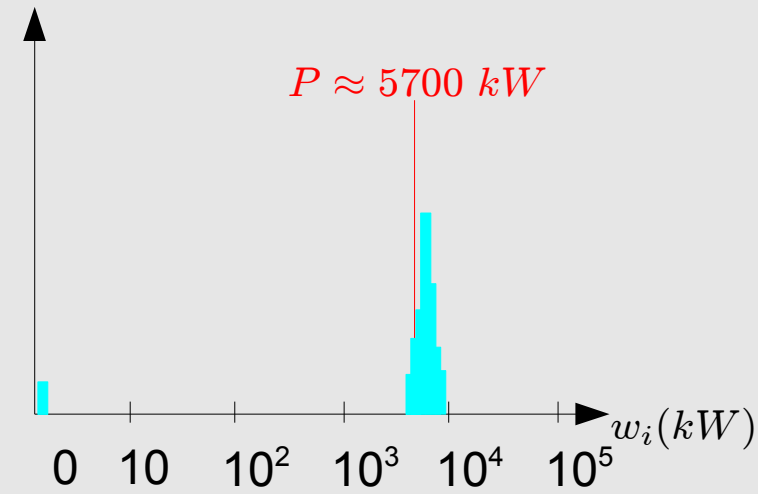
$\sigma_w \approx 13000 \text{ kW}$



**Algorithme Solfast (SF)**



$\sigma_w \approx 800 \text{ kW}$



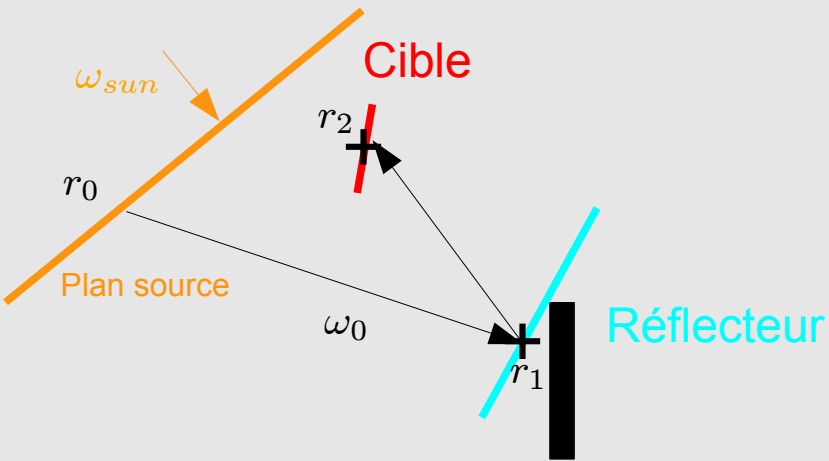
$$\sigma_P \approx \frac{1}{\sqrt{N}} \sigma_w$$

$$\sigma_P(AA) = \sigma_P(SF)$$

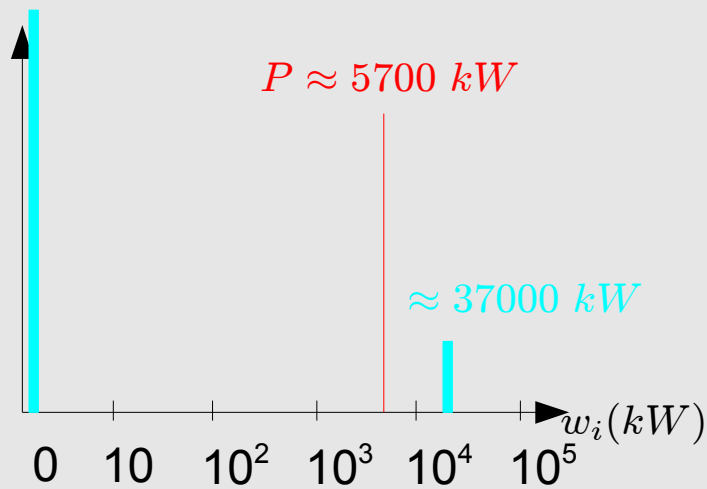
$$N(AA) = N(SF) \left[ \frac{\sigma_w(AA)}{\sigma_w(SF)} \right]^2$$

# Analyse des algo MC

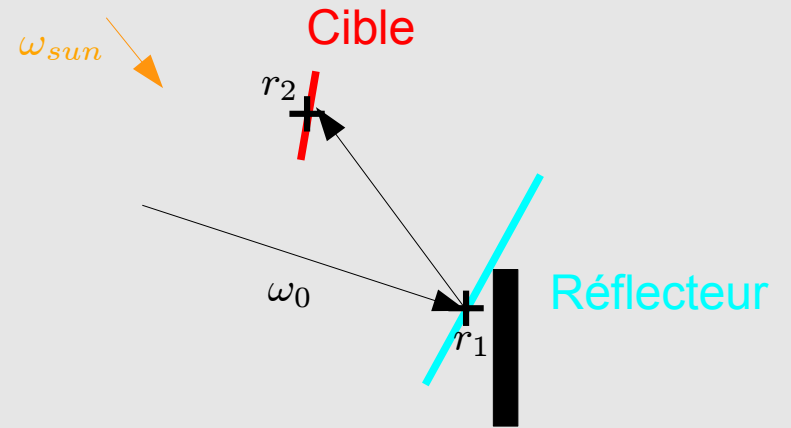
**Algorithme Analogue (AA)**



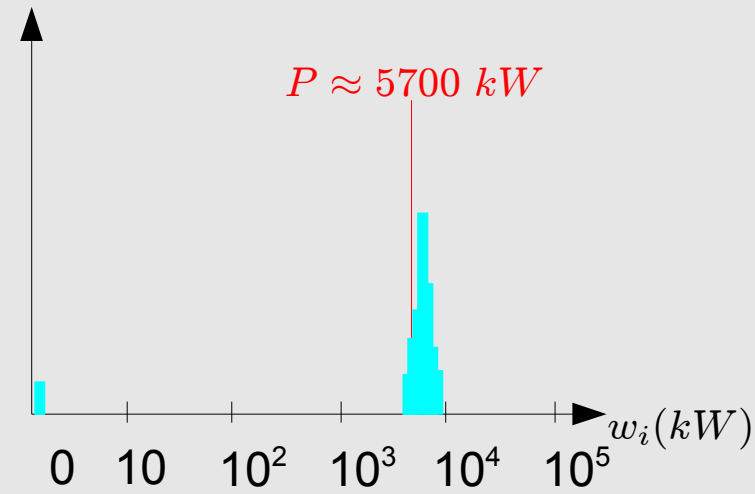
$\sigma_w \approx 13000 \text{ kW}$



**Algorithme Solfast (SF)**



$\sigma_w \approx 800 \text{ kW}$



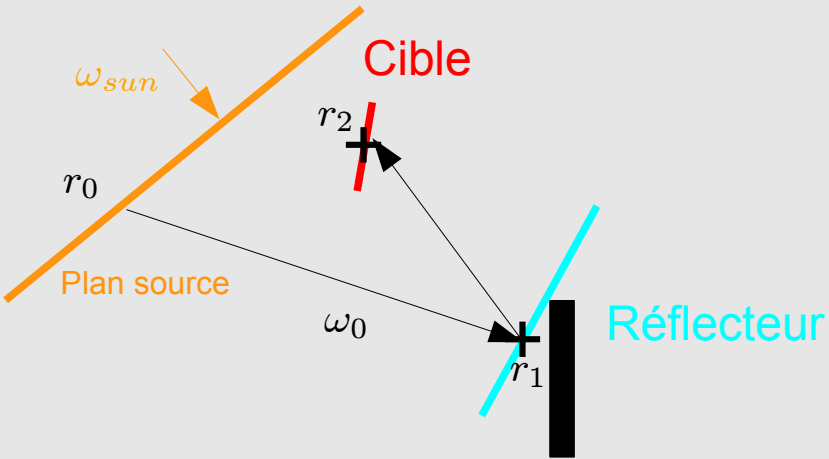
$$\sigma_P \approx \frac{1}{\sqrt{N}} \sigma_w$$

$$\sigma_P(AA) = \sigma_P(SF)$$

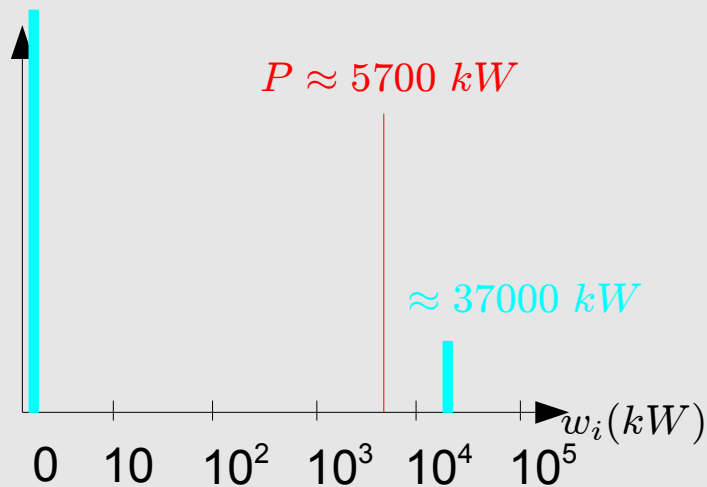
$$N(AA) = N(SF) \underbrace{\left[ \frac{\sigma_w(AA)}{\sigma_w(SF)} \right]^2}_{\approx 264}$$

# Analyse des algo MC

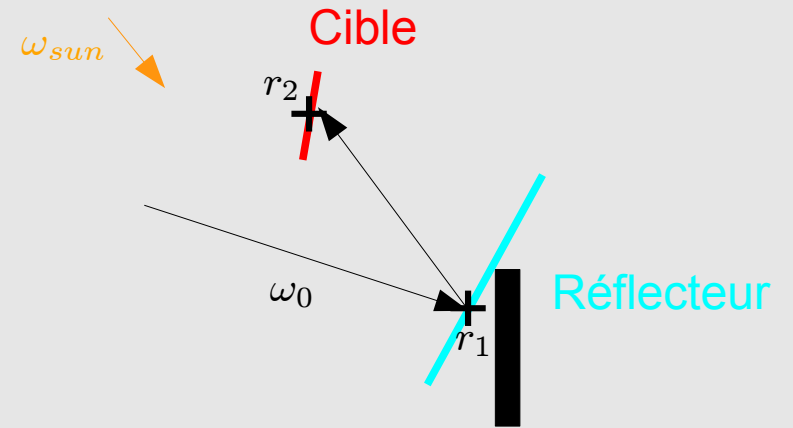
## Algorithme Analogue (AA)



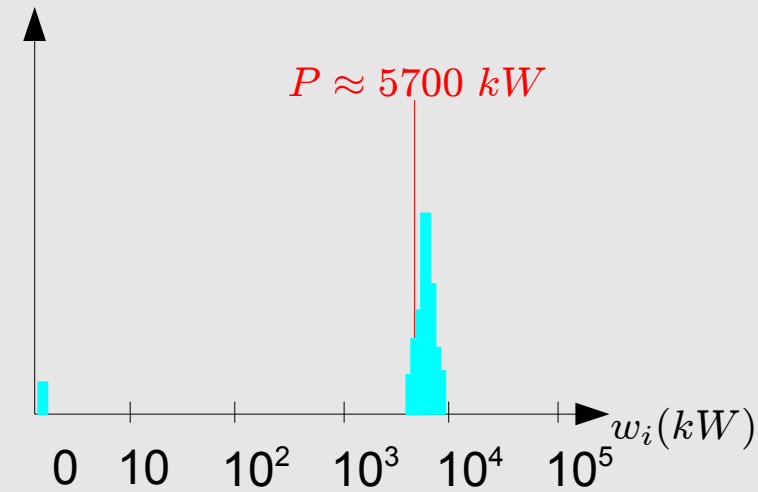
$$\sigma_w \approx 13000 \text{ kW}$$



## Algorithme Solfast (SF)



$$\sigma_w \approx 800 \text{ kW}$$



$$\sigma_P \approx \frac{1}{\sqrt{N}} \sigma_w$$

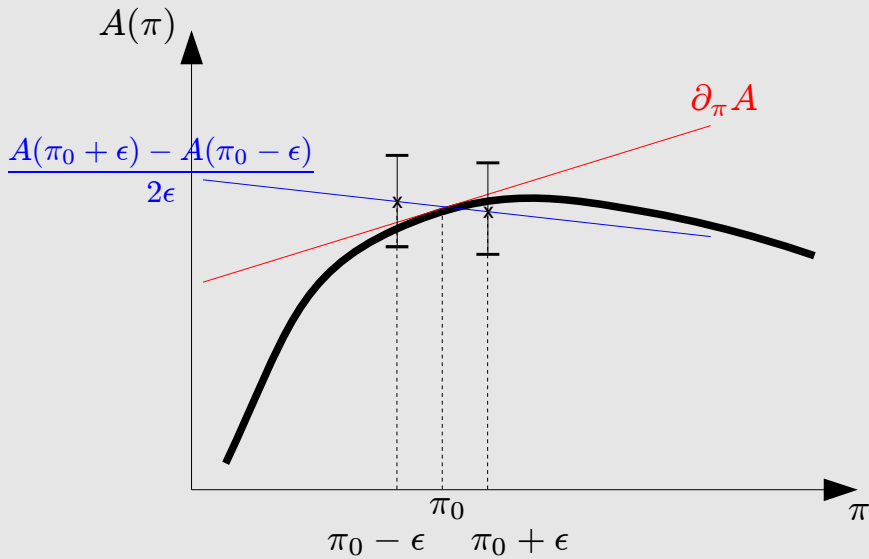
$$\sigma_P(AA) = \sigma_P(SF)$$

$$N(AA) = N(SF) \underbrace{\left[ \frac{\sigma_w(AA)}{\sigma_w(SF)} \right]^2}_{\approx 264}$$

Solfast	Tonatiuh	Soltrace
1.2	241	497

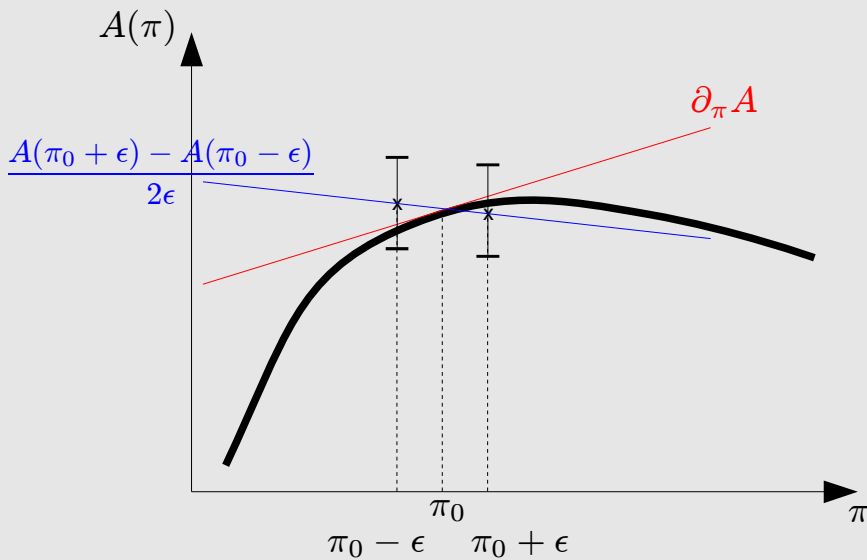
# Estimation de sensibilités

Sensibilité d'une grandeur  $A$  à un paramètre  $\pi$  :  $\partial_{\pi} A = \frac{\partial A}{\partial \pi}$



# Estimation de sensibilités

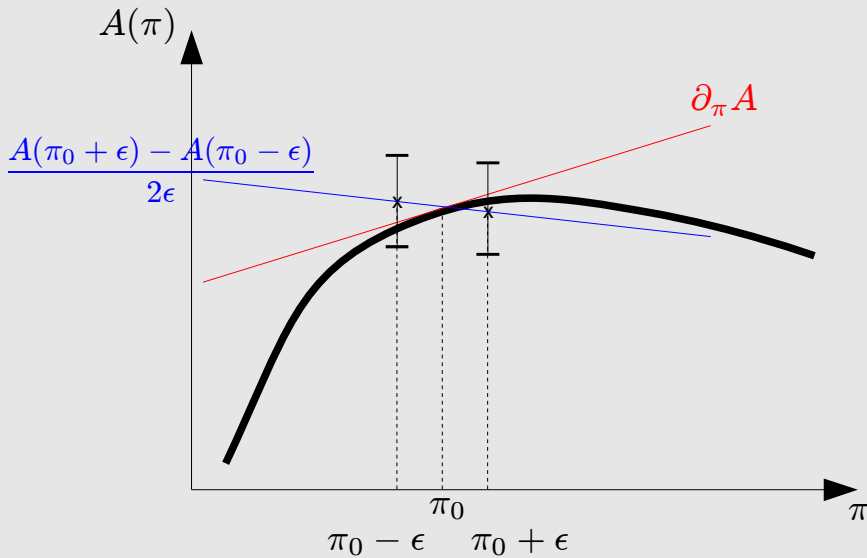
Sensibilité d'une grandeur  $A$  à un paramètre  $\pi$  :  $\partial_{\pi} A = \frac{\partial A}{\partial \pi}$



$$A(\pi) = \int dx P_X(x; \pi) w(x; \pi)$$

# Estimation de sensibilités

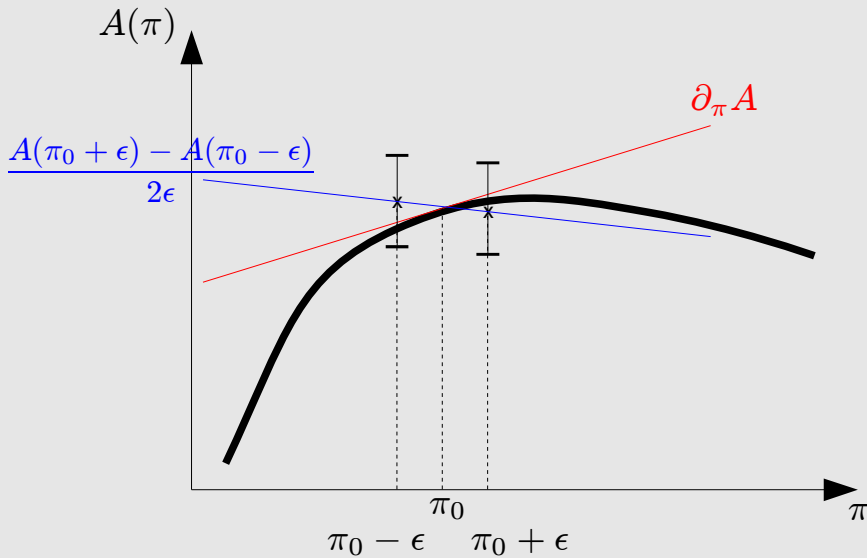
Sensibilité d'une grandeur  $A$  à un paramètre  $\pi$  :  $\partial_{\pi} A = \frac{\partial A}{\partial \pi}$



$$A(\pi) = \int dx P_X(x; \pi) w(x; \pi) \quad \Longrightarrow \quad \partial_{\pi} A = \int dx \partial_{\pi} [P_X(x; \pi) w(x; \pi)]$$

# Estimation de sensibilités

Sensibilité d'une grandeur  $A$  à un paramètre  $\pi$  :  $\partial_{\pi} A = \frac{\partial A}{\partial \pi}$

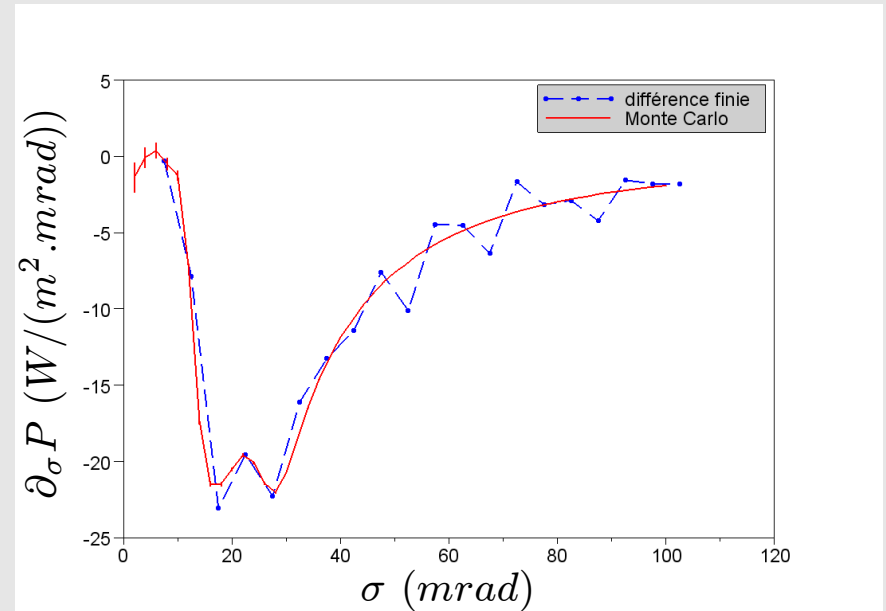
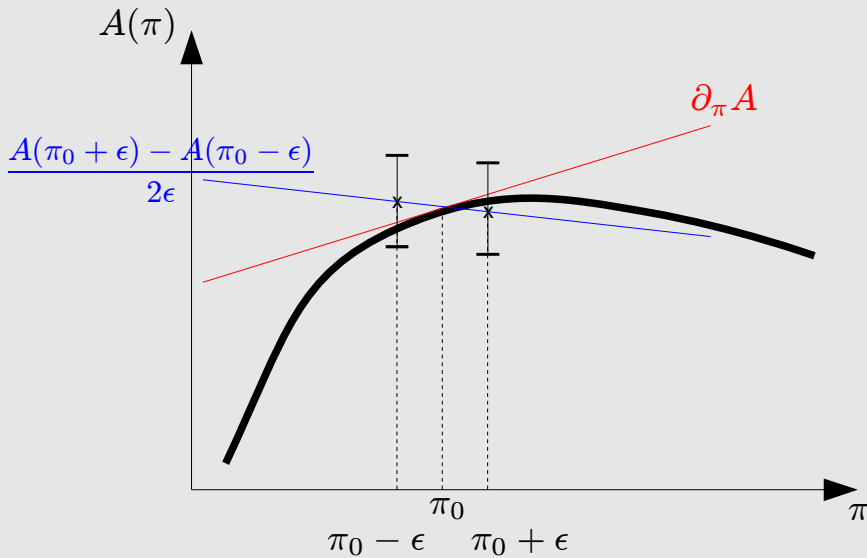


$$A(\pi) = \int dx P_X(x; \pi) w(x; \pi) \quad \Longrightarrow \quad \partial_{\pi} A = \int dx \partial_{\pi} [P_X(x; \pi) w(x; \pi)]$$

$A(\pi)$  et  $\partial_{\pi} A$  peuvent être évalués avec les mêmes générations aléatoires

# Estimation de sensibilités

Sensibilité d'une grandeur  $A$  à un paramètre  $\pi$  :  $\partial_{\pi} A = \frac{\partial A}{\partial \pi}$



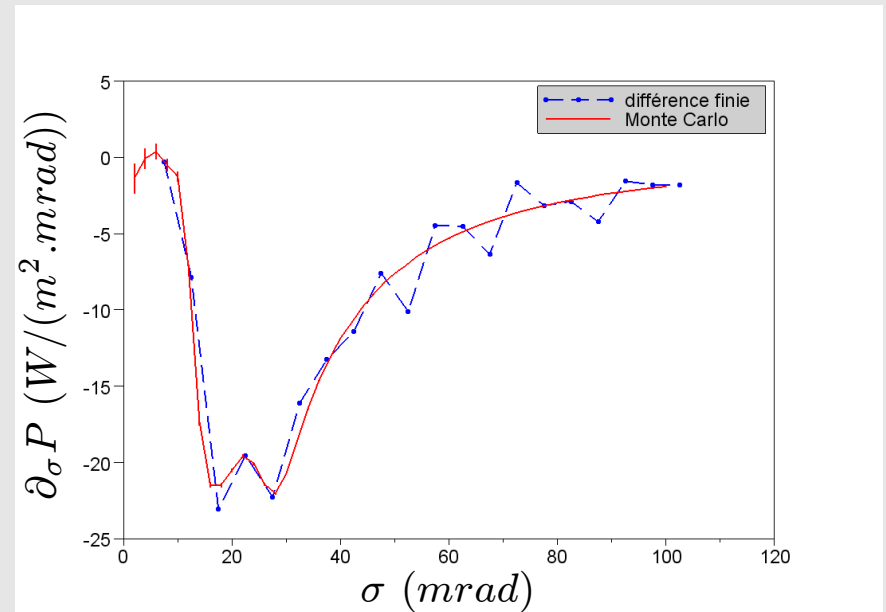
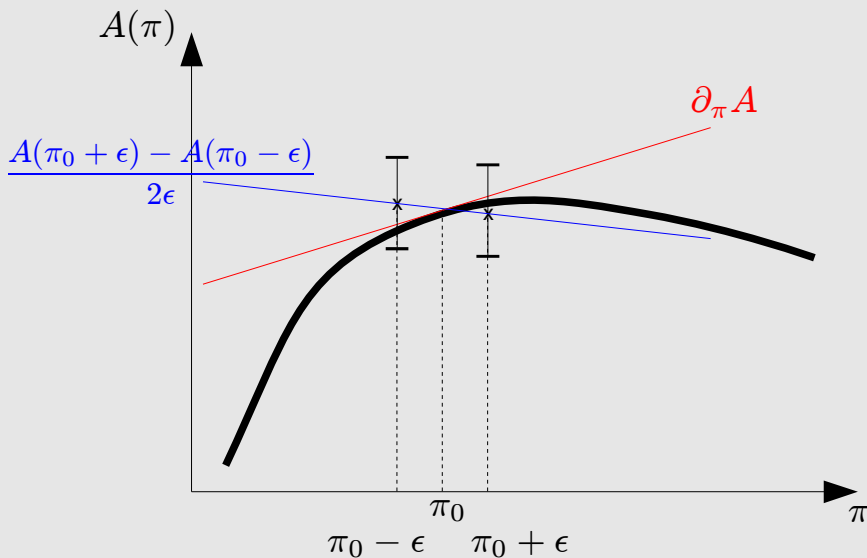
$$A(\pi) = \int dx P_X(x; \pi) w(x; \pi) \quad \Longrightarrow \quad \partial_{\pi} A = \int dx \partial_{\pi} [P_X(x; \pi) w(x; \pi)]$$

$A(\pi)$  et  $\partial_{\pi} A$  peuvent être évalués avec les mêmes générations aléatoires



# Estimation de sensibilités

Sensibilité d'une grandeur  $A$  à un paramètre  $\pi$  :  $\partial_{\pi} A = \frac{\partial A}{\partial \pi}$



$$A(\pi) = \int dx P_X(x; \pi) w(x; \pi) \quad \Longrightarrow \quad \partial_{\pi} A = \int dx \partial_{\pi} [P_X(x; \pi) w(x; \pi)]$$

$A(\pi)$  et  $\partial_{\pi} A$  peuvent être évalués avec les mêmes générations aléatoires

**!** si le domaine dépend de  $\pi$   $\Longrightarrow$  traitement particulier

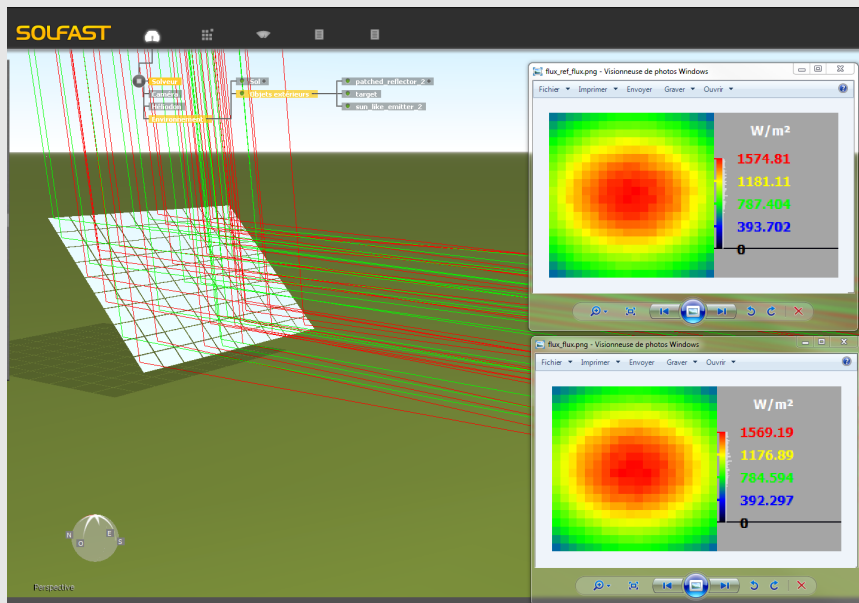
## Application à l'identification de paramètres

Héliostat de référence: parabole ( $f = 100 \text{ m}$ ) « fresnelisée »  
constitué de 100 miroirs plans carrés.

Dimensions:  $0.38 \text{ m} \times 0.38 \text{ m}$

Réflectivité:  $\rho \in [0.8, 0.9]$

Erreur optique:  $\sigma \in [4.5, 5.5] \text{ mrad}$



## Application à l'identification de paramètres

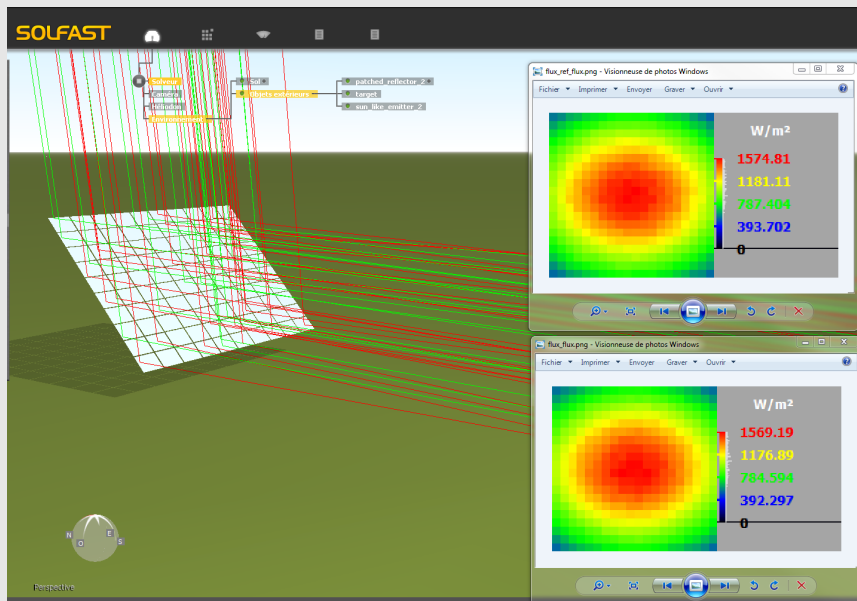
Héliostat de référence: parabole ( $f = 100 \text{ m}$ ) « fresnelisée »  
constitué de 100 miroirs plans carrés.

Dimensions:  $0.38 \text{ m} \times 0.38 \text{ m}$

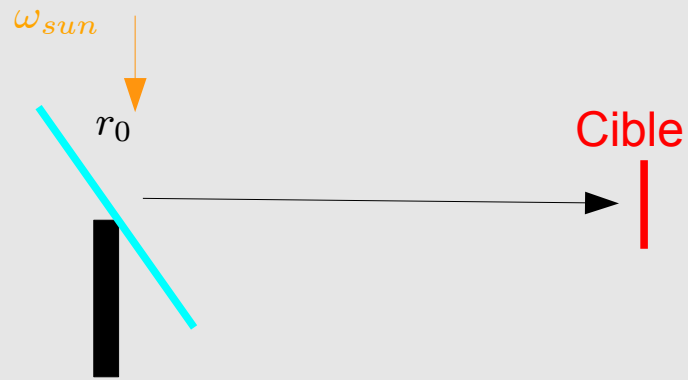
Réflectivité:  $\rho \in [0.8, 0.9]$

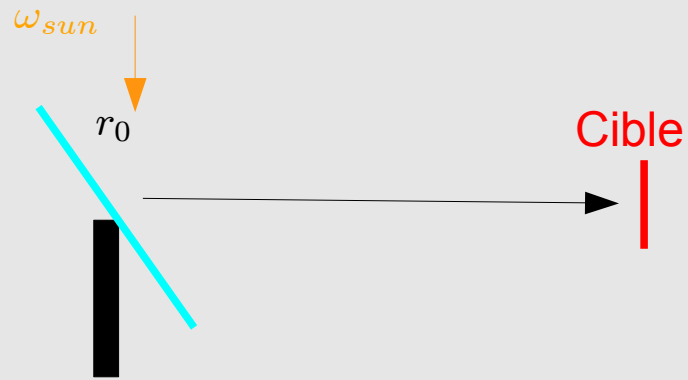
Erreur optique:  $\sigma \in [4.5, 5.5] \text{ mrad}$

Question?

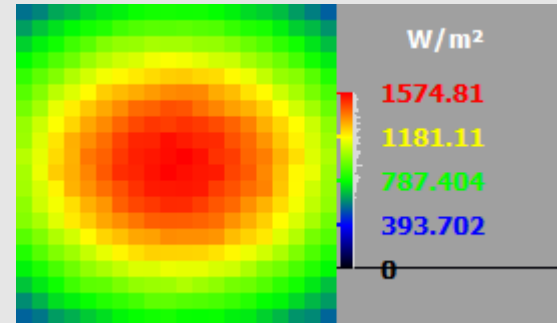


Quels sont les paramètres  $\rho_{eq}$  et  $\sigma_{eq}$  d'un miroir parabolique unique « équivalent » de dimensions  $4 \text{ m} \times 4 \text{ m}$  ?

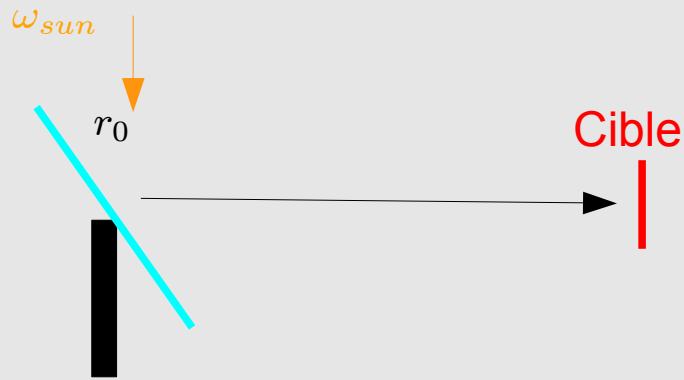




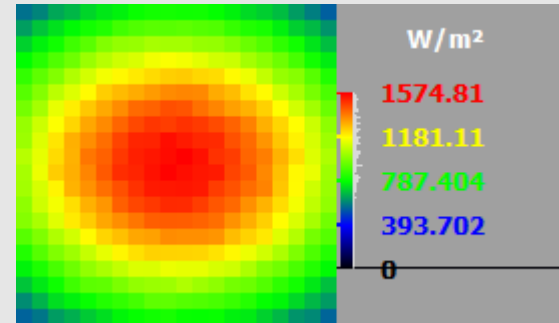
Carte de flux de référence (20 *pix* × 20 *pix*)



$\hat{\phi}_i$  Densité de flux du pixel  $i$  de l'héliostat de référence



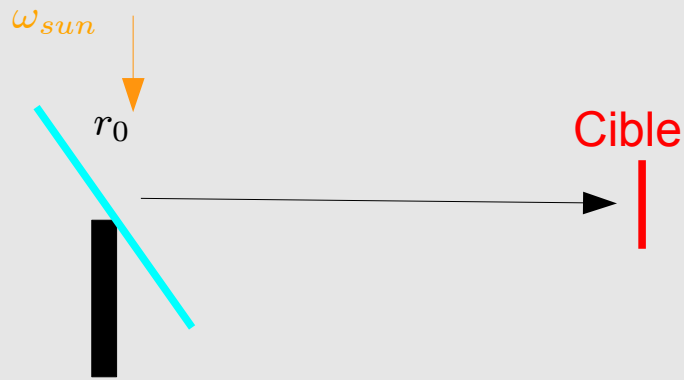
Carte de flux de référence (20 *pix* × 20 *pix*)



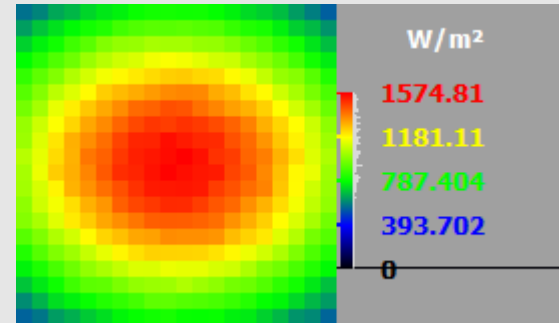
$\hat{\phi}_i$  Densité de flux du pixel  $i$  de l'héliostat de référence

Méthode des moindres carrés

$$\chi^2(\rho_{eq}, \sigma_{eq}) = \sum_i W_i \left[ \phi_i(\rho_{eq}, \sigma_{eq}) - \hat{\phi}_i \right]^2$$



Carte de flux de référence (20 pix × 20 pix)

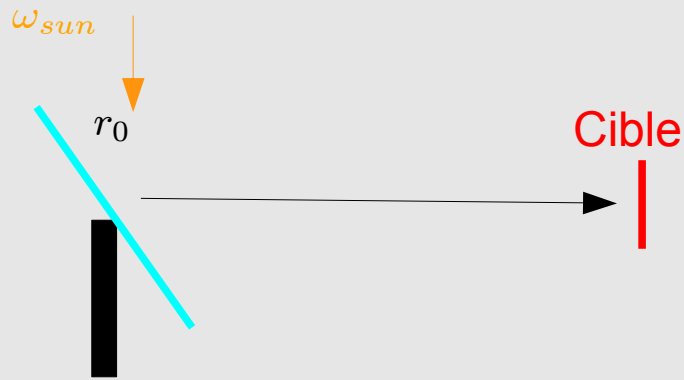


$\hat{\phi}_i$  Densité de flux du pixel  $i$  de l'héliostat de référence

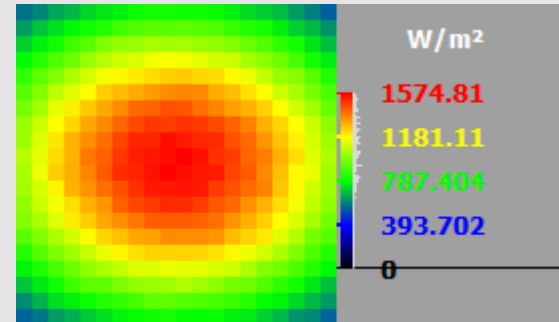
## Méthode des moindres carrés

$$\chi^2(\rho_{eq}, \sigma_{eq}) = \sum_i W_i \left[ \phi_i(\rho_{eq}, \sigma_{eq}) - \hat{\phi}_i \right]^2$$

$$\phi_i(\rho_{eq}, \sigma_{eq}) \approx \phi_i(\rho_0, \sigma_0) + (\rho_{eq} - \rho_0) \partial_\rho \phi_i + (\sigma_{eq} - \sigma_0) \partial_\sigma \phi_i$$



Carte de flux de référence (20 pix × 20 pix)



$\hat{\phi}_i$  Densité de flux du pixel  $i$  de l'héliostat de référence

## Méthode des moindres carrés

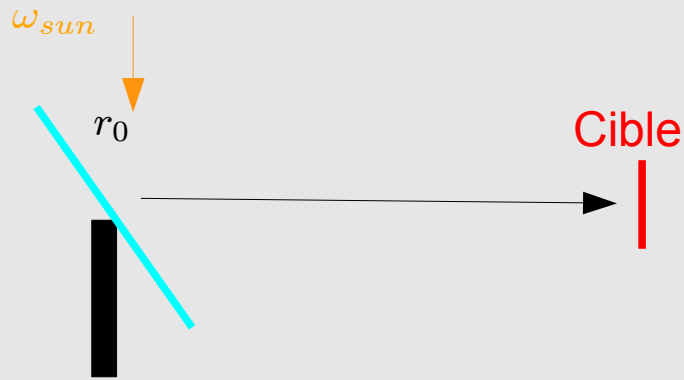
$$\chi^2(\rho_{eq}, \sigma_{eq}) = \sum_i W_i \left[ \phi_i(\rho_{eq}, \sigma_{eq}) - \hat{\phi}_i \right]^2$$

$$\phi_i(\rho_{eq}, \sigma_{eq}) \approx \phi_i(\rho_0, \sigma_0) + (\rho_{eq} - \rho_0) \partial_\rho \phi_i + (\sigma_{eq} - \sigma_0) \partial_\sigma \phi_i$$

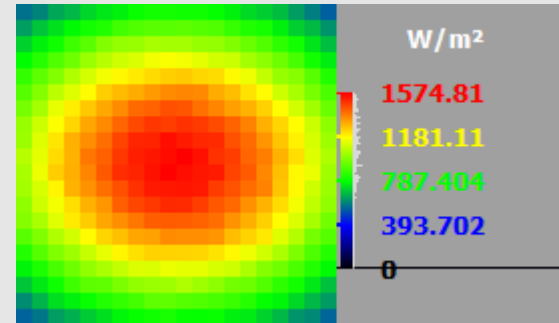
## Résolution itérative

$$\rho_0 = \rho_{eq}^{(t-1)} \quad \sigma_0 = \sigma_{eq}^{(t-1)}$$





Carte de flux de référence (20 pix × 20 pix)



$\hat{\phi}_i$  Densité de flux du pixel  $i$  de l'héliostat de référence

## Méthode des moindres carrés

$$\chi^2(\rho_{eq}, \sigma_{eq}) = \sum_i W_i \left[ \phi_i(\rho_{eq}, \sigma_{eq}) - \hat{\phi}_i \right]^2$$

$$\phi_i(\rho_{eq}, \sigma_{eq}) \approx \phi_i(\rho_0, \sigma_0) + (\rho_{eq} - \rho_0) \partial_\rho \phi_i + (\sigma_{eq} - \sigma_0) \partial_\sigma \phi_i$$

## Résolution itérative

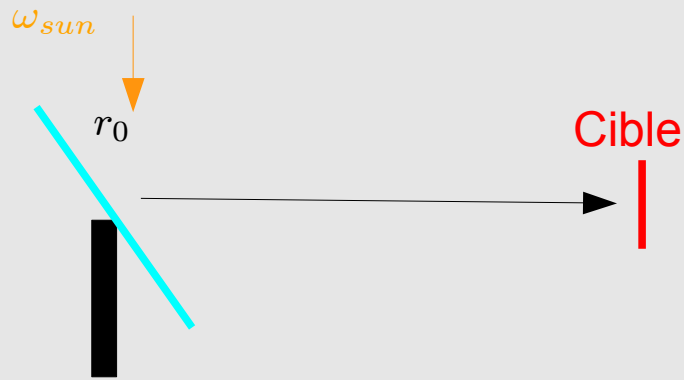
$$\rho_0 = \rho_{eq}^{(t-1)} \quad \sigma_0 = \sigma_{eq}^{(t-1)}$$

$$\nabla \chi^2(\rho_{eq}, \sigma_{eq}) = 0 \implies (\rho_{eq}, \sigma_{eq})$$

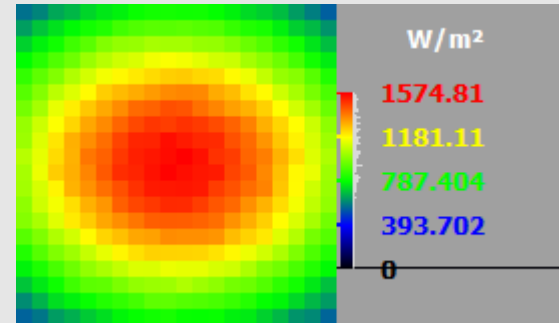
$$\phi_i(\rho_0, \sigma_0)$$

$$\partial_\rho \phi_i$$

$$\partial_\sigma \phi_i$$



Carte de flux de référence (20 pix × 20 pix)



$\hat{\phi}_i$  Densité de flux du pixel  $i$  de l'héliostat de référence

## Méthode des moindres carrés

$$\chi^2(\rho_{eq}, \sigma_{eq}) = \sum_i W_i \left[ \phi_i(\rho_{eq}, \sigma_{eq}) - \hat{\phi}_i \right]^2$$

$$\phi_i(\rho_{eq}, \sigma_{eq}) \approx \phi_i(\rho_0, \sigma_0) + (\rho_{eq} - \rho_0) \partial_\rho \phi_i + (\sigma_{eq} - \sigma_0) \partial_\sigma \phi_i$$

## Résolution itérative

$$\rho_0 = \rho_{eq}^{(t-1)} \quad \sigma_0 = \sigma_{eq}^{(t-1)}$$

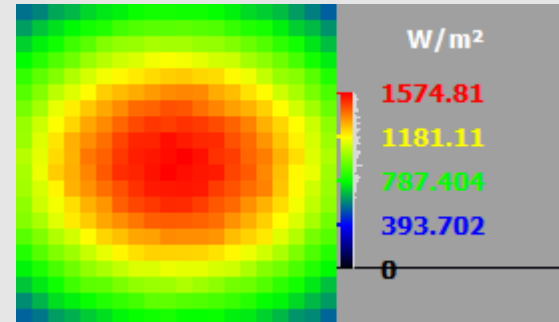
$$\nabla \chi^2(\rho_{eq}, \sigma_{eq}) = 0 \implies (\rho_{eq}, \sigma_{eq})$$

$$\begin{aligned} &\phi_i(\rho_0, \sigma_0) \\ &\partial_\rho \phi_i \\ &\partial_\sigma \phi_i \end{aligned}$$

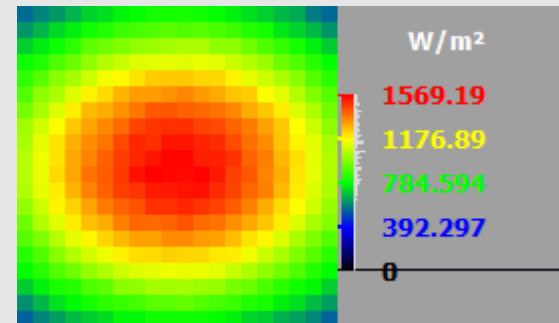




Carte de flux de référence (20 pix × 20 pix)



$\hat{\phi}_i$  Densité de flux du pixel  $i$  de l'héliostat de référence



$\rho_{eq} = 0.77$        $\sigma_{eq} = 5.03 \text{ mrad}$

Erreur relative < 2 % pour 86 % des pixels  
 Erreur relative max 4.7 %

## Méthode des moindres carrés

$$\chi^2(\rho_{eq}, \sigma_{eq}) = \sum_i W_i [\phi_i(\rho_{eq}, \sigma_{eq}) - \hat{\phi}_i]^2$$

$$\phi_i(\rho_{eq}, \sigma_{eq}) \approx \phi_i(\rho_0, \sigma_0) + (\rho_{eq} - \rho_0)\partial_\rho\phi_i + (\sigma_{eq} - \sigma_0)\partial_\sigma\phi_i$$

## Résolution itérative

$$\rho_0 = \rho_{eq}^{(t-1)} \quad \sigma_0 = \sigma_{eq}^{(t-1)}$$

$$\nabla\chi^2(\rho_{eq}, \sigma_{eq}) = 0 \implies (\rho_{eq}, \sigma_{eq})$$

$$\begin{aligned} &\phi_i(\rho_0, \sigma_0) \\ &\partial_\rho\phi_i \\ &\partial_\sigma\phi_i \end{aligned}$$

SOLFAST est un logiciel de simulation du flux solaire concentré

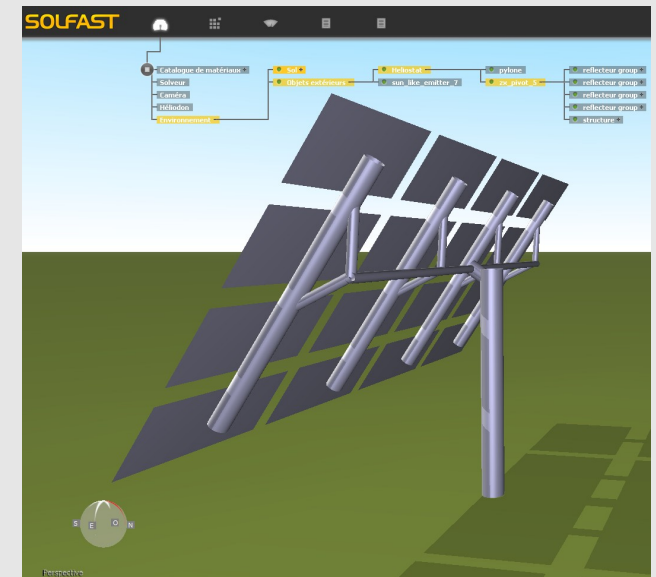
- Algorithme Monte-Carlo optimisé basé sur une formulation intégrale

- ➔ Performance de calcul très supérieurs aux logiciels existants

- ➔ Permet d'estimer des sensibilités paramétriques

- Modèles de matériaux: BRDF (modèle de microfacette)

- Algorithmes de tracking 1 axe/2 axes (« avec offset »)



**Pour aller plus loin:**

SOLFAST, a Ray-Tracing Monte-Carlo software for solar concentrating facilities, Eurotherm Seminar No. 95 (2012)

Monte-Carlo pour les applications solaires:

J. Delatorre et al., « Monte-Carlo advances and concentrated solar applications », soumis à Solar Energy (2012)