View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Loughborough University Institutional Repository

Loughborough University

This item was submitted to Loughborough's Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

creative commons

COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/

Accepted version: *Soler-Bientz, R. and F. Gómez-Castro. A computational model for estimating the performance of flat plane low concentration systems. in 19th European Photovoltaic Solar Energy Conference. 2004. Paris, France.*

A COMPUTATIONAL MODEL FOR ESTIMATING THE PERFORMANCE OF FLAT PLANE LOW CONCENTRATION SYSTEMS.

Soler Bientz R. and Gómez Castro F.
Energy Laboratory. Faculty of Engineering. Autonomous University of Yucatan (UADY).
Avenida Industrias no contaminantes por Anillo periférico norte s/n. Mérida. Yucatán. México.
Tel: +52 999 9410191, ext: 147. Fax: +52 999 9410189.
Email: sbientz@tunku.uady.mx

ABSTRACT: The implementation of PV systems is one of the feasible alternatives to meet the energy supply in communities far from the conventional electric grid. Several studies have shown that a flat plane concentrator can produce better electrical behaviour in a PV system, due to the reduction of the effective panel surface needed for the operation of the system. As the cost of a PV system is strongly conditioned by the cost of the PV panels, it can lead to an increase of the PV market. This work describes the computational model developed to represent the concentration pattern over a collector, generated by a flat plane reflector. A useful 3D representation of the system has been programmed to visualize the concentration system. The computational model was developed in Mathematica Programming Language Environment. The simulation tool created shows that Mathematica is a quick and powerful tool to represent the low-concentrating photovoltaic systems.
Keywords: Software, Modelling, Concentrators.

1. INTRODUCTION

The implementation of PV systems is one of the feasible alternatives to meet the energy supply in communities far from the conventional electric grid. Several studies have shown that a flat plane concentrator can produce better electrical behaviour in a PV system, due to the reduction of the effective panel surface needed for the operation of the system [1]. As the cost of a PV system is strongly conditioned by the cost of the PV panels, this can lead to an increase of the PV market.

The main objective of the concentrator is to increase the amount of solar radiation that arrives to the collector surface. If this surface is a PV module, the result will be an increase of generated power density.

One of the simplest concentrators is the flat plane concentrator, which is formed by two flat surfaces called the reflector and the collector. Several authors have studied the structure, the materials, and the performance of these kinds of concentrators [2, 3].

This paper describes the computational model developed to represent the concentration pattern generated by a flat plane reflector over a collector. A useful 3D representation of the system was programmed to visualize the concentration effects.

2. MATHEMATICAL MODEL

2.1 Coordinate systems

The path of the ray of light through the reflector-collector system can be fully described by employing three systems of coordinates: The horizontal system (X,Y,Z); the reflector system (X´,Y´,Z´); and the collector system (X”,Y”,Z”).

The horizontal coordinate system is defined as the system that contains the horizontal plane, taking directions South and West as positive directions to X and Y axis respectively. The reflector system contains the reflector plane, which is generated by a double rotation: First, “γ_C” angle over a Z axis, and secondly, “Ir” over the resulting Y axis. On the other hand, the collector coordinate system is formed when the reflector system is rotated an amount of (180-(Ir+Ic)), in the same direction of the Y axis. All the angles are taken as positive in the clockwise direction. Such parameters can be seen in Fig. 1.

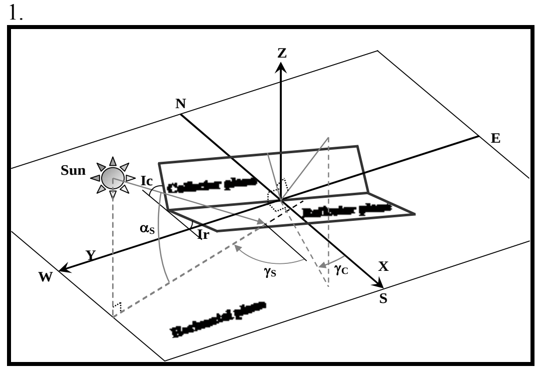

Figure 1: Graphical representation of the main parameters that defines the system under study.

2.2 Relations between the horizontal and the collector plane.

The main objective is to find the image that is produced by one ray of light—expressed in horizontal coordinates (X,Y,Z)—over the collector (X”,Y”,Z”), after it has been reflected on the (X´,Y´,Z´) system. This procedure is executed by the following equations system:

$$\begin{aligned} X' &= X\,Cos[I_r]Cos[\gamma_C] + Y\,Cos[I_r]Sin[\gamma_C] + Z\,Sin[I_r] \\ Y' &= -X\,Sin[\gamma_C] + Y\,Cos[\gamma_C] \\ Z' &= -X\,Sin[I_r]Cos[\gamma_C] - Y\,Sin[I_r]Sin[\gamma_C] + Z\,Cos[I_r] \end{aligned} \quad (1)$$

$$\begin{aligned} X'' &= X'\,Sin[I_c + I_r] + Z'\,Cos[I_c + I_r] \\ Y'' &= Y' \\ Z'' &= -X'\,Cos[I_c + I_r] + Z'\,Sin[I_c + I_r] \end{aligned} \quad (2)$$

An arbitrary ray of light could be expressed in vector form on the horizontal plane as:

$$\vec{S} = Cos[\alpha_S]Cos[\gamma_S]\hat{i} + Cos[\alpha_S]Sin[\gamma_S]\hat{j} + Sin[\alpha_S]\hat{k} \quad (3)$$

where $(\hat{i}, \hat{j}, \hat{k})$ are unit vectors in the horizontal coordinates system, and the other parameters are represented in Fig. 1.

By means of the above equation it is possible to find the angle of the incident ray over the reflector " θ_{IR} " and the angle that forms the reflected ray relative to the collector normal " θ_{IC} ":

$$\begin{aligned} Cos[\theta_{IR}] &= -Cos[\alpha_S]Cos[\gamma]Sin[I_r] + Cos[I_r]Sin[\alpha_S] \\ Cos[\theta_{IC}] &= Cos[\alpha_S]Cos[\gamma]Sin[I_c + 2I_r] \\ &- Cos[I_c + 2I_r]Sin[\alpha_S] \end{aligned} \quad (4)$$

where $\gamma = \gamma_S - \gamma_C$ an the rest of the angles are represented in Fig. 1.

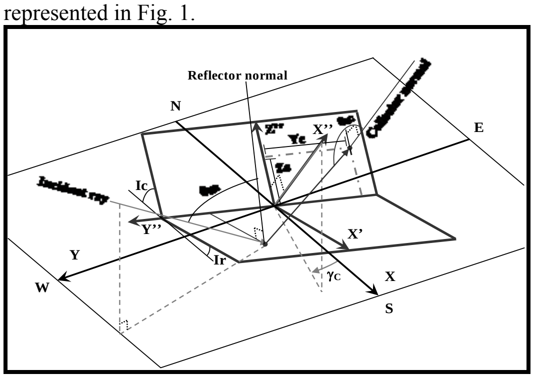

Figure 2: The path of the incident ray of light before and after the reflection.

Finally, the coordinates of the reflected ray of light over the collector plane are expressed by means of the following equations:

$$\begin{aligned} Yc &= \frac{X_R Sin[I_c + I_r]Sin[\gamma]}{Cos[\gamma]Sin[I_c + 2I_r] - Tan[\alpha_S]Cos[I_c + 2I_r]} + Y_R \\ Zc &= \frac{X_R(-Sin[I_r]Cos[\gamma] + Tan[\alpha_S]Cos[I_r])}{Cos[\gamma]Sin[I_c + 2I_r] - Tan[\alpha_S]Cos[I_c + 2I_r]} \end{aligned} \quad (5)$$

The parameters involved in the above equation are represented in Fig. 2. The model outlined considers that the horizontal coordinates of the ray of light, " α_S " and " γ_S ", indicate the direction from where a set of rays will impact the reflector plane.

3. COMPUTATIONAL MODEL

3.1 Program general structure

Due to the availability of resources which allows an easy description of a given physical problems and the representation of its graphical solution, the computational model was developed in Mathematica programming language. The software generated has been programmed in a flexible style, which leads to an open code application that can be easily modified and generalized to describe other concentrating situations.

This software is organized in three main parts as shown in Fig. 3, in which every part has been classified using a set of Mathematica-structured cells.

The first part is concerned with the theoretical explanations of all the mathematical procedure used. The cells of this part are not executables. Their function is to allow an easy understanding of the procedure followed in the computational model

The second part is dedicated to the executable programming codes and it is organized in the following subparts: Routines for general support tasks; routines for mathematical model of the concentration; routines for the making of graphics and the simulation of the concentrator systems; and the main routines which facilitate the interaction with the user.

Finally the third part gives an easy access to the main routines of the program, by means of the several prototype examples. Such examples present a brief summary of the syntax of the main functions, and specific cases ready to be executed.

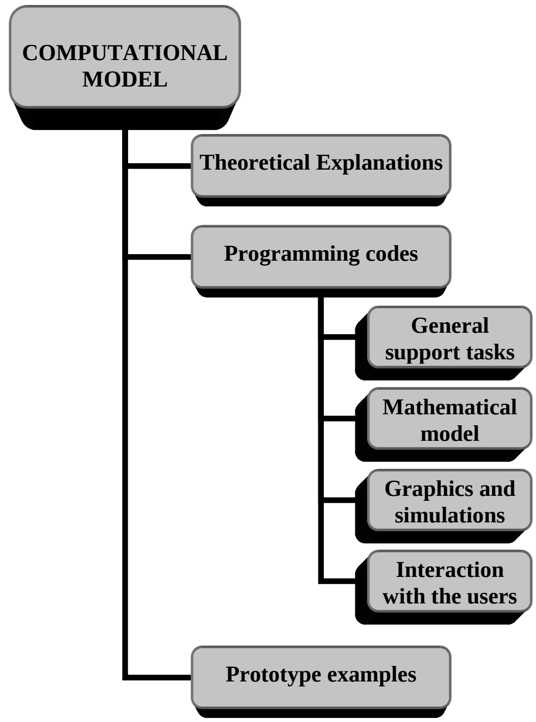

Figura 3: General diagram of the Computational model.

The Mathematica allows the combination of text cells with code cells. Therefore, the definition of every routine programmed has been designed considering the following five specification formats, each one in its corresponding cell:

- *Function:* Shows the syntax that must be used to execute the routine.
- *Objective:* Give a brief description of the purpose of the routine.
- *Input parameters: D*escribes the types of variables, the range of the possible values and the meaning of the variables within the routine.
- *Output parameters:* Explains the results obtained from the execution of the routines.
- *Routine programming code:* Defines the programming codes of the routines. This is the cell that is executed as part of the computational model.

3.2 Basic input parameters

The values of all input parameters are assigned by means of calls to the main functions. These parameters are organized in two types:

- Those related with the identification of the position of the Sun: Geographical coordinates, and date and time. Date and time are optional parameters. If they are not provided, the program uses the current computer date and time as a default value.
- Those related with reflector-collector system: Dimensions of the reflector and the collector; the

inclination—relative to the horizontal surface—of the reflector and the collector; and the azimuth angle of the entire system.

The above mention types of input parameters are grouped in two lists, one for Sun parameters, and one for system parameters. The optional parameters are presented in green, while the other parameters are presented in red. Before the parameters can be used within a specific routine, several control actions apply to verify their values.

3.3 Code sample

As a sample of the program code, Fig. 4 shows, the routine "*ArbitraryLightSource3D[{Wc_Real,ϵ_Real},{Lc_Real,σ_Real},{Ic_Real,Ir_Real,γc_Real},{γs_Real,αs_Real}]*". This routine is designed for the representation, in a three dimensional system, of the reflector-collector system, and of any arbitrary source light. As shown in Fig. 4, this routine makes use of the several auxiliary routines previously programmed. The input parameters are represented in four lists of variables. The first two describe the dimensions of the reflector-collector system. The third one is dedicated to the tilts angles and the azimuth of the entire system. Finally, the fourth list establishes the position of the source of light.

```
ArbitraryLightSource3D[{Wc_Real,ϵ_Real}, {Lc_Real, σ_Real},
      {Ic_Real,Ir_Real, γc_Real}, {γs_Real,αs_Real}] /;
   ((Wc > 0.) && (ϵ > 0.)  && (Lc > 0.)  && (σ > 0.)  &&
      (0. ≤ Ic ≤ 90.) &&  (0.≤ Ir ≤ 90.)  &&
      (-180. ≤ γc ≤ 180.) && (-180. ≤ γs ≤ 180.) &&
      (0. ≤ Abs[αs] ≤ 90.)) :=
 Module[{R, SystemList, SourcePossition, GraphicsObject},
   R = Max[Lr, Wr];
   (* Graphics primitive of Reflector-Collector*)
   SystemList = Design3DSystem[{Wc, ϵ}, {Lc, σ}, {Ic, Ir, γc}];
   (* Graphics primitive of Source of Light*)
   SourcePossition = SourceLightMotion[R, {γs, αs}];
   GraphicsObject =
     Graphics3D[{Soil[R], SystemList , SourcePossition},
       Boxed → True, ColorOutput → GrayLevel,
       ViewCenter → {1/2, 1/2, 1/2},
       PlotRange → {{-R, R}, {-R, R}, {0, R}}, ViewPoint →
        {1/2 (1/2 Lr Cos[γc°] - Wr Cos[Ir°] Sin[γc°]),
         1/2 (Wr Cos[Ir°] Cos[γc°] + 1/2 Lr Sin[γc°]),
         Wr Sin[Ir°]}];
   Return[GraphicsObject];
 ];
```

Figure 4: Example of the routine code which represents a 3D system and an arbitrary source of light.

3.4 Procedure to execute the simulation tool

This program was designed to allow the execution of the initialization process before the execution of the first routine. The execution of the initialization process thus prepares the proper environment with all the definitions required.

The computational model developed is ready to be used or modified in any level of programming. The general simulation tasks can be accessed specifically through the main routines.

The main routines available can be easily executed by means of prototype examples. These main routines include the following three tasks:

- 3D system reflector-collector representation.
- Concentration produced by any source of light located in arbitrary coordinates.
- Concentration produced by the Sun at specific date and time.

These tasks can be executed in any order. In the case of concentration produced by the Sun, six different geographical positions are included, to cover a wide range of examples: Toronto, Tropic of Cancer, Merida/Yucatan, Equator, Rio de Janeiro, Tropic of Capricorn and Sidney. The Sun movement model used was the described in reference [4, 5].

4. RESULTS

4.1 Arbitrary source of light

When working with an arbitrary source of light, this software allows the user to freely locate the position of the source of light. In this case, any given coordinates (azimuth, altitude) is used to calculate the concentration pattern produced by a set of parallel rays on collector, after it is reflected on the reflector plane.

Fig. 5 shows a 3D representation of the reflector-collector system with a source of light located at azimuth of -10° and altitude of 50°. The corresponding images on the collector can be observed in Fig. 6.

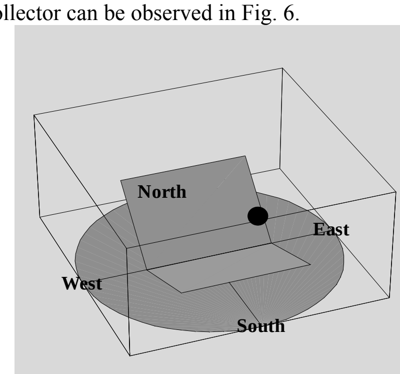

Figure 5: 3D view of reflector-collector system and arbitrary source of light.

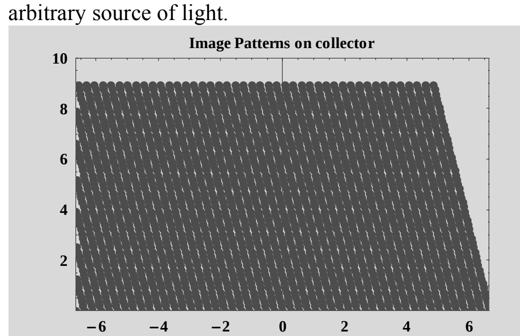

Figure 6: **I**mage produced on collector by arbitrary source of light, showed above.

The program code used to calculate the above results is:

ArbitraryLightPlanePatterns[{{13.3,1},{10,1}},{45,20,0}, {-10,50}, 50]

where the first list represents the reflector-collector dimensions; the second list establishes the tilt angles; the third list indicates the source of light position; and the final list specifies the amount of rays of light represented on the collector.

4.2 Specific solar Sun position patterns.

In cases where the Sun constitutes the source of light, the program needs primarily the geographical latitude and longitude and, secondarily, the date and the time, in order to identify the sun ray direction. Fig. 7 and Fig. 8 show the 3D system representation and the images on the collector.

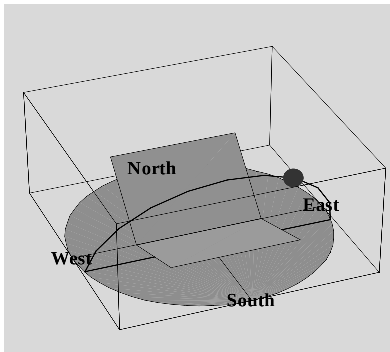

Figure 7: 3D view of the reflector-collector system, the Sun position, and the path of the Sun on a specific day, according to the geographical data provided.

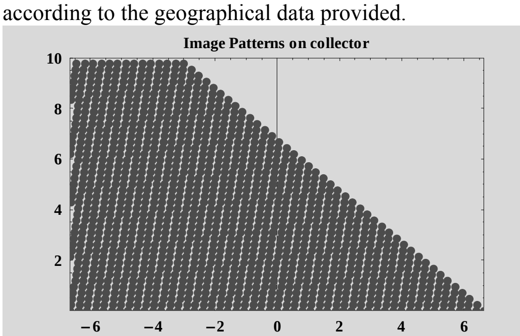

Figure 8: Image produced on collector by the Sun, in the case shown in Fig. 7.

Fig. 7 also shows the path of the Sun during a specific day and at a specific location. The corresponding prototype of the main routine that produces the results of the Fig. 7 and Fig. 8 is:

HourlySunLightLightPlanePatterns[{{13.3,1}, {10,1}},{45,20,0},{20.96,89.63}, {10,0,0}, 51, 2004]

where each argument represents:

- *{{13.3,1},{10,1}}*: Dimensions of the reflector-collector system.
- *{45,20,0}:* Tilt angles: 45° for the collector, 20° for the reflector and 0° for the azimuth of the entire system.
- *{20.96,89.63}:* Geographical latitude and longitude for Mérida city.
- *{10,0,0}:* Time: hours, minutes and seconds.
- *51:* Day number: February 20.
- *2004:* Year.

5. CONCLUSION

A flexible program has been developed to calculate the image produced by a plane reflector over the collector surface. Useful 3D representations of the systems can be generated to visualize the concentration effects.

The simulation tool created shows that Mathematica is a quick and powerful tool to represent the low-concentrating photovoltaic systems.

Using the routine programmed, future work is planned to calculate and represent the cumulative patterns of the concentration system over a period of time. In addition, other geometrical configurations of the concentrator are con considered to be included in the next step of the works presented.

NOMENCLATURE

α_S	Sun altitude angle over horizontal plane.
γ_S	Sun azimuth angle measured from South, in the clockwise direction.
γ_C	Collector azimuth angle measured from South in the clockwise direction.
γ	$(\gamma_S - \gamma_C)$.
Ir	Reflector tilt angle relative to horizontal plane.
Ic	Collector tilt angle relative to horizontal plane.
W_C	Collector width.
ϵ	Rate between the reflector width and the collector width.
L_C	Collector length.
σ	Rate between the reflector length and the collector length.
θ_{IR}	Incident angle formed by the source of light over the reflector`s surface.
θ_{IC}	Incident angle of the reflected ray over the collector.
X_R	Coordinate X´ on the reflector plane.
Y_R	Coordinate Y´ on the reflector plane.
Zc	Coordinate Z" in the collector plane.
Yc	Coordinate Y" in the collector plane.

ACKNOWLEDGEMENTS

This work constitutes a partial result of the internal project FING-04-012 entitled "Study of PV modules under low concentrator system" and has been presented on the 19th European Photovoltaic Solar Energy Conference (June 2004, Paris, France) under financial support of Faculty of Engineering / Autonomous University of Yucatan (UADY).

REFERENCES

[1] T. Matsushima, T. Setaka and S. Muroyama, *Solar Energy Materials and Solar Cells* 75 (2003) 603.

[2] B. Perers, B. Karlsson, M. Bergkvist, *Solar Energy* 53 (1994) 215.

[3] R. Mats, B. Karlsson, *Solar Energy* 65 (1999) 343.

[4] J. E. Braun, J. C Mitchell, *Solar Energy* 31 (1983) 439.

[5] M. C. Harris; M. E. Sydney; T. E. Irving, *Solar Energy Systems Design*. John Wiley & Sons, (1985) 106.