DOI: 10.48011/asba.v2i1.1102

SHADOW MODELLING ALGORITHM FOR PHOTOVOLTAIC SYSTEMS ANALYSIS AND SIMULATION

Bárbara A. de Sá*, Tiago Dezuo*, Douglas Ohf*

*Electrical Engineering Department, Santa Catarina State University, Joinville/SC, Brazil.

Emails: bah.azevedo99@gmail.com, tiago.dezuo@udesc.br, douglas_ohf@hotmail.com

Abstract— In this paper, an algorithm capable of modelling shadows from nearby obstructions onto photovoltaic arrays is proposed. The algorithm developed is based on the calculation of the solar position in the sky for any given instant in order to obtain the shadow projection for any object point. The convexity properties of objects and their shadows are used to allow a precise three-dimensional solution with reduced computational power without the need to consider a vast grid of points. The idea is extended to provide the shading patterns for a desired range of time and to calculate the efficiency rate of the irradiation power incident on the array in comparison to the non-shadowed case. The algorithm has interesting applications, such as optimizing array positioning and orientation, evaluating the impact of new obstructions on pre-existing array installations, allowing precise and practical data for control strategies and MPPT techniques for partially shaded systems, calculating more realistically constrained payback scenarios and finding the optimal PV array interconnection. The results obtained are illustrated by a numerical example, in which the effects of a nearby building in the irradiation received by a photovoltaic array throughout the year is analyzed.

Keywords— Photovoltaic systems, Solar position, Shadow modelling, Algorithm, Efficiency.

Resumo— Neste artigo, é proposto um algoritmo capaz de modelar sombras de obstruções próximas em arranjos fotovoltaicos. O algoritmo desenvolvido é baseado no cálculo da posição solar no céu em um determinado instante, a fim de obter a projeção da sombra para qualquer ponto de um objeto. As propriedades de convexidade de objetos e suas sombras são usadas para permitir uma solução tridimensional precisa com poder computacional reduzido, sem a necessidade de considerar uma vasta rede de pontos. A idéia é estendida para fornecer os padrões de sombreamento durante um intervalo de tempo desejado e calcular a taxa de eficiência da potência de irradiação incidente no arranjo em comparação com o caso não sombreado. O algoritmo possui aplicações interessantes, como otimizar o posicionamento e a orientação do arranjo, avaliar o impacto de novas obstruções em instalações pré-existentes, permitir dados precisos e práticos para simulação de estratégias de controle e técnicas de MPPT para sistemas parcialmente sombreados, calcular cenários de *payback* com restrições mais realistas e encontrar a interconexão ideal do arranjo fotovoltaico. Os resultados obtidos são ilustrados através de um exemplo numérico, no qual são analisados os efeitos de um edifício próximo na irradiação recebida por um arranjo fotovoltaico ao longo do ano.

Palavras-chave— Sistemas fotovoltaicos, Posição solar, Modelo de sombreamentos, Algoritmo, Eficiência.

1 Introduction

Photovoltaic (PV) generation is gaining more market space and investments around the world, mostly because of its decreasing costs through the development of new materials and techniques. Moreover, it is a clean energy source, thus it is more ecologically advantageous than other forms of energy generation as, for instance, those that use fossil fuels. The current studies in the area are focused on increasing the efficiency of every aspect of the PV systems, in order to make the most of the irradiation incidence for even more worthwhile systems. This is done by improving the module materials (Morgera and Lughi, 2015), the power electronic converters (Shi et al., 2018), control techniques (Dezuo et al., 2017) and Maximum Power Point Tracking (MPPT) methods (Soualmia and Chenni, 2016). Although, the literature is scarce regarding the installation environment analysis, which may have an enormous impact on the efficiency of PV generation.

When inspecting PV modules installed in locations close to buildings, trees and other obstacles, it is noted that these modules may suffer from periodic partial/total shading throughout the day and the year. This is more often the case in urban environments, which undergo though fast and unforeseen structural changes as they grow, and sometimes unusual module installation strategies have to be considered to overcome the irregularity of the surfaces levels and distribution, such as modules installed in building facades (Freitas, 2018). For PV arrays connected to single-converter topologies, the problem of partial shading is worse than just losing a proportion of solar energy, as it may result in power consumption by shaded PV cells and/or multiple local maximum power points, which prevents the maximum power of each PV module to be achieved simultaneously. This causes different PV array interconnections to have different maximum efficiencies for a given shading pattern (Squersato et al., 2019).

There are several tools available for solar radiation modelling, with different purposes and characteristics, as shown in the survey (Freitas et al., 2015). Empirical solar radiation models can be obtained from the global and diffuse irradiation data from meteorological stations. This data can be used to approximate irradiation received by other particular surfaces by means of in-

terpolation, geometrically-based formulations and transposition (Angelis-Dimakis et al., 2011). Although, empirical models fail to take the local environment into account. Computation based models, on the other hand, can use phenomenological modelling, Geographical Information System (GIS) tools and web-based solar maps. Most of these methods are used to analyse solar potential on cityscapes at a macroscale. After great advancement in 2D solar mapping, GIS tools are now progressively adapting to detailed 3D representation and spatial analysis, which is not a trivial task (Zlatanova, 2000) and is yet to reach its consensus (Freitas et al., 2015). There are several commercial purposed softwares used for array positioning, electrical components design and payback evaluation. Although, these lack to consider more detailed effects of periodic partial shading and may result in unrealistic economic estimations. Most of these softwares rely on satellite imagery available online, which falls on the precision issues of 3D GIS tools (Freitas et al., 2015). On the other hand, some of the softwares perform a detailed surroundings modelling, although by using very expensive equipment such as drones and advanced cameras.

Motivated by the need for improving the overall efficiency of PV systems at a local level, this paper presents a simple, yet effective, algorithm for modeling the shadowing effects of obstructions surrounding PV arrays. The proposed algorithm is intended to be an open-sourced reasonable 3D local shadowing analysis that do not require sophisticated environment sensing or time consuming irradiation measurements. In the absence of a more detailed irradiation profile for the location, the user may insert the average daily solar irradiation for the region. This information is promptly available from weather stations online. The use of the average is also useful to avoid measurements from atypical days. It takes into account any user-defined time interval, up to the complete cycle of a year, allowing a complete prediction of energy losses due to shadowing. The method is based on an accurate solar position calculations and uses the properties of convex objects[1] and their shadows for identifying the affected areas of the PV arrays and reducing the computational power demand. The algorithm speed can be further adjusted by varying the spatial complexity and time precision required.

Applications of the proposed method include **(i)** optimizing PV array positioning and orientation; **(ii)** evaluating the impact of new obstructions on pre-existing PV installations; **(iii)** allowing precise and practical data for control strategies and MPPT techniques for partially shaded systems; **(iv)** calculating more realistically constrained payback scenarios. **(v)** finding the optimal PV array interconnection.

The paper is organized as follows. This section ends with the symbols, basic functions and notation used in the paper. The next section presents solar position calculation method. The convexity based shadow projection is presented in the Section 3. Section 4 is devoted to present the main algorithm for the effects of shadow on the PV array considering time-lapse and efficiency calculation. The results are numerically illustrated in Section 5 and some concluding remarks end the paper.

Base Functions. TimeConversion$(\cdot)$ converts from a HH:MM format of time to a real number (in minutes). Date2Day$(\cdot)$ returns the number of the day within the year for a given DD/MM/YYYY format date. Pol2Cart(A, R) transforms corresponding elements of data stored in polar coordinates (angle A, radius R) to Cartesian coordinates X, Y. ConvHull$(\cdot)$ computes the convex hull (outermost vertices) from a given set of points. Normal$(\cdot)$ returns the normal vector of a surface characterized by at least the non collinear points in $(\cdot)$.

Symbols. Altitude angle (α); Azimuth angle (β); Equation of time (E); Number of the day (N); Time Correction (T_C); Latitude (L_A); Longitude (L_O); Local Solar Time (L_{ST}); Local Solar Time Meridian (L_{STM}); Local Time (L_T); Hour Angle (H_A); Declination (δ); North axis (x); East axis (y); Height axis (z); Displacement in the x axis (Δx); Displacement in the y axis (Δy); Object points (O); Number of object vertices (n_v); Projection points (P); Shadowed region vertices (S_r); Array points (C); Shadowed status matrix (M); Number of rows (n_r); Number of columns (n_c); Augmented region (A_r); Discrete time iteration (k); Number of rows (n_r); Number of columns (n_c); Time increment (ΔT); Final date (D_f); Final time (T_f); Illuminated proportion (η); Relative irradiation angle (θ); Irradiation direction ($\overrightarrow{i}$); Normal vector ($\overrightarrow{n}$); Undisturbed solar irradiation (P_{sun}); Power reaching the array (P_{arr}); Energy lost to shadowing (J_{lost}); Computation time (C_T).

Notation. $x(\cdot)$, $y(\cdot)$, $z(\cdot)$ are, respectively the x, y, z coordinates of a given vector $(\cdot)$. $\cup$ represents the union of sets. $|\cdot|$ represents the absolute value of a number. $\overrightarrow{a} \bullet \overrightarrow{b}$ stands for the inner product between the vectors $\overrightarrow{a}$ and $\overrightarrow{b}$. $M[k](i, j)$ represents the element (i, j) of the k-th matrix of a set M.

2 Solar Position Calculation

In this paper we are interested in calculating shadow projections over PV arrays and the first step is to obtain a numerical model for the spatial

[1] An object is convex if a line connecting any two points inside it is also completely inside it.

orientation of the irradiation. This methodology was chosen due to the predictive regularity of the position of the Sun in relation to any point on the Earth's surface. The Sun's position can be defined in spherical coordinates by the altitude and azimuth angles, which can be obtained simply by geometric calculations that are a function of time. The altitude angle α specifies the solar elevation in relation to the Earth's surface and the azimuth angle β is the angle on the surface plane in relation to the north in a clockwise direction, as illustrated in Figure 1.

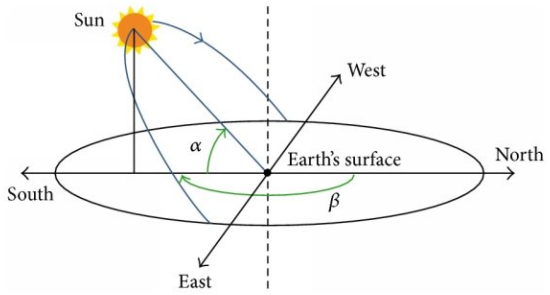


Figure 1: Illustration of the altitude angle α and the azimuth angle β (Ferdaus et al., 2014).

Taking into account that the Earth travels in an elliptical orbit around the Sun with an eccentricity around 0.016 and its rotation axis is tilted 23.45 degrees (the obliquity of the ecliptic), several effects exert influence over α and β throughout an year. The astronomic equations for the relevant effects are known and the procedure for obtaining the altitude and azimuth angles is as follows.

First, in order to reconcile the difference between the apparent solar time (sundial time) and the mean solar time (clock time), caused by the non-zero eccentricity and obliquity, it is necessary to calculate the concept known as *Equation of Time*. This equation is the sum of asynchronous sinusoidal curves, with periods of six months and one year, and can be approximated by the following expression (Milne, 1921):

$$E = 9.87\sin(2B) - 7.53\cos(B) - 1.5\sin(B)\,, \quad (1)$$

where E is the Equation of Time expressed in minutes and, for the sin and cos expressed in degrees,

$$B = \frac{360^\circ\,(N-81)}{365}, \quad (2)$$

where N is the number of the day, starting in January 1^{st} ($N = 1$).

The net Time Correction (T_C) factor (in minutes) accounts for the variation of the Local Solar Time (L_{ST}) within a given time zone due to the longitude variations within the time zone and also incorporates the E above:

$$T_C = 4(L_O - L_{STM}) + E, \quad (3)$$

where the factor of 4 minutes comes from the fact that the Earth rotates 1° every 4 minutes and the Local Standard Time Meridian (L_{STM}) is given by

$$L_{STM} = 15^\circ T_Z, \quad (4)$$

where $15^\circ = {}^{360^\circ}/_{24}$ is the arc length of a Time Zone (T_Z) and T_Z is the time offset from the Coordinated Universal Time (UTC) in hours.

Finally, the Local Solar Time (L_{ST}) can be found by using the previous two corrections to adjust the Local Time (L_T)

$$L_{ST} = L_T + \frac{T_C}{60} \quad (5)$$

and the Hour Angle (H_A) that converts L_{ST} into the number of degrees which the sun moves across the sky is

$$H_A = 15^\circ\,(L_{ST} - 12)\,. \quad (6)$$

Now we can determine the altitude and azimuth angles (in degrees) by the geometric formulas

$$\alpha = 90^\circ - \cos^{-1}\Big(\cos(L_A)\cos(\delta)\cos(H_A) + \sin(L_A)\sin(\delta)\Big), \quad (7)$$

$$\beta = 180^\circ - \cos^{-1}\left(\frac{\sin(L_A)\sin(\alpha) - \sin(\delta)}{\cos(\alpha)\cos(\delta)}\right), \quad (8)$$

where L_A is the local latitude and δ is the declination angle given by

$$\delta = -23.45^\circ \cos\big(2\pi(N+10)/365\big). \quad (9)$$

The solar position calculation is summarized in Function 1, where it is implied that Equations (1)-(6) and (9) are used to calculate (7)-(8).

Function 1 Solar Position Calculation

```
Data: L_A, L_O, T_Z
function SolarPosition(D_ate, T_ime)
    N ← Date2Day(D_ate)
    L_T ← TimeConversion(T_ime)
    α ← Equation (7)
    β ← Equation (8)
    return α, β
```

In order to consider daylight saving time, it suffices to subtract 1 hour from the input T_{ime}. For leap years, divide by 366[2] instead of 365 in Equations (2) and (9). Alternatively, one may consider the astronomical year (synchronized with the seasons) which has approximately 365.25 days, instead of the calendar year.

[2]This has slightly different results for leap years, imperceptible in practice.

3 Shadow Projection

With the solar position information, it is now possible to cast a projection of any given three-dimensional spatial coordinate onto the Earth's surface level. If there is an object on the given coordinate, then there is a shadow in every point that lies in the line segment between the coordinate and its projection.

The first step is to define the origin $(x, y, z) = (0, 0, 0)$, where x is the North direction, y is the East direction and z is the height from the surface. While the origin of x, y can be arbitrarily chosen, $z = 0$ must be the Earth's surface level. Next, define $O_i \subseteq O$ as a vector containing the three-dimensional coordinates (x, y, z) for the i-th object point, where O is the set of object points considered. Now recall from Figure 1 that, from the point of view of a projection point, the object has the same direction angles α and β as the Sun. That is, the object point has a β angle (on the surface) in relation to its projection and there is a horizontal distance (also on the surface) from the x, y coordinates of the object to it x, y coordinates of the projection given by $|z(O_i)\tan(\alpha)|$. Thus the projection coordinates can be obtained from Function 2.

Function 2 Projection

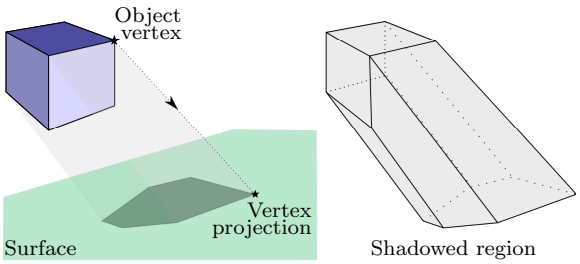
```
1: function PROJECTION(O_i, α, β)
2:     (Δx, Δy) ← POL2CART(β, |z(O_i) tan(α)|)
3:     P_i ← (x(O_i) + Δx, y(O_i) + Δy, 0)
4:     return P_i
```

However, solid objects are comprised of an infinite number of points and it would not be computationally viable to consider all of them. Also, considering a discrete grid of points, it would still be difficult to evaluate if a particular point on a module is shadowed if it is not intersected by the resulting set of line segments.

The solution we propose for this problem is to use the convexity properties of shadows. The idea is to consider only the outermost vertices of solid and convex objects, i.e. the convex hull of the object. A projection can be cast on the surface for each vertex of the 3D object, as shown in Figure 2(a). Note that the vertices of the shadow are projections of some (not all) of the objects vertices. For instance, in Figure 2(a) a cube (8 vertices) results in a shadow projection with only 6 vertices.

Due to convexity, any point belonging to the 3D convex hull of the object casts a projection that belongs to the convex hull of the vertices projections. Finally, all the points that belong to the convex hull of the union of the vertices of object and of the projected shadow are inside the shadowed region, as shown in Figure 2(b).

The shadowed region determination is summarized in Function 3, where S_r is the set of vertices of the shadowed region, n_v is the number of object vertices, $P_i \subseteq P$ is a vector containing the three-dimensional coordinates x, y, z for the i-th object vertex and P is the set of projection points.



(a) (b)

Figure 2: (a) Example of a solid and convex object casting a shadow. (b) Convex hull of the object vertices and their projections.

Function 3 Shadow Region

```
1: Data: O, n_v
2: function SHADOWREGION(α, β)
3:     for i ← 1 to n_v do
4:         P_i ←PROJECTION(O_i, α, β)
5:     S_r ←CONVHULL(O ∪ P)
6:     return S_r
```

The use of the convex hull approach results in less computation time, eliminating irrelevant points from the analysis. The early calculation of the possible negative altitude angles with Function 1 can also be used to avoid computations for night times in Function 3.

The shadow projection obtained from Function 3 considers only direct sunlight. It is assumed that atmosphere reflection and refraction, reflections on the ground and other objects and gradient shadow borders due to the Sun's diameter are neglected.

4 Complete Environment Modelling

The surroundings environment is composed of the modules, their orientation, the space and obstructions around them and the shadows projected onto them. The latter varies according to the time of the day and the day of the year due to the solar position. Is this section we are interested in modelling the effects of shading in the PV array part of the environment and also analyzing the cumulative effects of time on the overall efficiency. By efficiency, in this context, we mean how much solar energy is reaching the array in comparison with the non-shaded case.

First, the module vertex coordinates are also informed to the algorithm by specifying its four vertices or at least one vertex accompanied by

the panel dimensions, orientation and tilted angle. With this information, the algorithm divides the array semi-plane into a two-dimensional grid of points, composing: **(i)** a structure C containing the coordinates of each point on the module; **(ii)** a matrix M in which each element is a binary number represents the status of each point (0 for shaded and 1 otherwise). The number of rows (n_r) and columns (n_c) of these matrices can be defined by the user for a compromise between computational power required and precision of the method.

The idea is then to test each point to check if it is inside the shadowed region obtained by Function 3. This can be done simply by verifying if convex hull of S_r changes when that particular point is included in the set. If it changes, then the point is not inside the convex hull and, therefore, it is not shaded by that particular object. This is summarized in Function 4.

Function 4 Array testing

```
1:  Data: C, n_r, n_c
2:  function ShadowTest(S_r)
3:      for i ← 1 to n_r do
4:          for j ← 1 to n_c do
5:              A_r ← ConvHull(S_r ∪ C(i, j))
6:              if A_r = S_r then
7:                  M(i, j) ← 0
8:              else
9:                  M(i, j) ← 1
10:     return M
```

The procedure presented so far is used to determine the shadowed parts of a PV array but only for a given instant, that is, for a particular solar position. Although, little can be concluded by the efficiency of a system regarding shadow influence as it will change over time due to the effects discussed in Section 2. For this reason it is interesting that the analysis can cover a long range of the time as a day or, better yet, a year, which is sufficient for a complete solar cycle.

This is done by a loop in the time lapse algorithm presented in Function 5, where the user can define any initial value for D_{ate} and for T_{ime} and increment it with ΔT minutes until the end of the desired interval (D_f, T_f) is reached. The size of ΔT also has to be specified keeping in mind a compromise between accuracy and computation time. The date and time variables are provided to the algorithm using, respectively, the DD/MM/YYYY and HH:MM formats due to the easier relation with times of the day, year and seasons, for instance.

The procedure in Function 5 returns the shadow status matrix M, $\forall k \in \{1, \ldots, n_k\}$, where n_k is the final number of time iterations, thus the notation $M[k]$. With this, it is possible to calculate the percentage of solar incidence that is harnessed due to the cumulative effect of all shadow patterns over the PV array and thus assert about the efficiency. The first step is to calculate the proportion η of the array that receives sunlight at every tested instant, as in Function 6.

Function 5 Time lapse

```
1:  Data: D_ate, T_ime, D_f, T_f, ΔT
2:  function TimeLapse
3:      k ← 1
4:      while D_ate < D_f or T_ime < T_f do
5:          (α, β) ← SolarPosition(D_ate, T_ime)
6:          S_r ← ShadowRegion(α, β)
7:          M[k] ← ShadowTest(S_r)
8:          k ← k + 1
9:          T_ime ← T_ime + ΔT
10:         if T_ime > 24 : 00 then
11:             T_ime ← T_ime − 24 : 00
12:             D_ate ← D_ate + 01/00/0000
13:     return M[k]
```

Function 6 Illuminated proportion of the array

```
1:  Data: n_k, n_r, n_c
2:  function IlluminatedProportion(M[k])
3:      η ← 0
4:      for k ← 1 to n_k do
5:          for i ← 1 to n_r do
6:              for j ← 1 to n_c do
7:                  η[k] ← η[k] + M[k](i, j)
8:      η[k] ← η[k]/(n_r n_c)
9:      return η[k]
```

However, the amount of solar power (in Watts) that reaches the panel, also changes during the day and year due to the orientation of the panel, in case there is no Sun tracking device. This is because there is a certain amount of power per square meter around the Sun, thus the larger the area facing the Sun, the more power it receives. In other words, if the normal vector of the panel surface is parallel to the irradiation direction, then it receives the maximum irradiation available. If the angular difference (θ) between the normal vector ($\overrightarrow{n}$) and the irradiation direction ($\overrightarrow{i}$) is between 0° and 90°, the apparent area is proportional to $\cos(\theta)$. In case $\theta > 90°$, the solar position is "behind" the panel and, therefore, no solar power is received.

The vector that characterizes $\overrightarrow{i}$ can be obtained from the line segment between any arbitrary object point from O and its respective projection from P. In order to obtain $\overrightarrow{n}$, at least three non collinear panel points belonging to C are necessary to characterize the surface. Finally, the relative irradiation angle θ is expressed by

$$\theta[k] = \cos^{-1}\left(\frac{\overrightarrow{n} \bullet \overrightarrow{i}[k]}{\|\overrightarrow{n}\| \cdot \|\overrightarrow{i}[k]\|}\right) \qquad (10)$$

and the algorithm for determining it is in Function 7.

Function 7 Relative irradiation angle

```
Data: O, C, P[k], n_k
function RelativeAngle
    for k ← 1 to n_k do
        n⃗ ← Normal(C)
        i⃗[k] ← O_1 − P_1[k]
        θ[k] ← Equation (10)
    return θ[k]
```

To perform the calculations related to the efficiency of the module, let us suppose the solar irradiation that reaches the Earth's surface (P_{sun}) is constant, that is, we are neglecting the effects of different air masses absorption throughout the day. It is sufficient to multiply both power decreasing effects (shadowing and misalignment) to obtain the power reaching the array (P_{arr}). Thus,

$$\begin{cases} P_{arr}[k] = P_{sun} \cdot \eta[k] \cdot \cos(\theta[k]), & \text{if} \quad \theta[k] < 90°, \\ P_{arr}[k] = 0, & \text{if} \quad \theta[k] \geq 90°. \end{cases} \tag{11}$$

The resulting $P_{arr}[k]$ can be used for all the objectives of this paper, such as efficiency analysis and control simulations. Finally, the instantaneous efficiency between the shaded and non shaded cases is simply given by $\eta[k]$.

5 Illustrative Example

For this numerical example, consider the environment composed by a building and a photovoltaic panel spatially disposed as in Figure 3, where the coordinates are expressed in nonspecific distance units. This configuration emulates a real scenario. The origin $(x, y) = (0, 0)$ defined in the x, y position of one of the vertices of the PV array, which has a 100×100 dimension and is inclined according to the latitude angle. The exact object and array coordinates are given in Tables 1 and 2, respectively.

Table 1: Object coordinates

O	x	y	z
O_1	500	200	0
O_2	500	100	0
O_3	300	200	0
O_4	300	100	0
O_5	500	200	300
O_6	500	100	300
O_7	300	200	300
O_8	300	100	300

Let us consider the time precision as $\Delta T = 5$ minutes, and the array matrix with size $n_r = 10$, $n_c = 10$, that is 100 points. Also, assume the local solar irradiation is constantly $P_{sun} = 1000$ W/m^2. Consider that the system is located in Joinville/SC, which has latitude and longitude, respectively, of approximately $-26°$ and $-48°$ and $T_Z = -3$ hours.

Table 2: Array coordinates

C	x	y	z
C_1	0	0	$100\sin(\lvert L_A \rvert)$
C_2	0	100	$100\sin(\lvert L_A \rvert)$
C_3	$100\cos(\lvert L_A \rvert)$	0	0
C_4	$100\cos(\lvert L_A \rvert)$	100	0

The code[3] is implemented in Matlab, which makes it easily adaptable for converter control and MPPT simulations using this software. A one year $P_{arr}[k]$ evaluation starting 01/01/2019 at 00:00 was performed.

First, let us compare two particular dates: one of the days with almost perfect alignment and no shadow (30/09/2019) and the day with the most aggressive misalignment and shadowing (22/06/2019). Figure 3 depicts these situations in a simple interface build to visualize the time lapse and for the test phase of the algorithm. Figure 4 shows the effects of these characteristics on $P_{arr}[k]$. Also note the slight horizontal offset between the curves due to the Equation of Time.

Now, the results for the entire year can be seen in Figure 5, where the misalignment effect is clear on the sinusoidal amplitude of the daily power peaks. Moreover, Figure 6 shows the percentage of energy (in Joules) is lost due to shadows only, reaching up to 8.676% for this configuration. This is a very significant value as power electronic researchers work hard for improvements on a 1% scale.

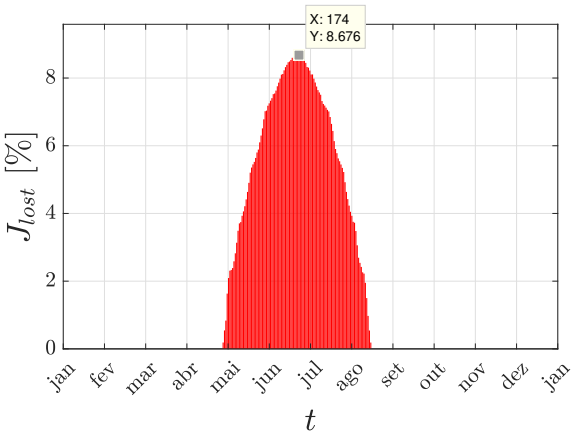


Figure 6: Percentage of energy lost due to shadowing at each day of the year.

Finally, the computation time (C_T) for different cases of time precision ΔT is presented in

[3] Thank (Mikofski, 2020) for a valuable piece of code for the solar position calculation.

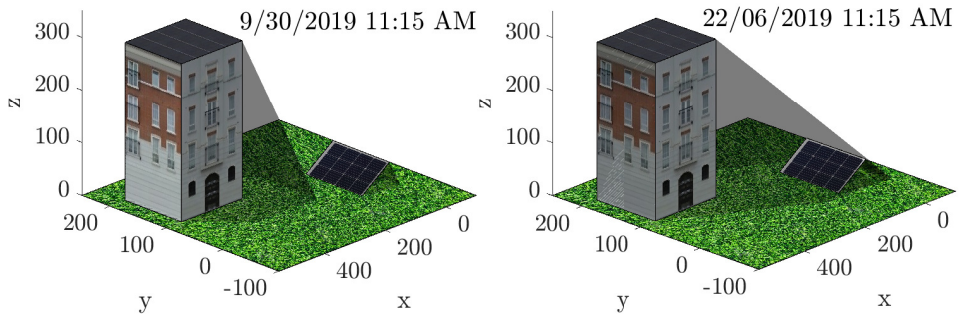


Figure 3: Experimental graphical interface showing the shading pattern on two different dates.

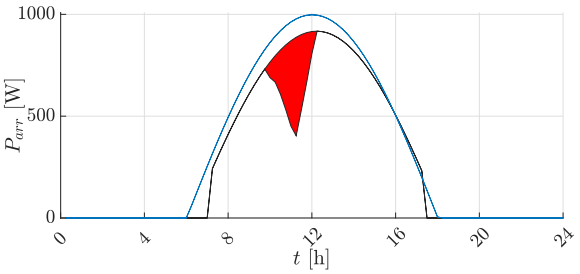


Figure 4: Power received by the array on 30/09/2019 (blue curve) and on 22/06/2019 (black curve). The area in red color represents the energy lost due to shadows.

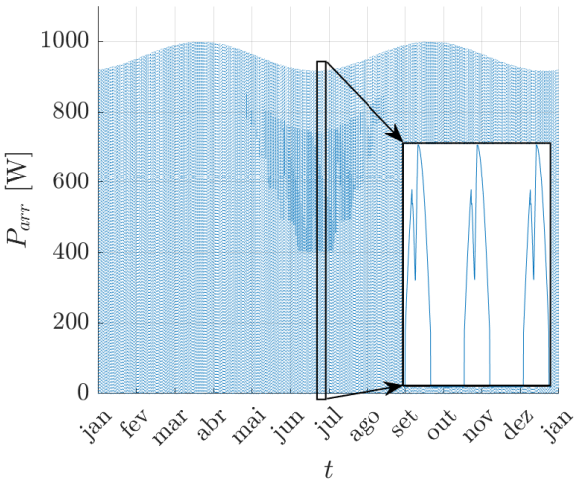


Figure 5: Complete year analysis.

Table 3. These results were obtained with a Samsung notebook running a 64 bits Windows 10 on Intel Core i7, 2.40 Ghz, 8 GB of RAM. It is noted that the time increases almost exponentially for higher precisions. Although, the less precise case can be used without significant difference on the results as shadows often do not undergo very fast variations.

6 Concluding Remarks

This paper proposed an algorithm capable of modelling shadows from nearby obstructions onto photovoltaic arrays. The strategy was based on the phenomenological model of the shadow by geometry of the solar position and by the convexity properties of convex and solid objects. The efficacy and some of the advantages were illustrated through a numerical example. Moreover, non convex obstructions can also be considered by simply dividing the object into multiple convex sub-objects. For instance, a non-convex L-shaped object could be divided into two convex rectangular boxes.

Table 3: Computation times

ΔT (minutes)	1	5	10	15	20
C_T (seconds)	4888	919	473	339	229

As future work, some possibilities are to improve the algorithm by including the influence of reflection, refraction and absorption by the air mass, the ground and nearby objects. Another considerable improvement would be to consider non-binary shading levels and, consequently, multiple shading layers with different intensities. Although, it expected that the contribution of these inclusions in the solar power received are very small. Another suggestion is to include an optimal analysis to indicate the best spacial orientation and location for an array that would minimizes the influence of shadowing in a constrained environment. Finally, considering the PV array also as a (2D) convex object to calculate the intersection with the shaded region could accelerate the algorithm computation by avoiding the quadratic growth of the two-dimensional matrix M for higher precision. Although, calculating the intersection between polytopic regions in a practical manner is still an open problem.

Acknowledgements

This study was financed in part by FAPESC through the process 873/2019, by FNDE through the *Programa de Educação Tutorial* from SESu/MEC, Brazil and by the TIM Institute through the TIM-OBMEP scholarship.

References

Angelis-Dimakis, A., Biberacher, M., Dominguez, J., Fiorese, G., Gadocha, S., Gnansounou, E., Guariso, G., Kartalidis, A., Panichelli, L., Pinedo, I. and Robba, M. (2011). Methods and tools to evaluate the availability of renewable energy sources, *Renewable and Sustainable Energy Reviews* **15**(2): 1182–1200.

Dezuo, T., Lunardi, H. and Trofino, A. (2017). Robust switching rule design for photovoltaic systems under non-uniform conditions, *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Melbourne, Australia, pp. 2342–2347.

Ferdaus, R., Mohammed, M. A., Rahman, S., Salehin, S. and Mannan, M. (2014). Energy efficient hybrid dual axis solar tracking system, *Journal of Renewable Energy* **2014**.

Freitas, S. (2018). *Photovoltaic Potential in Building Façades*, PhD thesis, Universidade de Lisboa, Lisboa, Portugal.

Freitas, S., Catita, C., Redweik, P. and Brito, M. C. (2015). Modelling solar potential in the urban environment: State-of-the-art review, *Renewable and Sustainable Energy Reviews* **41**: 915–931.

Mikofski, M. (2020). Solar position calculator, `https://www.mathworks.com/matlabcentral/fileexchange/58405-solar-position-calculator`, MATLAB Central File Exchange. Retrieved February 21, 2020.

Milne, R. M. (1921). Note on the equation of time, *The Mathematical Gazette* **10**(155): 372–375.

Morgera, A. F. and Lughi, V. (2015). Frontiers of photovoltaic technology: A review, *2015 International Conference on Clean Electrical Power (ICCEP)*, pp. 115–121.

Shi, F., Li, R., Yang, J. and Yu, W. (2018). High efficiency bidirectional dc-dc converter with wide gain range for photovoltaic energy storage system utilization, *2018 IEEE International Power Electronics and Application Conference and Exposition (PEAC)*, pp. 1–6.

Soualmia, A. and Chenni, R. (2016). A survey of maximum peak power tracking techniques used in photovoltaic power systems, *IEEE Future Technologies Conference 2016* pp. 430–443.

Squersato, I., Lunardi, H. C. and Dezuo, T. J. M. (2019). Algoritmo para simulação e análise de arranjos fotovoltaicos reconfiguráveis sob condições não uniformes, *Anais do 14° Simpósio Brasileiro de Automação Inteligente*, Ouro Preto, Brazil, pp. 1783–1788.

Zlatanova, S. (2000). On 3d topological relationships, *Proceedings 11th international workshop on database and expert systems applications*, pp. 913–919.