mathematics

*Article*

# Optimal Orientation of Solar Panels for Multi-Apartment Buildings

**Sungha Yoon [1], Jintae Park [2], Chaeyoung Lee [2], Sangkwon Kim [2], Yongho Choi [3], Soobin Kwak [2], Hyundong Kim [4] and Junseok Kim [2,*]**

1 Institute of Mathematical Sciences, Ewha Womans University, Seoul 03760, Republic of Korea
2 Department of Mathematics, Korea University, Seoul 02841, Republic of Korea
3 Department of Computer & Information Engineering, Daegu University, Gyeongsan-si 38453, Republic of Korea
4 Department of Mathematics, Gangneung-Wonju National University, Gangneung 25457, Republic of Korea
* Correspondence: cfdkim@korea.ac.kr

**Abstract:** In this paper, we present a mathematical algorithm for the optimal orientation of solar panels for multi-apartment buildings. Currently, photovoltaic power generation has increasingly become an effective method. It has the advantage of not causing environmental pollution; however, it has the disadvantage of relatively low power generation efficiency. To increase the power efficiency of the panel, one can consider a rotation. However, if there is a limitation to the rotation angle of the solar panel, especially in multi-apartment buildings, it is desirable to install the panel at the optimal angle under given constraints. Therefore, we present a simple and practical method to evaluate the optimal installation angle of the panel. Using the proposed method, it is easy to find an optimal installation angle to achieve the best power generation efficiency based on the latitude and azimuth angles. To demonstrate the effectiveness of the proposed algorithm, several numerical simulation results are provided.

**Keywords:** solar energy; solar panel; optimal orientation; mathematical modeling; code implementations

**MSC:** 49M25; 65K10; 65Z05



**Citation:** Yoon, S.; Park, J.; Lee, C.; Kim, S.; Choi, Y.; Kwak, S.; Kim, H.; Kim, J. Optimal Orientation of Solar Panels for Multi-Apartment Buildings. *Mathematics* **2023**, *11*, 938. https://doi.org/10.3390/math11040938

Academic Editors: Arturo Hidalgo, Lourdes Tello and Duarte Valério

Received: 25 November 2022
Revised: 2 February 2023
Accepted: 10 February 2023
Published: 12 February 2023



## 1. Introduction

Solar energy is one of the important energy sources, and many countries, therefore, have realized the important role of renewable energies due to the depletion of conventional energy sources [1,2]. In particular, interest in solar power generation is increasing due to the dangers of nuclear/thermal power generation that have recently occurred in several countries. The advantages of using solar power generation are as follows. First, because it uses solar energy, it does not require fuel costs and is a clean energy source that does not generate air pollution or waste. Second, because the power generation unit is composed of semiconductor devices, there is no vibration and noise, and automation is easy. Lastly, efficient power generation is possible because power can be maximized during the daytime and summer when the power load is high.

However, photovoltaic power generation has several problems. First, there is a time constraint on energy generation. This is a very important issue and inevitably causes the problem of maximum efficiency and storage capacity in a limited time. Second, the profitability is not good due to the high unit price for power generation. For example, it is about 4 to 5 times more expensive than nuclear power generation.

In everyday life, one can often see solar panels installed in multi-apartment buildings to generate solar energy, as shown in Figure 1. An installation angle of the solar panel is important to collect photons efficiently [3]. If the solar panels are installed in a parallel direction to the building, then the energy efficiency is usually not optimal at each location.

Therefore, we propose a simple mathematical algorithm for setting the angle and direction of the solar panel that can produce the maximum solar energy efficiency at each location based on the latitude and azimuth angle under several constraints.

(**a**) (**b**)

**Figure 1.** (**a**) The actual solar panel installed in the apartment. (**b**) An enlarged view.

As there is a lot of interest in solar panels and photon collectors around the world, many researchers have investigated calculating their optimal tilt angles [4–18]. Not only the optimal angle calculation but also the research on factors that can affect the performance of solar panels such as wind speed [19], weather [20], latitude and longitude [21] are in progress. In South Africa, the authors in [5] calculated the annual solar insolation for all possible angles on fixed collectors using the data obtained from the nine measuring stations. Other studies proposed using incident beam radiation-based maximization approach [6], a latitude-based optimization approach in the Mediterranean region [7], and non-linear time-varying particle swarm optimization in Taiwan [8].

As most researchers have reported that the maximum power can be harvested when a solar panel is equipped with a sun-tracking system. Referred researches suggest that the different adjustment number of intervals from 4 to 12 to obtain maximum energy in each country: 8 times a year in Iraq [10], 12 times a year in Syria [11], and also 12 times a year but each month adjusting the tilt angle which can achieve 8% more surface radiation compared with yearly adjustment in Saudi Arabia [12]. Furthermore, the solar panel with a fixed tilt angle without representing the number of yearly adjustments got 10–25% higher irradiations with increasing latitude in USA [13], the variation of optimal tilt angle throughout a year is between 0° and 65° in Turkey [14].

In [16], the authors present several models to calculate the optimal slope of the solar panel accurately at any location in the world. Using the proposed models, the authors estimate the annual energy output along with the amount of irradiance stored per hour according to the slope of the solar panel. There is a study that calculates the optimal tilt angle of a solar panel using machine learning recently [17]. In this study, energy generation simulations of panels are performed to calculate monthly and yearly panel angles, and economic cost effects are also evaluated in real-world environments. In [18], the optimal solar panel angle is calculated for the maximum energy yield. Calculating the cosine effect of the angle of the sun and the angle of the panel, the authors present a permanent fixed optimal angle but also a four-season or monthly adjustment to improve the total annual solar energy yield. In [22], the authors investigate the arrangement of solar panels for various spaces. Solar panels can be arranged to maximize energy production in limited spaces such as rooftops of buildings. The results suggest that the suitable deployment of solar panels could increase energy production by up to 6%. In [23], the authors introduce a new term, the tolerance angle. The tolerance angles refer to the angular range of optimal solar panels that minimize economic losses. It can be useful during actually installing solar panels.

The main purpose of this paper is to present a mathematical algorithm for the optimal orientation of solar panels for multi-apartment buildings under several constraints. Let **m** and **n** be the representative of the solar beam and normal vector of the panel, respectively. Then, we consider the following optimizing problem

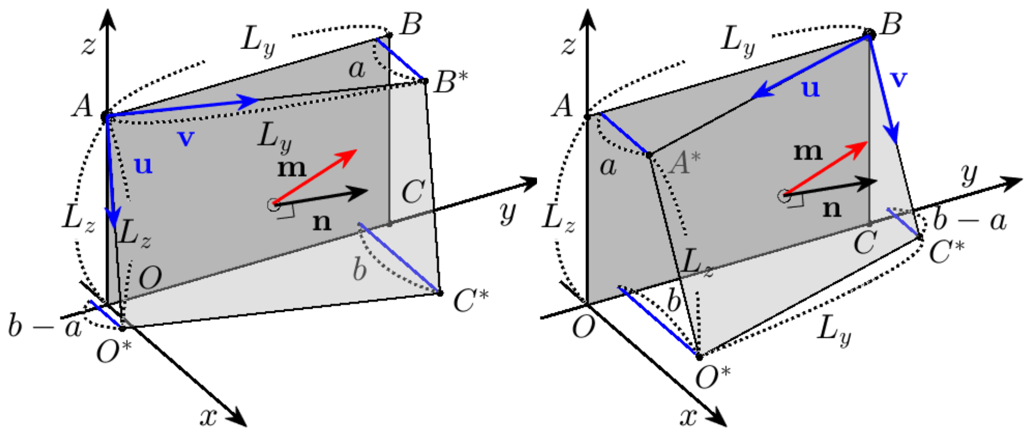
$$\underset{\mathbf{n}}{\arg\min}\, \|\mathbf{m}-\mathbf{n}\|_2\,, \tag{1}$$

when the solar vector **m** and certain constraint for **n** is given.

The outline of this paper is as follows. A mathematical model for finding the optimal position of a solar panel for buildings is described in Section 2. Numerical results are listed in Section 3. We finalize the paper with the conclusion in Section 4. A code implementation is included in Appendix A for users who may not be familiar with the content.

## 2. Mathematical Modeling

In this section, we propose our mathematical modeling to derive the optimal orientation of solar panels under several constraints. First, we consider a rectangular panel $OABC$ (represented dark gray) whose horizontal and vertical lengths are $L_y$ and $L_z$, respectively, as shown in Figure 2.




(**a**) (**b**)

**Figure 2.** Schematic illustrations of our mathematical modeling to the rotated panel, (**a**) fixing $A$ and (**b**) fixing $B$.

As shown in Figure 2a, we suppose that the panel $OABC$ is on $yz$-plane and the coordinates of $A$, $B$, and $C$ are determined automatically. Then, we move the solar panel while keeping the coordinate of $A(0,0,L_z)$ fixed. Moreover, we fix the height of panel $L_z$. We then obtain a moved solar panel $AB^*C^*O^*$, which is represented by light gray. Here, let $a$ and $b$ be the lengths of the perpendicularity from $B^*$ and $C^*$ to the plane $OABC$, respectively. We can know that the length of the perpendicularity from $O^*$ to $OABC$ is $b-a$. Note that the foot perpendicular on the plane $OABC$ from $B^*$ is a point on the plane, not $B$. Similarly, it is important to know that each foot perpendicular on the plane $OABC$ from $O^*$ and $C^*$ is also a point on the plane. By using the Pythagorean theorem, we can obtain the coordinate of $B^*$ as $(a, \sqrt{L_y^2-a^2}, L_z)$. However, we only obtain the $x$-coordinate of $C^*$ as $b$. In this case, we define the coordinate of $C^*$ as $(b,y,z)$, in which $y$ and $z$ are unknown constants. The purpose of our modeling is to find the best-fitted normal vector of the plane $AB^*C^*O^*$ to a given sunlight vector. In other words, finding $(a,b,y,z)$ is our primary goal. In our modeling, we can get the two equations from the geometry. The first equation is obtained from the fact that the length of $AC^*$ equals the diagonal of a rectangular panel. That is,

$$b^2 + y^2 + (z - L_z)^2 = L_y^2 + L_z^2\,. \tag{2}$$

Moreover, we know that the length of $B^*C^*$ is equal to the height of the panel, $L_z$, we have the following

$$(a - b)^2 + \left(\sqrt{L_y^2 - a^2} - y\right)^2 + (L_z - z)^2 = L_z^2\,. \tag{3}$$

Then, we can combine the two Equations (2) and (3) to get $y$ and $z$ in terms of $a$, $b$, $L_y$, and $L_z$ as follows:

$$y = \frac{L_y^2 - ab}{\sqrt{L_y^2 - a^2}},\ z = L_z - \sqrt{L_y^2 + L_z^2 - b^2 - y^2}\,. \tag{4}$$

Now, we get $O^* = (b - a, y - \sqrt{L_y^2 - a^2}, z)$ from the fact that it shares the midpoint of the rectangle ($O^* + B^* = A + C^*$). To find the normal vector $\mathbf{n}$ of plane $AB^*C^*O^*$, we define both vectors $\mathbf{u}$ and $\mathbf{v}$ using $A, O^*$, and $B^*$. By using the cross product of $\mathbf{u}$ and $\mathbf{v}$, we can find the normal vector $\mathbf{n}$ as follows:

$$\mathbf{n} = \frac{\left((L_z - z)\sqrt{L_y^2 - a^2}, a(z - L_z), b\sqrt{L_y^2 - a^2} - ay\right)}{L_y L_z}\,. \tag{5}$$

Similarly, we consider the case in which $B$ is fixed as shown in Figure 2b. Following the above method, we can obtain the specific coordinates of $O^*$, $A^*$, and $C^*$, which are moved points from $O$, $A$, and $C$, respectively. To sum up, let us assume that the coordinates $O, A, B$, and $C$ are given as follows:

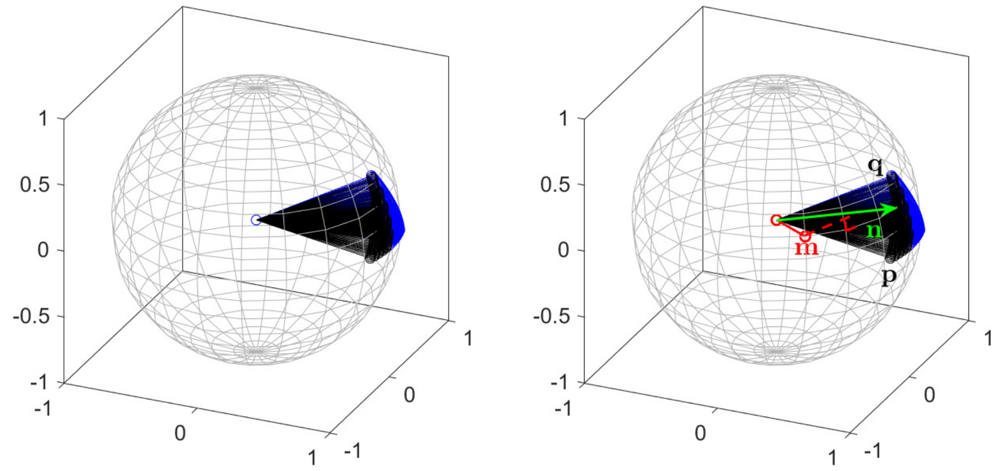
$$O = (0, 0, 0),\ A = (0, 0, L_z),\ B = (0, L_y, L_z),\ C = (0, L_y, 0)\,. \tag{6}$$

For the cases (i) fixed $A$ and (ii) fixed $B$, we can derive the remaining coordinates of the panel as follows:

$$\text{(i) } O^* = (b - a, y - \sqrt{L_y^2 - a^2}, z),\ B^* = (a, \sqrt{L_y^2 - a^2}, L_z),\ C^* = (b, y, z)\,, \tag{7}$$

$$\text{(ii) } O^* = (b, L_y - y, z),\ A^* = (a, L_y - \sqrt{L_y^2 - a^2}, L_z),\ C^* = (b - a, L_y - y + \sqrt{L_y^2 - a^2}, z)\,. \tag{8}$$

Subsequently, we need to evaluate the normal vector of the panel. In multi-apartment buildings, if an inclination angle is formed between the panel and the bottom plane perpendicular to the wall where the panel is installed less than a certain degree, it can be a nuisance downstairs due to a shade from the panel. For this reason, the angle is restricted by regulation, and we take a threshold as $\pi/3$ in this paper. Therefore, we can set the maximum value of $b$ to $L_z/2$; hence we have the constraint $0 \le a \le b \le L_z/2$ for the problem (1). Figure 3a shows the curved triangular region where all the possible normal vectors can be located. Note that we set $L_z = 1$ for simplicity.




(**a**) (**b**)

**Figure 3.** (**a**) All the possibilities of normal vector can be located. It is a curved triangular shape. Note that the black part corresponds to Figure 2a, and the blue part corresponds to Figure 2b. (**b**) Schematic illustration of finding the nearest normal vector $\mathbf{n}$, which is represented by green color, using the projection of solar beam $\mathbf{m}$, which is represented by red color, to the plane generated by any two vectors, $\mathbf{p}$ and $\mathbf{q}$, such that lie on the nearest boundary of the region to $\mathbf{m}$. Note that we normalize the magnitude of the normal vector.

There are two cases for the given vector representing a solar beam $\mathbf{m}$. One is in the curved triangular region, and the other is outside of the curved region. We just take $\mathbf{n} = \mathbf{m}$ to the former. However, one cannot take a normal vector that coincides with $\mathbf{m}$ if $\mathbf{m}$ is outside of the region. In this case, we have to choose $\mathbf{n}$ such that $\mathbf{m} \cdot \mathbf{n}$ is maximized. Such a vector can be found as normalizing the vector, which is given by the projection of $\mathbf{m}$ to the plane generated by any two vectors that lie on the nearest boundary of the region to $\mathbf{m}$. Figure 3b depicts the process of how to find the corresponding fitted normal vector to $\mathbf{m}$.

To be more precise, we can obtain the normal vector $\mathbf{n}$ as follows:

$$\mathbf{n} = \frac{\|\mathbf{n}_1\|^2\mathbf{m} - (\mathbf{m} \cdot \mathbf{n}_1)\mathbf{n}_1}{\|\|\mathbf{n}_1\|^2\mathbf{m} - (\mathbf{m} \cdot \mathbf{n}_1)\mathbf{n}_1\|}, \tag{9}$$

where $\mathbf{n}_1 = \mathbf{p} \times \mathbf{q}$. An important thing is that we can simplify the representation of boundary vectors as the relation between $a$ and $b$. Three endpoints of the triangular boundary are expressed by (i) $a = 0$, $b = L_z/2$, (ii) $a = b$, $b = L_z/2$ where $A$ is fixed, (iii) $a = b$, $b = L_z/2$ where $B$ is fixed. Thus, one can easily get the best-fitted orientation for the solar panel by following the above process.

The solar beam $\mathbf{m}$ is treated as a given constant throughout the whole process so far. The only remaining thing is, therefore, to choose an appropriate representative $\mathbf{m}$. One way to find the representative of a solar beam is by applying the mean value property to the trace of the sun. Assume that Earth is a sphere, and the orbit of Earth is a circle. Then the azimuth can be computed. Considering the spherical coordinates system while looking at the celestial body from the current position. Note that we fix the current position as Seoul, whose latitude is approximately 37.5326°. The south meridian altitude of the sun is on a great circle of spheres when the celestial body passes through the meridian of Earth. Furthermore, there are two times when the orbit of Earth and its axis of rotation are perpendicular; we call this vernal / autumnal equinox, respectively. Moreover, summer and winter solstices are parallel to the equinox. Therefore, the procedure assumes that the values between the maximum and minimum south meridian altitude have the same normal vector. To depict it clearly, we present Figure 4a as follows.

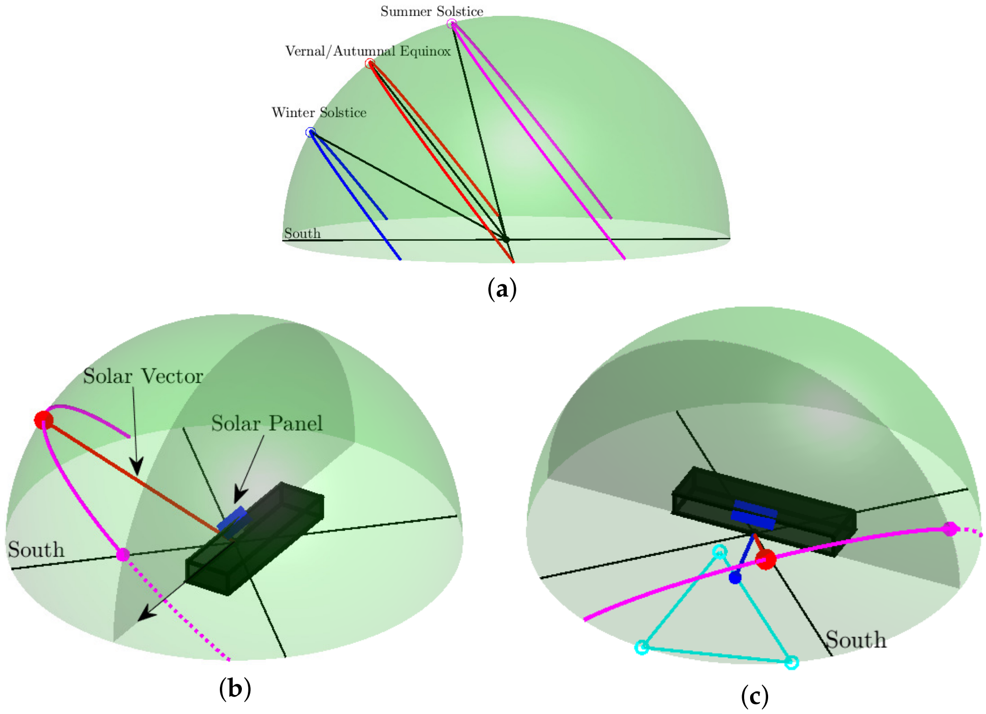



**Figure 4.** (**a**) Illustration of the trajectory of the sun when the positions of Earth are on solstice and equinox in Seoul. Note that these three traces have an identical normal vector. (**b**) Illustration of a trace of the sun (magenta) and possibly generating trajectory (solid line). Note that the dotted line is part of the trace where the sunlight does not shine due to the azimuth angle of buildings. Solar vector (red) and solar panel (blue) are represented with the shaded region, which represents a possible area to receive sunlight. The black cuboid is a schematic illustration of the residence; it is actually treated as a point (the origin) in our method. (**c**) Illustration of the solar vector (red) and the best-fitted normal vector (blue). Note that a cyan-colored triangle represents the boundary of all possibly taken normal vectors of the solar panel.

Because the normal vectors of solstice and equinox are identical, we can easily find the trajectory of average solar movement via the intersection curve with this unique normal vector. Then we can compute $\mathbf{m}$ if we know the azimuth angle of the place to install the solar panel. Suppose that we wish to install the panel in the building to a certain degree. Figure 4b depicts a trace of solar movement and the corresponding representative solar vector.

Therefore, combining all of the above conditions, we can find the best-fitted vector out of all the normal vectors that the solar panel can take along with respect to the solar vector. Figure 4c shows the evaluated solar vector and the derived best-fitted normal vector corresponding to the solar vector.

In summary, the optimal solar panel installation can be determined by knowing the south-middle altitude of the desired area and the azimuth angle of the building where the solar panel will be installed. Figure 5 depicts the procedure for finding the optimal installation specification of solar panels with respect to geometric features.

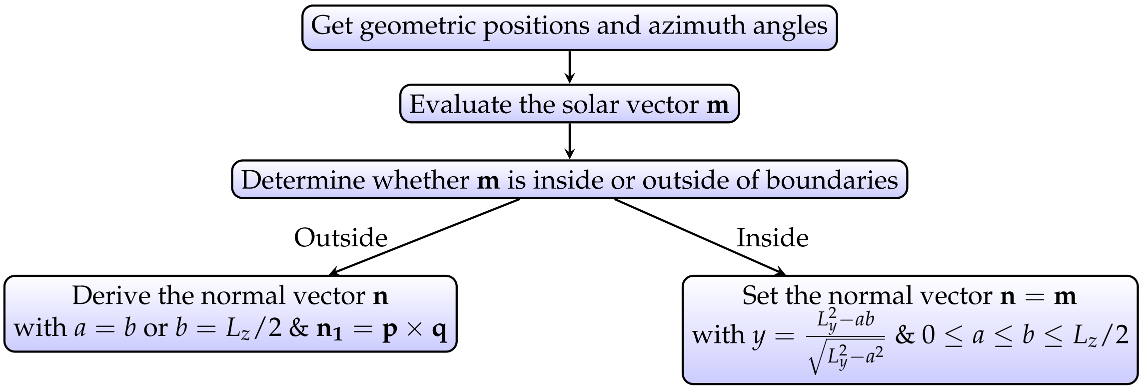



**Figure 5.** Schematic illustration to find the optimal installation specification of solar panels.

## 3. Numerical Simulation Results

We present several numerical simulation results in this section. We fix the solar panel specification as $L_y = 3$ and $L_z = 2$ because this is the common length ratio of the standard solar panel. To evaluate the solar elevation angle $\alpha$ at noon, we use the following formula [24].

$$\alpha = \arcsin(\sin\phi \sin\delta + \cos\phi \cos\delta)\,, \tag{10}$$

where $\phi$ is the latitude, and $\delta$ is the declination angle defined as

$$\delta = \arcsin\left(-\sin\Theta \cos\left(\frac{2\pi(d+10)}{365.24} + 0.0334 \sin\left(\frac{2\pi(d-2)}{365.24}\right)\right)\right), \tag{11}$$

where $\Theta \approx 23.44^\circ$ is degree of the rotation axis of Earth, and $d$ represents the number of days since January 1st in 2022. For instance, one can take $d = 3 + 9/24$ for 09:00 A.M. on January 4th 2022. Note that this approximation formula is fitted to UTC+00:00. Though there is a slight difference (up to the second decimal place) between the fixed and local time-based formula, this condition can be ignored because we consider the declination angle only up to the first decimal place. All the latitudes are sourced from [25]. Table 1 shows monthly installation specification data with respect to the monthly arithmetic average of the south altitude of each city with the azimuth angle $73\pi/72$ from the north. Note that we use the notation $(a, b, y, z)$ to represent the computed parameters hereafter.

**Table 1.** Installation specification with respect to the south meridian altitude of each city. The azimuth angle is $73\pi/72$. Note that the data is aligned by month from the top (January) to the bottom (December).

| City | Seoul | Ottawa | Dublin |
|---|---|---|---|
| $(a, b, y, z)$ | $(0.0006, 1, 2.9998, 0.2677)$ | $(0.1309, 0.9244, 2.8540, 0.1665)$ | $(0.1309, 0.6668, 2.9250, 0.0739)$ |
| | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ | $(0.1309, 0.9270, 2.8532, 0.1677)$ |
| | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ |
| | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ |
| | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ |
| | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ |
| | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ |
| | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ |
| | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ |
| | $(0, 1, 3, 0.2679)$ | $(0, 1, 3, 0.2679)$ | $(0.0999, 1, 2.9684, 0.2275)$ |
| | $(0, 1, 3, 0.2679)$ | $(0.1309, 1, 2.8271, 0.2037)$ | $(0.1309, 0.7508, 2.9045, 0.0996)$ |
| | $(0.0631, 1, 2.9796, 0.2423)$ | $(0.1309, 0.8553, 2.8755, 0.1376)$ | $(0.1309, 0.5941, 2.9406, 0.0549)$ |

Next, we present Table 2, a monthly installation specification data with respect to the monthly arithmetic average of the south altitude of each city with the azimuth angle $35\pi/36$ from the north.

**Table 2.** Installation specification with respect to the south meridian altitude of each city. The azimuth angle is $35\pi/36$. Note that the data is aligned by month from the top (January) to the bottom (December).

| City | Wellington | Copenhagen | Paris |
|---|---|---|---|
| $(a,b,y,z)$ | $(0,1,3,0.2679)$ | $(0.2615,0.7166,2.9132,0.0536)$ | $(0.2615,0.9395,2.8491,0.1219)$ |
| | $(0,1,3,0.2679)$ | $(0.2615,0.9775,2.8363,0.1367)$ | $(0.1356,1,2.9578,0.2133)$ |
| | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0.0882,1,2.9719,0.2322)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0.2035,1,2.9389,0.1866)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0.1574,1,2.9516,0.2046)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0.2031,1,2.9391,0.1868)$ | $(0.0129,1,2.9957,0.2627)$ |
| | $(0,1,3,0.2679)$ | $(0.2615,0.8008,2.8912,0.0759)$ | $(0.2615,1,2.8213,0.1542)$ |
| | $(0,1,3,0.2679)$ | $(0.2615,0.6438,2.9301,0.0376)$ | $(0.2615,0.8695,2.8712,0.0972)$ |

Subsequently, we present Table 3, which represents optimal installation specification data with respect to the monthly average of the south altitude of each city with the azimuth angle $17\pi/18$.

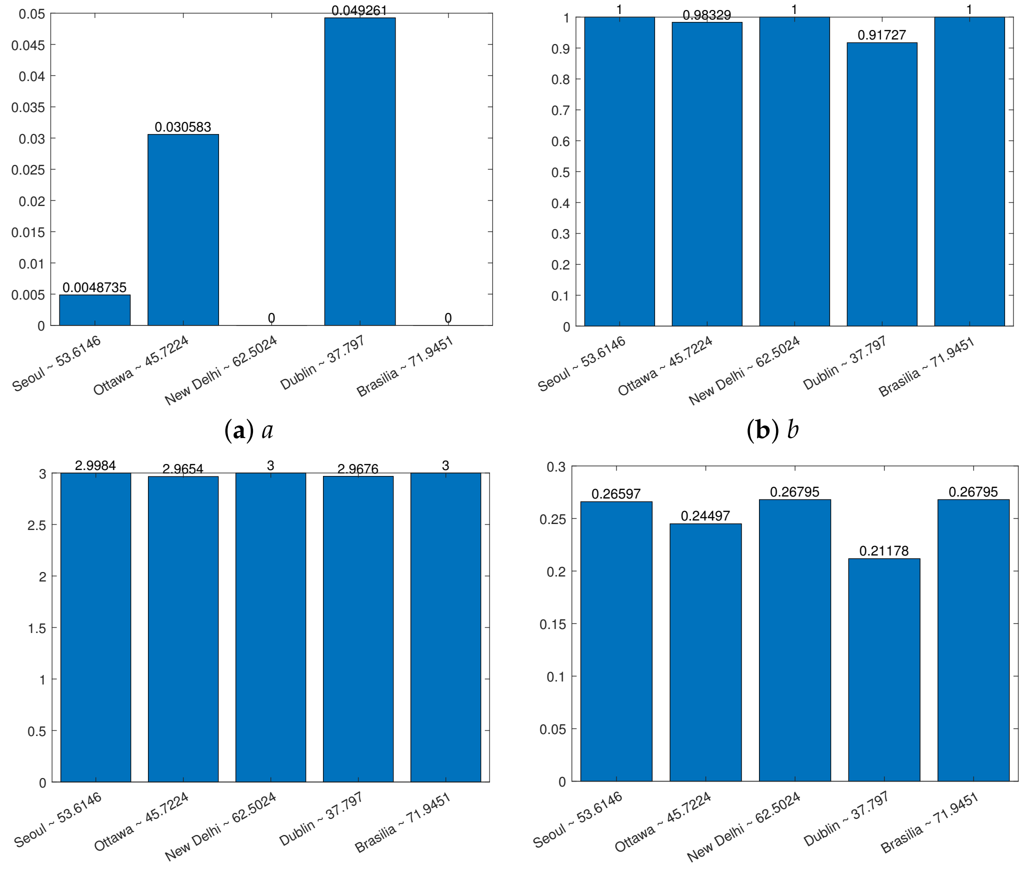
**Table 3.** Installation specification with respect to the south meridian altitude of each city. The azimuth angle is $17\pi/18$. Note that the data is aligned by month from the top (January) to the bottom (December).

| City | Rome | Moscow | Riyadh |
|---|---|---|---|
| $(a,b,y,z)$ | $(0.2391,1,2.9296,0.1730)$ | $(0.5209,0.9653,2.8404,0.0530)$ | $(0,1,3,0.2679)$ |
| | $(0.0021,1,2.9993,0.2671)$ | $(0.3910,1,2.8944,0.1180)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0.0199,1,2.9934,0.2598)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ | $(0,1,3,0.2679)$ |
| | $(0,1,3,0.2679)$ | $(0.2798,1,2.9195,0.1577)$ | $(0,1,3,0.2679)$ |
| | $(0.1694,1,2.9482,0.1999)$ | $(0.5209,1,2.8113,0.0752)$ | $(0,1,3,0.2679)$ |
| | $(0.2969,1,2.9154,0.1514)$ | $(0.5209,0.8942,2.8636,0.0371)$ | $(0,1,3,0.2679)$ |

According to the above tables, it can be verified that the rate of change of the parameter $(a,b,y,z)$ is nearly zero when the south meridian altitude is high, and the latitude is low. These results are independent of the azimuth angle of buildings. Because time series data can be generated, it would be better to rotate the panel in real-time; however, in the case of installing a solar panel in a residential area, it is expensive to construct the panel support that moves according to the time. Therefore, we employ a weighted mean of monthly $(a,b,y,z)$ based on daylight time. More precisely, we adopt the following weight formula to each monthly weight $w_i$ as

$$w_i = \frac{l_i}{\sum_{i=1}^{12} l_i}, \; i = 1, \ldots, 12, \tag{12}$$

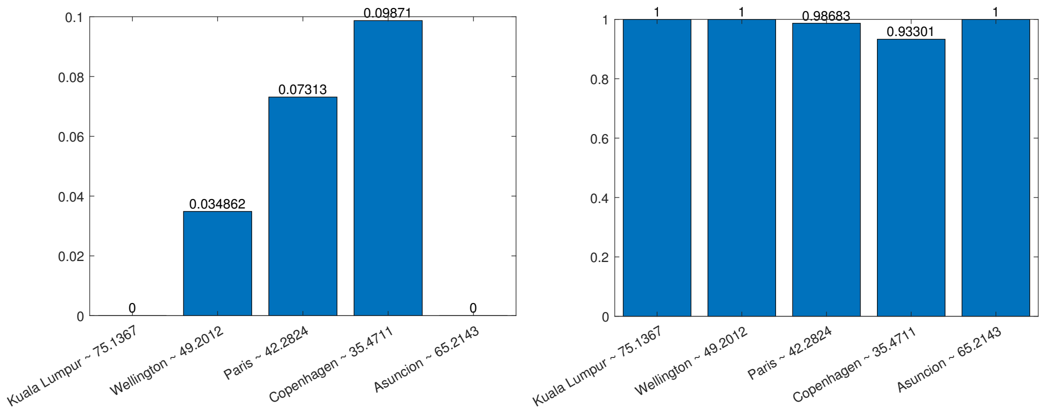
where $l_i$ represents the length of the trajectory of the sun for the $i$-th month. Figures 6–8 represent the weighted average values of the south meridian altitude in selected regions and those of parameters $(a, b, y, z)$.




(**a**) $a$ (**b**) $b$

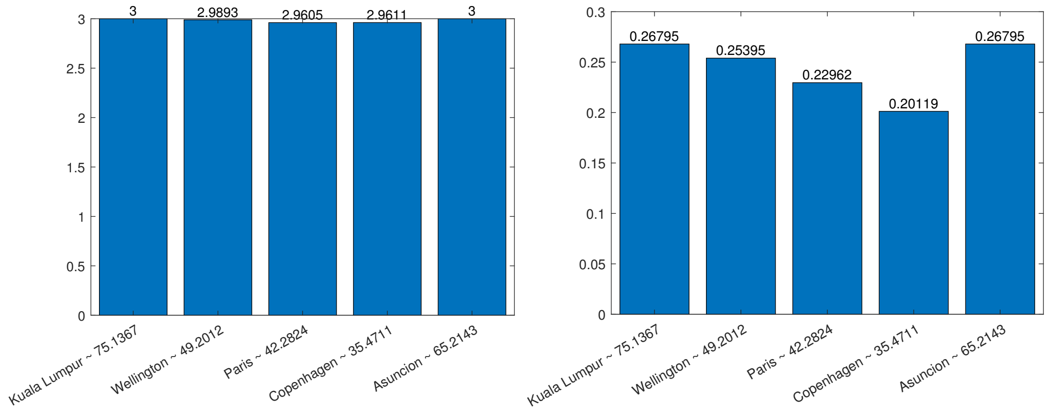
(**c**) $y$ (**d**) $z$

**Figure 6.** Weighted average values of parameters $(a, b, y, z)$ with respect to each region in Table 1. Note that the number displayed after the city is the weighted average value of the south meridian altitude.



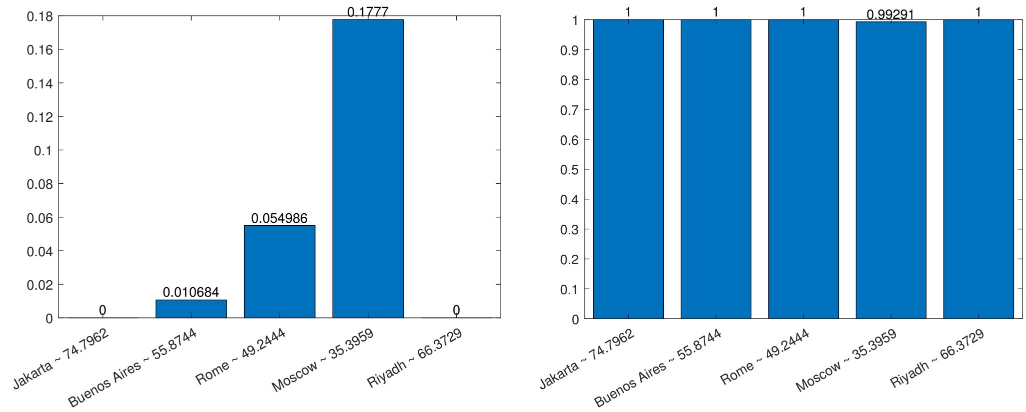

(**a**) $a$ (**b**) $b$

**Figure 7.** *Cont.*

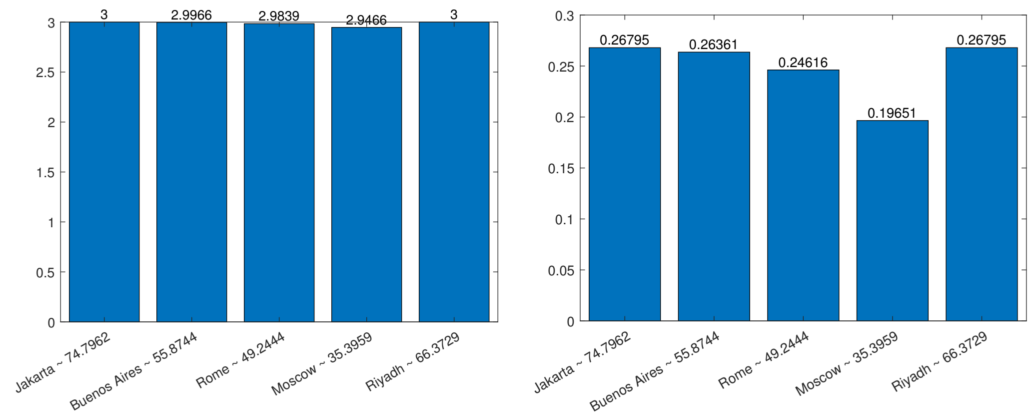



(**c**) $y$ (**d**) $z$

**Figure 7.** Weighted average values of parameters $(a, b, y, z)$ with respect to each region in Table 2. Note that the number displayed after the city is the weighted average value of the south meridian altitude.




(**a**) $a$ (**b**) $b$




(**c**) $y$ (**d**) $z$

**Figure 8.** Weighted average values of parameters $(a, b, y, z)$ with respect to each region in Table 3. Note that the number displayed after the city is the weighted average value of the south meridian altitude.

In addition, we evaluate the energy efficiency compared to the panel installed in parallel to its support by using the following function,

$$E(\mathbf{n}; \mathbf{m}, \mathbf{n}_0) = \sum_{i=1}^{12} D_i \mathbf{m} \cdot (\mathbf{n}_i - \mathbf{n}_0) \times 100(\%), \tag{13}$$

where $D_i$ is given constant and $\mathbf{n}_0$ represents a normal vector of the panel installed in parallel to its support. Table 4 shows the annual efficiency based on (13) for each city.

**Table 4.** The annual energy efficiency based on (13) for each city. The azimuth angle is (a) $73\pi/72$, (b) $35\pi/36$, (c) $17\pi/18$.

| (a) | Seoul | Ottawa | New Delhi | Dublin | Brasilia |
|---|---|---|---|---|---|
| Efficiency (%) | 12.6241 | 9.7297 | 15.7840 | 7.1309 | 18.9755 |
| (b) | Kuala Lumpur | Wellington | Paris | Copenhagen | Asuncion |
| Efficiency (%) | 20.1175 | 11.0114 | 8.5754 | 6.5555 | 16.558 |
| (c) | Jakarta | Buenos Aires | Rome | Moscow | Riyadh |
| Efficiency (%) | 19.0135 | 13.1064 | 10.9335 | 6.8133 | 16.2883 |

From the above results, it is effective to set the installation angle of the solar panel using the proposed method. In particular, the reason why the efficiency is higher in the area near the equator is that the south meridian altitude is much higher than in other areas.

## 4. Conclusions

In this paper, we propose a simple mathematical method to obtain the optimal installation conditions for solar panels in multi-apartment buildings. The presented method provides optimal installation information by considering the azimuth of the building and the amount of sunlight (per month). Through the numerical simulation results, we can confirm that more efficient energy generation can be expected via our method than the panel simply fixed on support without rotation (parallel to the building). Note that the efficiency is maximized if the panel can be rotated in real-time; however, it is not practical for residence; hence, finding an optimal fixed position for the panel is essential to have maximal efficiency under several constraints. In this study, the shadow effect (caused by itself or any other buildings) and the diffuse and reflected irradiations are not considered in the proposed method. These effects are common in urban areas; therefore, we plan to address these aspects in future works.

**Author Contributions:** Conceptualization, Y.C. and J.K.; Methodology, S.Y. and J.K.; Software, S.Y.; Validation, S.Y., J.P., C.L., S.K. (Sangkwon Kim), Y.C., S.K. (Soobin Kwak), H.K. and J.K.; Formal analysis, S.Y. and J.K.; Investigation, S.Y., Y.C., S.K. (Soobin Kwak); Data curation, S.Y.; Writing—original draft, S.Y., J.P., C.L., S.K. (Sangkwon Kim), S.K. (Soobin Kwak), H.K. and J.K.; Writing—review & editing, S.Y., S.K. (Soobin Kwak) and J.K.; Visualization, S.Y., J.P. and S.K. (Soobin Kwak); Supervision, J.K.; Project administration, S.K. (Soobin Kwak), H.K. and J.K.; Funding acquisition, S.Y. and J.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** The first author (S. Yoon) was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT of Korea (MSIT) (No. 2019R1A6A1A11051177). The corresponding author (J. Kim) and the APC were supported by the Brain Korea 21 (BK 21) FOUR from the Ministry of Education.

**Data Availability Statement:** The data used to support the findings of this study are available from the corresponding author upon request.

**Acknowledgments:** The authors wish to thank the reviewers for their constructive and helpful comments on the revision of this article.

**Conflicts of Interest:** The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Appendix A

The following MATLAB codes are available from the corresponding author's webpage: https://mathematicians.korea.ac.kr/cfdkim/open-source-codes/ (accessed on 1 November 2022).

```
clear; close all;

flag=0; % If you have data file then flag=1

if flag==0
    TAB=table({'South Korea'},{'Seoul'},37.5326);
else
    TAB=readtable('./data1.xlsx'); % Data file
end


%% Pre-computing


lat=table2array(TAB(:,3))/180*pi;
dd=linspace(0,364.5,730);
% declination angle formula
del=asin(sin(-23.44/180*pi)*cos(2/365.24*(dd+10)*pi+2*0.0167*sin(360/365.24*(dd-2)/180*pi)));
for i=1:length(lat)
    alp(i,:)=asin(sin(lat(i))*sin(del)+cos(lat(i))*cos(del))/pi*180; % elevation angle formula
end

eqx=asin(cos(lat))/pi*180; % equinox meridian altitude
dat=alp(:,2:2:end);
Date=datetime(2022,1,1):datetime(2022,12,31); Date=Date';
Year=Date.Year; Month=Date.Month; Day=Date.Day;
for i=1:length(lat)
    showdat=dat(i,:)';
    Table=table(Date,Year,Month,Day,showdat);
    % month-avg elevation angle
    Info{i}=varfun(@mean,Table,'GroupingVariables',{'Year' 'Month'},'InputVariables','showdat');
end


%% Procedure 1 & 2


N=100;
x=linspace(-2,2,N);
y=x; z=x;
r=1;
[xx,yy,zz]=ndgrid(x,y,z);
d=sqrt(xx.^2+yy.^2+zz.^2)-r;

ct=0;

for ii=1:12 % month
    for jj=1:length(lat) % # of cities
        ct=ct+1;
        clf;
        th=table2array(Info{jj}(ii,4)); % meridian altitude of month
        t=linspace(-1,1,100);
        cpt=[cos((180-eqx(jj))*pi/180) 0 sin((180-eqx(jj))*pi/180)];
        nv=cross([0 1 0],cpt,2);
        upt=[cos((180-th)*pi/180) 0 sin((180-th)*pi/180)];

        % Use the curve of intersection
        clear T k Y C X Z;
        Y=linspace(-1,1,1000);
        k=(nv(1)*nv(2)*Y-nv(1)^2*upt(1)-nv(1)*nv(3)*upt(3))/(nv(1)^2+nv(3)^2);
        C=(nv(3)^2-nv(3)^2*Y.^2-nv(1)^2*upt(1)^2-nv(2)^2*Y.^2-nv(3)^2*upt(3)^2+...
        2*nv(1)*nv(2)*upt(1)*Y+2*nv(2)*nv(3)*upt(3)*Y-2*nv(1)*nv(3)*upt(1)*upt(3))/(nv(1)^2+nv(3)^2)+k.^2;

        idx=C<0;
        aa=find(idx==0,1,'first');
        bb=find(idx==0,1,'last');
        for i=1:aa-1
            C(aa-i)=C(aa+i);
            Y(aa-i)=Y(aa+i);
        end
        for i=1:length(C)-bb
            C(bb+i)=C(bb-i);
            Y(bb+i)=Y(bb-i);
        end

        C=[C(1:aa-1) 0 C(aa:bb) 0 C(bb+1:end)]; k=[k k(end) k(end)];
        YA=(sqrt(4*nv(3)^2*((nv(3)^2-nv(1)^2*upt(1)^2-nv(3)^2*upt(3)^2-...
        2*nv(1)*nv(3)*upt(1)*upt(3))/(nv(1)^2+nv(3)^2)+k(1)^2)))/(-2*nv(3)^2);
        Y=[Y(1:aa-1) YA Y(aa:bb) -YA Y(bb+1:end)];
        X1=-k+sqrt(C); X2=-k-sqrt(C); X3=-X2; X4=-X1;
        Z1=sqrt(r^2-X1.^2-Y.^2); Z2=sqrt(r^2-X2.^2-Y.^2);
```

```
        Z3=-sqrt(r^2-X1.^2-Y.^2); Z4=-sqrt(r^2-X2.^2-Y.^2);

        idx1=find(Y==min(Y)); idx2=find(Y==max(Y)); idxs=find(Z2==min(Z2));



        if (180-th) < (180-eqx(jj)) % upper than equinox
            ZL=-Z3(1:aa-1); ZR=-Z3(bb+3:end);
            zidl=find(ZL==min(ZL)); zidr=find(ZR==min(ZR)); % find redundant indices
            ZL=ZL(zidl:end); ZR=ZR(1:zidr);
            XL=X1(1:aa-1); XR=X1(bb+3:end);
            XL=XL(zidl:end); XR=XR(1:zidr);
            YL=Y(1:aa-1); YR=Y(bb+3:end);
            YL=YL(zidl:end); YR=YR(1:zidr);
            X2=X2(idx1:idx2);
            X2=[XL X2 XR];
            Y=Y(idx1:idx2);
            Y=[YL Y YR];
            Z2=Z2(idx1:idx2);
            Z2=[ZL Z2 ZR];
            Z3=Z3(idx1:idx2);

        elseif (180-th)==(180-eqx(jj)) % same
            X2=X2(idx1:idx2);
            Y=Y(idx1:idx2);
            Z2=Z2(idx1:idx2);

        else % lower than equinox
            if length(idxs)==2
                idx3=idxs(1); idx4=idxs(2);
                Z2(idx3+1)=0; Z2(idx4-1)=0;
                Y(idx3+1)=-sqrt(-4*(upt(1)^2+nv(3)^2*upt(3)^2/nv(1)^2+2*upt(1)*upt(3)*nv(3)/nv(1)-1))/2;
                Y(idx4-1)=sqrt(-4*(upt(1)^2+nv(3)^2*upt(3)^2/nv(1)^2+2*upt(1)*upt(3)*nv(3)/nv(1)-1))/2;
                X2(idx3+1)=-sqrt(1-Y(idx3+1)^2);
                X2(idx4-1)=-sqrt(1-Y(idx4-1)^2);
                idx3=idx3+1; idx4=idx4-1;
            else
                idx3=idxs(find(idxs>idx1,1,'first')); idx4=idxs(find(idxs<idx2,1,'last'));
            end
            X2=X2(idx3:idx4);
            Y=Y(idx3:idx4);
            Z2=Z2(idx3:idx4);
        end


        % Azimuth angle formula - standard south(pi) // 0: south / rthe +: west / rthe -: east
        rthe=0; % range = (-pi/2,pi/2)
        azthe=pi+rthe;
        if azthe==pi
            id=find(X2>0); id2=find(X2≤0);
        elseif rthe>0
            id=find(Y<abs(tan(0.5*pi-rthe))*X2); id2=find(Y>=abs(tan(0.5*pi-rthe))*X2);
        else
            id=find(Y>-abs(tan(0.5*pi+rthe))*X2); id2=find(Y≤-abs(tan(0.5*pi+rthe))*X2);
        end

        P=0.5*[X2(id2(end))+X2(id2(1)) Y(id2(end))+Y(id2(1)) Z2(id2(end))+Z2(id2(1))];
        M=[X2(id2(1)) Y(id2(1)) Z2(id2(1))];
        P1=P-M;
        S=cross(nv,P1,2);
        tt=sqrt((1-norm(P,2)^2+dot(P,S)^2/norm(S,2)^2)/norm(S,2)^2)-dot(P,S)/norm(S,2)^2;
        solar=P+tt*S;


%% Procedure 3 & 4 & 5


        the=pi-rthe;
        Rot=[cos(the) -sin(the) 0; sin(the) cos(the) 0; 0 0 1];
        Ly=3; Lz=2; b=Lz/2; a=0;
        y1=(2*Ly^2-2*a*b)/(2*sqrt(Ly^2-a^2));
        z1=Lz-sqrt(Ly^2+Lz^2-b^2-y1^2);
        pl1=[b-a y1-sqrt(Ly^2-a^2) z1
            b y1 z1
            a sqrt(Ly^2-a^2) Lz
            0 0 Lz
            b-a y1-sqrt(Ly^2-a^2) z1];
        u1 = pl1(1,:) - pl1(4,:); v1 = pl1(3,:) - pl1(4,:);
        n1 = cross(u1,v1); n1 = n1/norm(n1);
        A=Rot*n1';
```

```
a=b; y=(2*Ly^2-2*a*b)/(2*sqrt(Ly^2-a^2));
z=Lz-sqrt(Ly^2+Lz^2-b^2-y^2);
pl=[b-a y-sqrt(Ly^2-a^2) z
    b y z
    a sqrt(Ly^2-a^2) Lz
    0 0 Lz
    b-a y-sqrt(Ly^2-a^2) z];

pr=[0 Ly Lz
    a Ly-sqrt(Ly^2-a^2) Lz
    b Ly-y z
    b-a Ly-y+sqrt(Ly^2-a^2) z
    0 Ly Lz];

u = pl(1,:) - pl(4,:); v = pl(3,:) - pl(4,:);
n = cross(u,v); n = n/norm(n);
u2 = pr(2,:) - pr(1,:); v2 = pr(4,:) - pr(1,:);
n2 = cross(u2,v2); n2 = n2/norm(n2);
B=Rot*n'; C=Rot*n2';
aaa=norm(A-B,2); bbb=norm(B-C,2); ccc=norm(C-A,2);
sss=.5*(aaa+bbb+ccc);
SSS=sqrt(sss*(sss-aaa)*(sss-bbb)*(sss-ccc));
tempn=cross(B-A,C-A); tempn=tempn/norm(tempn,2);
D=-dot(tempn,A);
X=-D/(tempn(1)+tempn(2)*solar(2)...
/solar(1)+tempn(3)*solar(3)/solar(1));
Y=solar(2)*X/solar(1); Z=solar(3)*X/solar(1);
kk=[X Y Z]; kk=kk';
sss1=.5*(norm(A-kk,2)+norm(B-kk,2)+aaa);
sss2=.5*(norm(B-kk,2)+norm(C-kk,2)+bbb);
sss3=.5*(norm(C-kk,2)+norm(A-kk,2)+ccc);
SSS2=sqrt(sss1*(sss1-norm(A-kk,2))...
*(sss1-norm(B-kk,2))*(sss1-aaa))+ ...
    sqrt(sss2*(sss2-norm(B-kk,2))...
    *(sss2-norm(C-kk,2))*(sss2-bbb))+ ...
    sqrt(sss3*(sss3-norm(C-kk,2))...
    *(sss3-norm(A-kk,2))*(sss3-ccc));

distan=[norm(A'-solar,2) norm(B'-solar,2) norm(C'-solar,2)];
maxd=max(distan); indd=find(distan==maxd);
Rot2=[cos(rthe) -sin(rthe) 0; sin(rthe) cos(rthe) 0; 0 0 1];
if length(indd) < 2
    if indd==1
        pnv=cross(C,B);
    elseif indd==2
        pnv=cross(A,C);
    else
        pnv=cross(B,A);
    end
    pnv=pnv';
    if abs(SSS-SSS2) < 1e-7
        rn=solar;
        temprn=Rot2*rn';
        rz=Lz-Lz*sqrt(temprn(1)^2+temprn(2)^2);
        a=abs(temprn(2)*Ly*Lz/(Lz-rz));
        KK=temprn(3)*Lz*a/Ly;
        CC=-((Ly^2-a^2)*((Lz-z)^2-Ly^2-Lz^2)...
        +temprn(3)^2*Ly^2*Lz^2)/Ly^2;
        ry=sqrt(CC+KK^2)-KK;
        b=abs((temprn(3)*Ly*Lz+a*ry)...
        /sqrt(Ly^2-a^2));
    else
        rn=solar-dot(solar,pnv)/norm(pnv,2)^2*pnv;
        rn=rn/norm(rn,2);
        if rn(3)>A(3)
            rn=A;
            b=Lz/2;
            a=0;
            ry=(Ly^2-a*b)/sqrt(Ly^2-a^2);
            rz=Lz-sqrt(Ly^2+Lz^2-b^2-ry^2);
        elseif rn(3) < 0
            dist1=norm(solar-B,2);
            dist2=norm(solar-C,2);
            rn=(dist1>dist2)*C+(dist1<dist2)*B;
            b=Lz/2;
            a=b;
            ry=(Ly^2-a*b)/sqrt(Ly^2-a^2);
            rz=Lz-sqrt(Ly^2+Lz^2-b^2-ry^2);
        else
            temprn=Rot2*rn';
            b=Lz/2;
```

```
                    rz=Lz-Lz*sqrt(temprn(1)^2+temprn(2)^2);
                    a=abs(temprn(2)*Ly*Lz/(Lz-rz));
                    ry=(Ly^2-a*b)/sqrt(Ly^2-a^2);
                end
            end
        else
            if solar(3) > 0
                rn=A;
                b=Lz/2;
                a=0;
                ry=(Ly^2-a*b)/sqrt(Ly^2-a^2);
                rz=Lz-sqrt(Ly^2+Lz^2-b^2-ry^2);
            else
                rn=.5*(B+C);
                b=0;
                a=0;
                ry=(Ly^2-a*b)/sqrt(Ly^2-a^2);
                rz=Lz-sqrt(Ly^2+Lz^2-b^2-ry^2);
            end
        end
        val{jj,ii}=[a b ry rz];
    end
end
```

## References


1. Acosta-Silva, Y.d.J.; Torres-Pacheco, I.; Matsumoto, Y.; Toledano-Ayala, M.; Soto-Zarazúa, G.M.; Zelaya-Ángel, O.; Méndez-López, A. Applications of solar and wind renewable energy in agriculture: A review. *Sci. Prog.* **2019**, *102*, 127–140. [CrossRef]
2. Firozjaei, M.K.; Nematollahi, O.; Mijani, N.; Shorabeh, S.N.; Firozjaei, H.K.; Toomanian, A. An integrated GIS-based Ordered Weighted Averaging analysis for solar energy evaluation in Iran: Current conditions and future planning. *Renew. Energy* **2019**, *136*, 1130–1146. [CrossRef]
3. Sintali, I.S.; Abioye, A.M. Effects of Tilt Angles on the Thermal Performance of a Solar Parabolic Trough Collector System. *Am. J. Appl. Sci. Res.* **2020**, *6*, 76–82. [CrossRef]
4. Skerlić, J.; Radulović, J.; Nikolić, D.; Bojić, M. Maximizing performances of variable tilt flat-plate solar collectors for Belgrade (Serbia). *J. Renew. Sustain. Energy* **2013**, *5*, 041820. [CrossRef]
5. Le Roux, W.G. Optimum tilt and azimuth angles for fixed solar collectors in South Africa using measured data. *Renew. Energy* **2016**, *96*, 603–612. [CrossRef]
6. Handoyo, E.A.; Ichsani, D. The optimal tilt angle of a solar collector. *Energy Procedia* **2013**, *32*, 166–175. [CrossRef]
7. Darhmaoui, H.; Lahjouji, D. Latitude based model for tilt angle optimization for solar collectors in the Mediterranean region. *Energy Procedia* **2013**, *42*, 426–435. [CrossRef]
8. Chang, Y.P. Optimal the tilt angles for photovoltaic modules in Taiwan. *Int. J. Electr. Power Energy Syst.* **2010**, *32*, 956–964. [CrossRef]
9. Despotovic, M.; Nedic, V. Comparison of optimum tilt angles of solar collectors determined at yearly, seasonal and monthly levels. *Energy Convers. Manag.* **2015**, *97*, 121–131. [CrossRef]
10. Saraf, G.; Hamad, F.A.W. Optimum tilt angle for a flat plate solar collector. *Energy Convers. Manag.* **1988**, *28*, 185–191. [CrossRef]
11. Skeiker, K. Optimum tilt angle and orientation for solar collectors in Syria. *Energy Convers. Manag.* **2009**, *50*, 2439–2448. [CrossRef]
12. Benghanem, M. Optimization of tilt angle for solar panel: Case study for Madinah, Saudi Arabia. *Appl. Energy* **2011**, *88*, 1427–1433. [CrossRef]
13. Lave, M.; Kleissl, J. Optimum fixed orientations and benefits of tracking for capturing solar radiation in the continental United States. *Renew. Energy* **2011**, *36*, 1145–1152. [CrossRef]
14. Bakirci, K. General models for optimum tilt angles of solar panels: Turkey case study. *Renew. Sustain. Energy Rev.* **2012**, *16*, 6149–6159. [CrossRef]
15. Hartner, M.; Ortner, A.; Hiesl, A.; Haas, R. East to west–The optimal tilt angle and orientation of photovoltaic panels from an electricity system perspective. *Appl. Energy* **2015**, *160*, 94–107. [CrossRef]
16. Chinchilla, M.; Santos-Martín, D.; Carpintero-Rentería, M.; Lemon, S. Worldwide annual optimum tilt angle model for solar collectors and photovoltaic systems in the absence of site meteorological data. *Appl. Energy* **2021**, *281*, 116056. [CrossRef]
17. Kim, G.Y.; Han, D.S.; Lee, Z. Solar panel tilt angle optimization using machine learning model: A case study of Daegu City, South Korea. *Energies* **2020**, *13*, 529. [CrossRef]
18. Gwesha, A.O.; Li, P.; Alfulayyih, Y.M. Optimization of Fixed Photovoltaic Panel "Tilt" Angles for Maximal Energy Harvest Considering Year-Around Sky Coverage Conditions. *J. Sol. Energy Eng.* **2021**, *143*, 021002. [CrossRef]
19. Gökmen, N.; Hu, W.; Hou, P.; Chen, Z.; Sera, D.; Spataru, S. Investigation of wind speed cooling effect on PV panels in windy locations. *Renew. Energy* **2016**, *90*, 283–290. [CrossRef]
20. Armstrong, S.; Hurley, W. A new methodology to optimise solar energy extraction under cloudy conditions. *Renew. Energy* **2010**, *35*, 780–787. [CrossRef]

21. Elsayed, M.M. Optimum orientation of absorber plates. *Sol. Energy* **1989**, *42*, 89–102. [CrossRef]
22. Zhong, Q.; Tong, D. Spatial layout optimization for solar photovoltaic (PV) panel installation. *Renew. Energy* **2020**, *150*, 1–11. [CrossRef]
23. Oh, M.; Kim, J.Y.; Kim, B.; Yun, C.Y.; Kim, C.K.; Kang, Y.H.; Kim, H.G. Tolerance angle concept and formula for practical optimal orientation of photovoltaic panels. *Renew. Energy* **2021**, *167*, 384–394. [CrossRef]
24. Woolf, H.M. *On the Computation of Solar Elevation Angles and the Determination of Sunrise and Sunset Times*; National Aeronautics and Space Administration: IL, USA, 1968.
25. Available online: https://latlong.net (accessed on 6 January 2023).