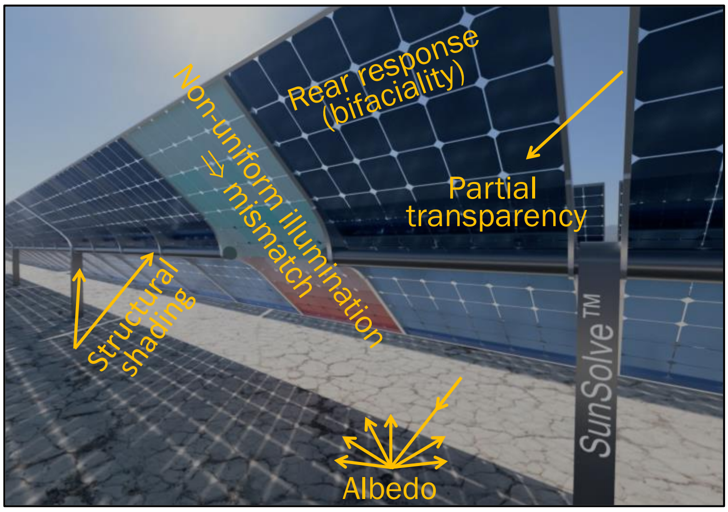
Step-by-step guide to determine PVSyst bifacial inputs with SunSolve™

Version 6.0 (Apr-2024)



PVSyst uses these factors to account for optical complications relating to bifacial systems:

- the transmission factor accounts for light passing between or through modules;
- the albedo factor is the fraction of light reflected by the ground;
- the shading factor accounts for the change in rear illumination due to shading and reflection from structural supports—like posts and torque tubes;
- the bifaciality factor states how the module's efficiency under rear illumination compares to its efficiency under front illumination; and
- the rear mismatch factor account for the reduction in module power due to the rear illumination being non-uniform.

What values should be used for these input factors? That's difficult to know because some of the factors cannot be measured, and because they depend on system configuration, weather, and the location of the system.

This guide describes a method to determine the input factors using the simulation software called SunSolve-Yield: www.pvlighthouse.com.au/sunsolve.

Contents

1. Desired PVSyst inputs

The figure below shows the inputs required to simulate a bifacial module with PVSyst.[1] In this info sheet, we describe a methodology to determine the factors using SunSolve-Yield.

Factor	Symbol	Represents
Transmission	f_T	Light passing between or through modules.
Albedo	f_A	Ground reflectance.
Structural shading	f_S	Shading from torque-tube, posts, clamps, etc.
Rear mismatch	f_{MR}	Reduction in the extra power arising from rear illumination due to cell-to-cell mismatch.
Bifaciality	f_B	Module response to rear illum relative to front illum.

Bifacial system definition tab in PVSyst

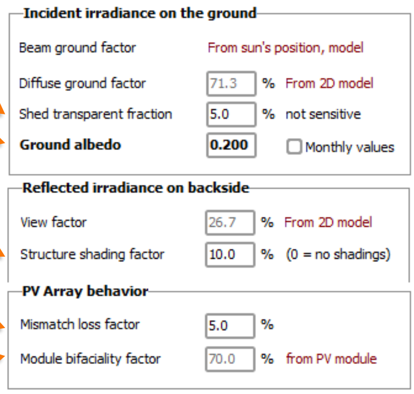


Figure 1: Screenshot of PVSyst (7.4.6) showing bifacial inputs and an explanation of each.

[1] We used version 7.4.6 of PVSyst when compiling this white paper. It's possible that other versions of PVSyst will use different inputs, input names, equations, and user interfaces.

2. Improvements to our previous methodology

The approach presented here supersedes the approach we've documented previously.[2] There is one new option, which arose when PVSyst released PVSyst Version 7.4.6, and one correction.

<u>New option due to the change in PVSyst</u>

Prior to V7.4.6, PVSyst calculated the rear mismatch loss P_{MR} with the equation

$$P_{MR} = f_{MR} \cdot \eta_F \cdot \Phi_R, \tag{1}$$

where f_{MR} is the rear mismatch factor, η_F is the front-side module efficiency at STC, and Φ_R is the IAM-corrected rear irradiance. With Version 7.4.6, this equation was modified to depend on η_R, the rear-side module efficiency at STC:

$$P_{MR} = f_{MR} \cdot \eta_R \cdot \Phi_R, \tag{2}$$

equivalent to

$$P_{MR} = f_{MR} \cdot f_B \cdot \eta_F \cdot \Phi_R, \tag{3}$$

where f_B is the bifaciality of the module, defined as $f_B = \eta_R / \eta_F$.

Our goal is to determine the most appropriate value to insert as f_{MR} in PVSyst. Thus, the method for determining f_{MR} now depends on which version of PVSyst is being used. For versions prior to V7.4.6, we use

$$f_{MR} = (f_M - f_{MF}) \cdot f_B \cdot \left(1 + \frac{I_{F1}}{I_{R1}}\right), \tag{4}$$

where I_{F1} and I_{R1} are the front and rear generation current as determined by SunSolve during Simulation 1 (the simulation where the system is solved without alteration).

---

[2] The previous version (Version 5) was released in Sep-2022. (Version 4) was released in May-2021.

For PVSyst V7.4.6 and beyond, we use

$$f_{MR} = (f_M - f_{MF}) \cdot \left(1 + \frac{I_{F1}}{I_{R1}}\right). \quad (5)$$

The derivation of these equations is given in Section 6.3.

The Excel spreadsheet that combines the simulations has been modified accordingly. It now contains the option to state which equation should be applied.

<u>Correction when using PVSyst prior to Version 7.4.6</u>

In the previous version of this document, which only applied to PVSyst prior to V7.4.6, the equation for f_{MR} was incorrectly written as

$$f_{MR} = (f_M - f_{MF}) \cdot \left(f_B + \frac{I_{F1}}{I_{R1}}\right) \quad (6)$$

instead of as shown in Equation (4). The Excel spreadsheet that combines the simulations also applied (6) instead of (4) and this is now corrected.

For a modern c-Si module and system, I_F/I_R is about 10–20, which is much larger than f_B. Thus, previous applications of our procedure would overestimate f_{MR} by a factor of $(1 – f_B)$. Since a modern c-Si module has f_B of about 0.75 or 0.8, this means that the old procedure would overestimate f_{MR}, and hence P_{MR}, by about 20% or 25%. P_{MR} is usually very low, like < 0.2% of annual yield, and thus this correction is unlikely to have a significant impact on yield forecasts.

3. General procedure

The procedure to determine all of the PVSyst input factors requires six SunSolve simulations, as illustrated in

Figure 2. The outputs from these simulations are then combined to determine f_A, f_T, f_S, f_M and f_{MR} at every hour of the year. Finally, we determine a weighted-average value of each factor to represent a year for the specified weather, location and system design.[3]

In this way, we determine PVSyst inputs that account for many subtle effects that cannot otherwise be accounted for by PVSyst. These include cell-to-cell mismatch due to row-to-row shading of diffuse light; shading and reflection from torque tubes, posts, frames and clamps; spectral and angular effects; and secondary reflections, like the reflection from one module to another.

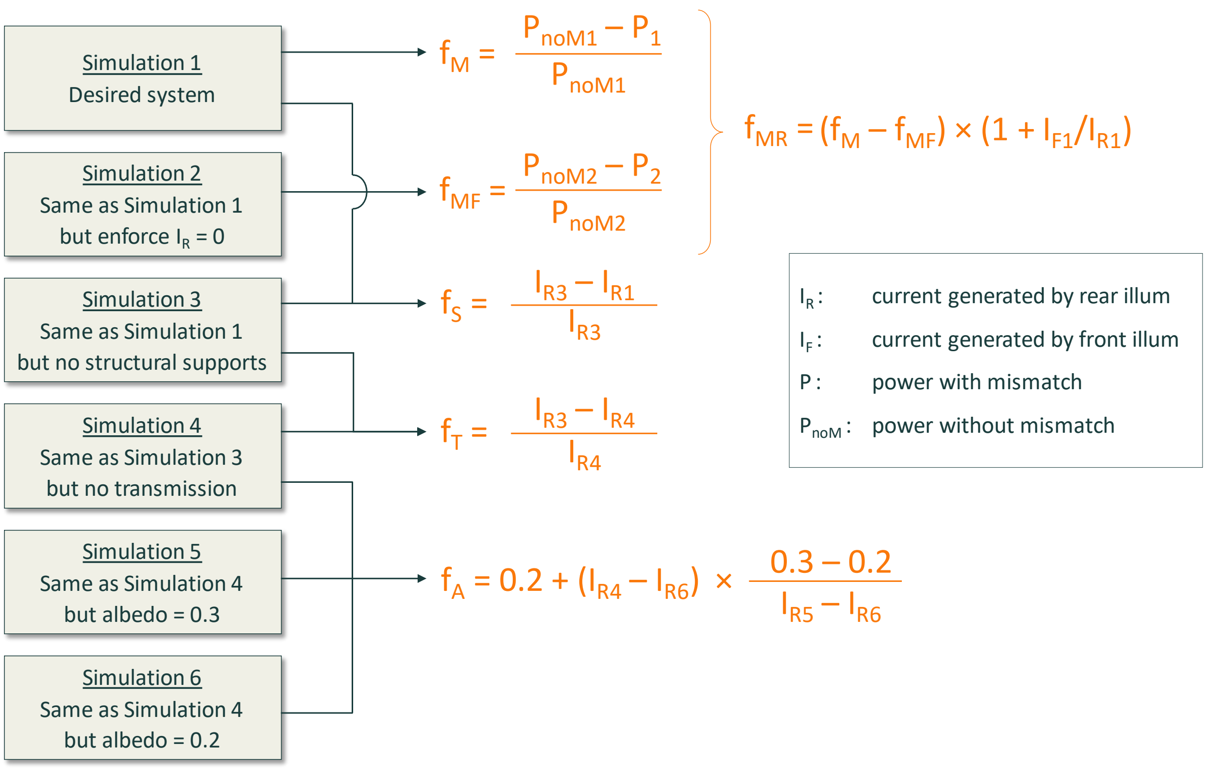


Figure 2: General approach for determining f_A, f_T, f_S, f_M and f_{MR}.[4]

---

[3] In Section 7.1 we describe the cases where some of these steps can be omitted.

[4] The equation for f_{MR} is for PVSyst V6.7.4 and later versions. Use Equation (4) when using prior versions of PVSyst. The bifaciality factor f_B in the PAN file is sufficient.

4. Step-by-step instructions

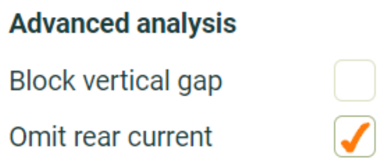
We now describe how to determine PVSyst factors using SunSolve-Yield. The full procedure requires the six simulations summarised in Table I.[5] All of the inputs we mention are found on either the Systems or Options tab of SunSolve-Yield.

Table I: Differences between the five simulations.

Simulation	Omit rear collection	Structural supports	Transmission between modules	Albedo
1	No	As desired for site	As desired for site	As desired for location
2	Yes	As desired for site	As desired for site	As desired for location
3	No	None	As desired for site	As desired for location
4	No	None	None	As desired for location
5	No	None	None	30% at all wavelengths
6	No	None	None	20% at all wavelengths

First, run the six simulations as follows:

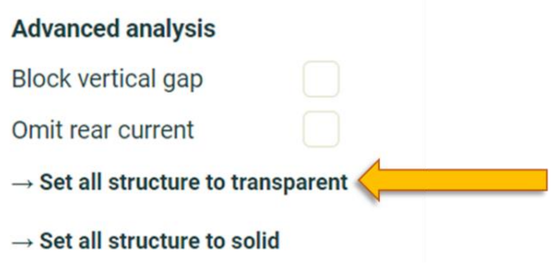
1. Load the inputs for the desired module, system and weather and run Simulation 1.
2. Duplicate Simulation 1, check "omit rear current" on the Options tab and run Simulation 2.



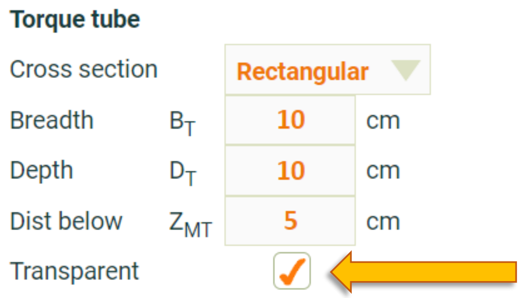
3. Duplicate Simulation 2, uncheck "Omit rear current" on the Options tab, make all mounting structures transparent and run Simulation 3. The structures (including custom objects) are made transparent by clicking the "Set all structure to transparent" button on the Options tab.

---

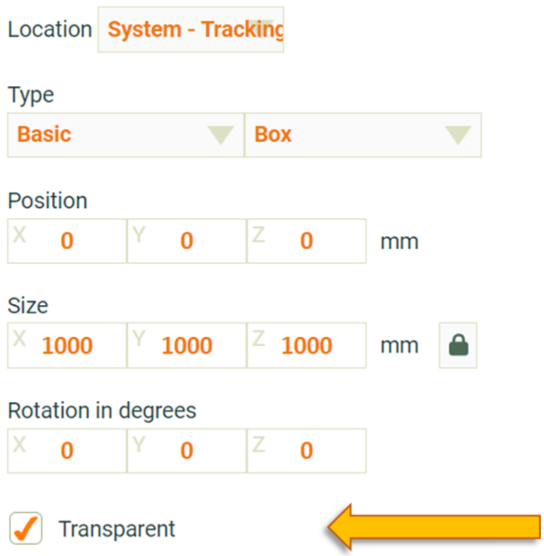
[5] There are many interesting subtleties to the methodology that may be of interest to advanced users. We describe them in Section 7. For example, you can sometimes skip a simulation with no loss in accuracy, or you might improve accuracy by modifying the albedo in Simulations 5 and 6.



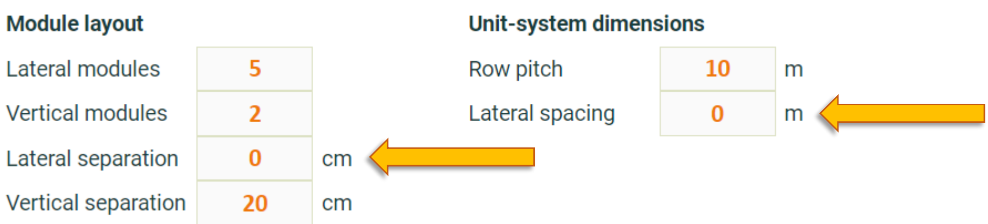
Alternatively, this can be done manually with the checkbox under each mounting structure, as shown below.



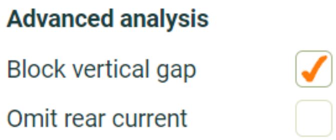
And for custom objects as shown below:



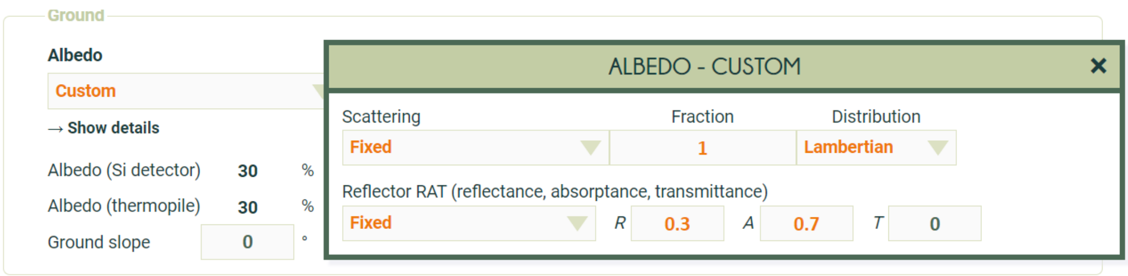
4. Duplicate Simulation 3, remove the lateral spacing, block the vertical gap, and run Simulation 4. The lateral spacing is removed by setting the lateral spacing to zero under the module layout and the unit-system dimensions, as shown below.



Rather than setting the vertical separation to zero, check “Block vertical gap” on the Options tab.



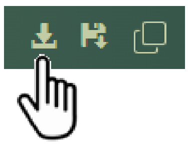
5. Duplicate Simulation 4, set the ground albedo to have a fixed 30% reflectance with 100% Lambertian scattering and run Simulation 5. The albedo can be set as shown in the image below.



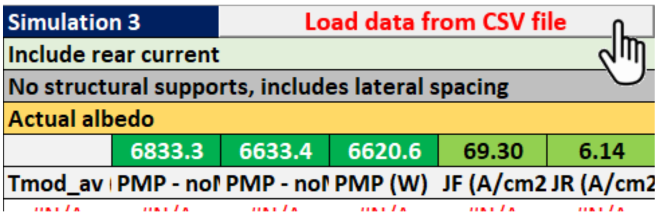
6. Duplicate Simulation 5, set the ground albedo to be a fixed 20% and run Simulation 6.

The simulations are now complete. Next, perform the analysis:

7. For each simulation, download the results in a CSV file, by clicking the download icon,



   or by clicking button “→ Download all results in CSV file” at the bottom of the Summary tab in the Output section. (It’s also usually worth saving the SIM file of the completed simulation, although that’s not required for this procedure.)
8. Open the spreadsheet “PVSystInputs-analysis.xlsx”. You can download the spreadsheet here. The spreadsheet contains macros.
9. Navigate to the Data tab and load the results from each simulation by clicking on the load buttons. This will run a macro to help find and load the data from the relevant CSV file. Loading Simulation 1 will also load the relevant weather data.



10. Set Cell AZ2 to TRUE if solving for PVSyst V6.7.4 or later version. Otherwise set Cell AZ2 to FALSE.

11. Set the module's bifaciality factor f_B in cell AZ5. This affects f_{MR} when Cell AZ2 is FALSE.[6]
12. Observe the calculated PVSyst inputs for f_A, f_S, f_T and f_{MR},
    a. Values for every hour are in columns AW, AX, AY and BC.
    b. Power-weighted averages for the year are in cells AW5, AX5, AY5 and BC5.
    c. Various plots of the PVSyst inputs can be found on the other tabs.

And finally, load the inputs into PVSyst, referring to the screen shots of Figure 1.

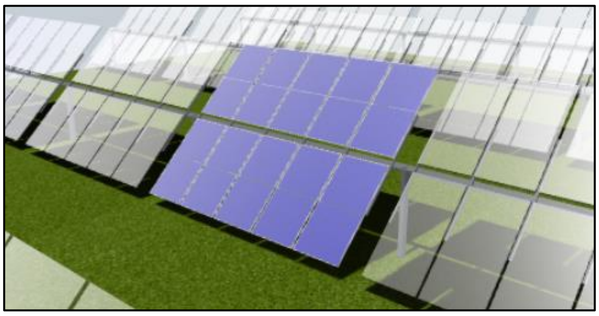
13. On the bifacial system definition tab, set
    d. 'ground albedo' to f_A,
    e. 'shed transparent factor' to f_T,
    f. 'structure shading factor' to f_S
    g. 'bifaciality factor' to f_B, which is the bifaciality factor in the PAN file, and
    h. 'mismatch factor' to f_{MR}.
14. There is one additional input that is useful for PVSyst simulations, and that's the mismatch factor relating to front illumination f_{MF}. We discuss how this can be used in Section 0.

---

6 If the module you are simulating was loaded from a PAN file, you'll find the bifaciality factor f_B listed in that PAN file. Or, if the module was loaded from a PVL or PVM file, you can determine f_B by simulating the module in SunSolve-Power under front and rear illumination; f_B is then the maximum power under rear illumination divided by the maximum power under front illumination. Note, f_B is only required for the calculation of f_{MR}.

5. Examples

We now determine the PVSyst factors for the two example systems shown in Figure 3: (i) a 2P single-axis tracker and (ii) a 4L fixed system tilted at 25°.



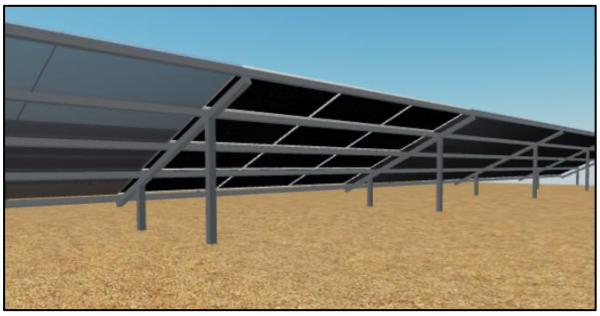


Figure 3: Examples: **(left)** a 2P single-axis tracker and **(right)** a 4L fixed-tilt system.

What PVSyst factors best represent these systems? That depends on many aspects. Here is some relevant information about both of the simulated systems.

- They are located near Chicago, USA.
- Their one-hourly weather data is generated by Meteonorm 7.2.
- They contain bifacial Longi LR5 530-W modules with 144 half-cut cells and f_B = 0.7.
- Their structural supports are made of galvanised-steel and have the wavelength-dependent reflection shown in Figure 4.
- The ground is green grass throughout the year with the albedo shown in Figure 4.

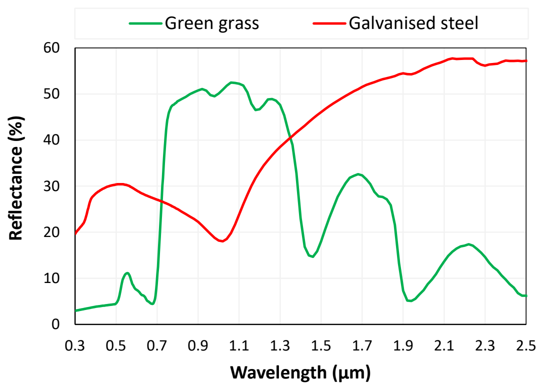


Figure 4: Wavelength dependent reflectance under air for (red) galvanised steel and (green) green rye grass. Data is from the ECOSTRESS Spectral Library.[7]

[7] Meerdink, S. K., Hook, S. J., Roberts, D. A., & Abbott, E. A. (2019). The ECOSTRESS spectral library version 1.0. Remote Sensing of Environment, 230(111196), 1–8. Library available online at https://speclib.jpl.nasa.gov/

So, what albedo factor f_A should be applied to represent this green grass in when its albedo is 5−10% below 700 nm and 45−55% between 700 and 1300 nm? Let's find out:

In Section 4 we described how six simulations could be used to determine all PVSyst factors. We now present some images to help understand those simulations. On the previous page, the images represent Simulation 1, for which the system is set to have all desired inputs. In the images below, we see the setup for Simulations 3 and 4 of the single-axis tracker. For Simulation 3, the structural components are made transparent, and for Simulation 4, the lateral spacing is removed and the light is blocked from passing through any vertical separation between the modules. Simulations 5 and 6, which modify just the albedo, would look the same as Simulation 4 in the graphical representation of the system.

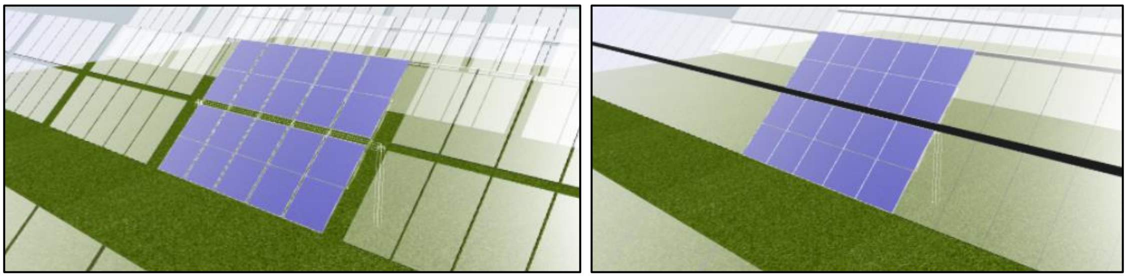


Figure 5: Single-axis system with the inputs set for **(left)** Simulation 3 and **(right)** Simulation 4.

By following the procedure in Section 4 we get the results listed in Table II. These are the weighted-average PVSyst factors that represent an entire year. We find that the value of f_A that best represents the green grass for our site is ~26%. We also find that the total cell-to-cell mismatch loss is small (f_M = 0.3% and 0.5%) and dominated by the front irradiance. In fact, for the single-axis tracker, f_{MR} is effectively zero (0.01%).

Table II: Weighted-average yearly PVSyst outputs for the two examples.

Factor	Example 1 Single-axis tracker	Example 2 Fixed-tilt system
f_B	70%	70%
f_A	25.5%	26.2%
f_T	17.8%	0%
f_S	8.1%	27.4%
f_M	0.3%	0.5%
f_{MR}	0.01%	1.2%

Although these weighted-average values are best-fit for a PVSyst annual yield simulation, they are not ideal at all times of day and year. In Figure 6 we present the weighted-average daily values of the single-axis tracker example and plot them against time of year and "diffuse fraction", which is the fraction of the irradiance composed of diffuse sunlight (equal to DHI/GHI).

Figure 6 compares the daily factors (symbols) to the yearly factor (lines). It's clear that the factors trend with season and cloud.

For example, the albedo factor f_A varies from 24% to 28% and is highest in the winter. Since the incident spectrum tends to be redder in winter (the sun is lower in the sky), and since the grass is more reflective at long wavelengths, this effectively makes the albedo higher.

The transmission factor f_T varies from 8% to 32%. It decreases as the diffuse fraction increases because less diffuse sunlight can pass between the modules to land on the ground.

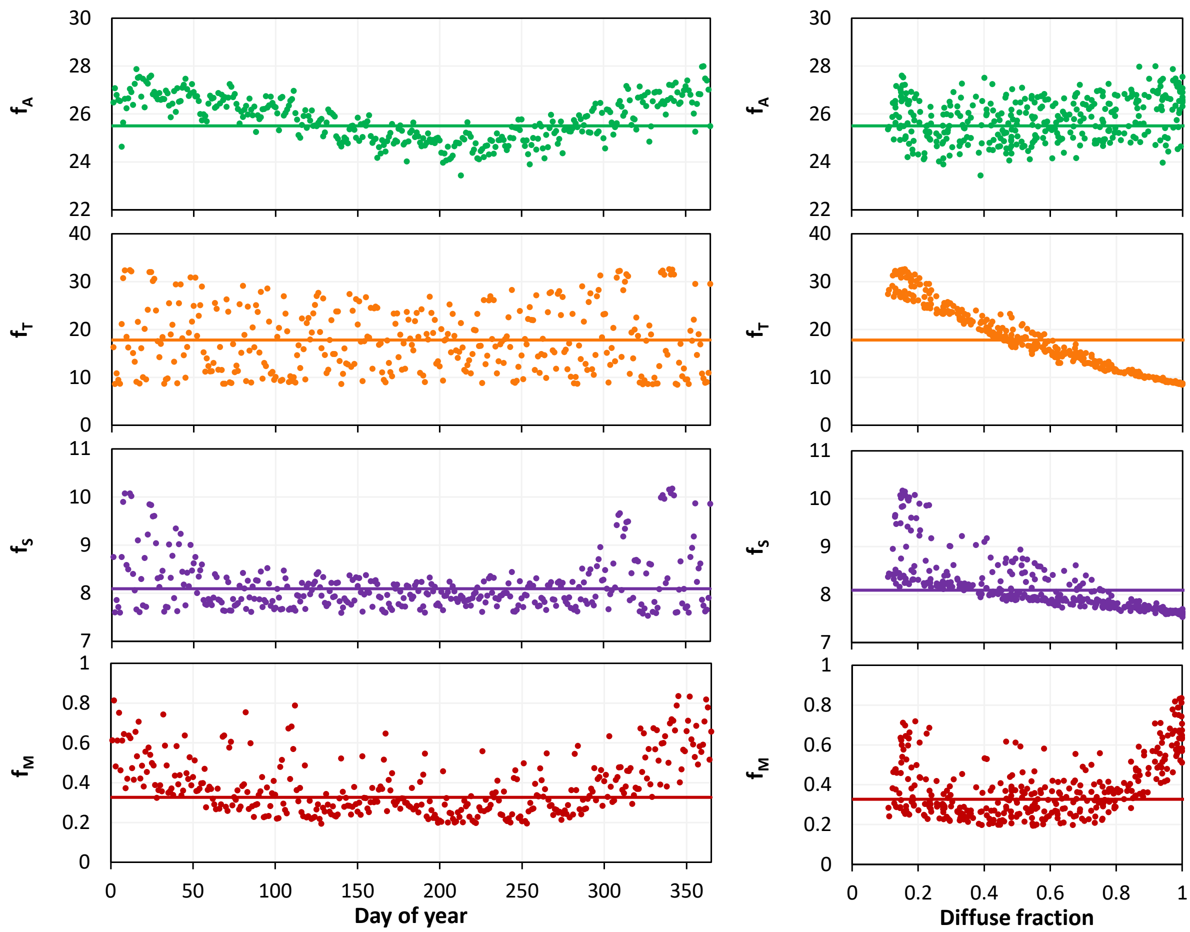


Figure 6: Energy-weighted daily values for PVSyst factors determined for the single-axis tracker example. The factors are plotted against **(left)** day of year and **(right)** diffuse fraction.

The shading factor f_S varies between 7.5% and 10%. It decreases as the diffuse fraction increases because more diffuse light will fall underneath the modules (rather than between the rows) from where there is less shading from the torque tube and clamps. Sunny winter days give the highest f_S.[8]

Finally, the mismatch factor f_M varies between 0.2% and 0.8%, where f_M depends strongly on cloud. While mismatch arises from non-uniform irradiance on both the front and rear, the strongest contribution arises from row-to-row shading of diffuse light, so f_M tends to be much higher on cloudy days. (For a similar 1P tracker, f_M had the same trend but varied from 0.2 to 1.8%.)

When we examine f_{MR}, as shown in Figure 7, we make the curious observation that f_{MR} is sometimes positive and sometimes negative. That is, the rear irradiance is sometimes increasing mismatch loss (when positive) and sometimes reducing mismatch loss (when negative). Under direct light, the front irradiance is uniform,[9] so rear irradiance can only increase non-uniformity and create cell-to-cell mismatch. Under diffuse light, however, the rear irradiance can act to reduce the non-uniformity, or at least, to contribute additional absorption by some cells that were the main cause of the mismatch.

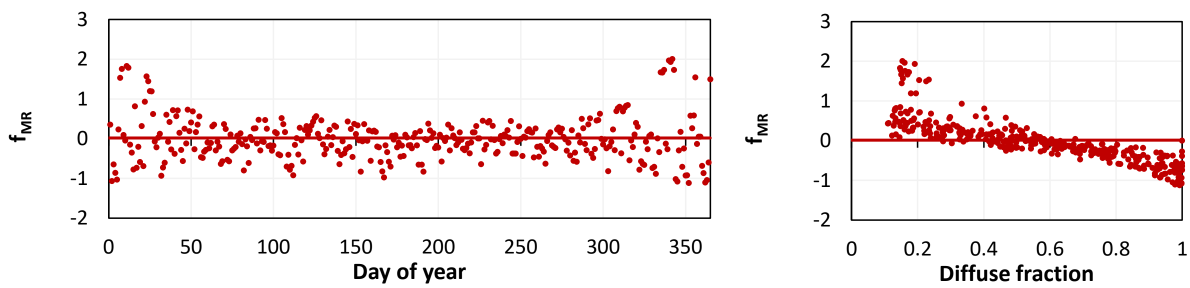


Figure 7: Energy-weighted daily values for f_{MR} determined for the single-axis tracker example, plotted against **(left)** day of year and **(right)** diffuse fraction.

8 For a 1P single-axis tracker, which has a torque tube across the middle of the module, fS is higher and increases when the sunlight becomes more diffuse. That's because on a diffuse day, more light reaches the rear of the modules from the sky (without reflecting from the ground) and this light travels obliquely to the module. Thus, on diffuse days, there is more rear irradiance but it is more non-uniform due to torque-tube shading.

9 Since the tracking algorithm includes ideal backtracking, there is never any row-to-row shading of direct light.

The factors of the fixed-tilt system differ from those of the single-axis tracker:

The albedo factor f_A is about 0.7% higher at all times of year. It might seem strange that the albedo differs between fixed and tracking systems, but remember that f_A represents an effective albedo. The real albedo is indeed the same for both systems (plotted in Figure 4). For the fixed-tilt system, however, a smaller fraction of the ground-reflected sunlight reaches the rear of the panel around midday than in the morning or afternoon. Hence, for this fixed-tilt system, the red-shifted morning and afternoon sunlight, which is reflected more by grass, is more important and contributes to a higher f_A.

The transmission factor f_T is 0% because in this fixed-tilt system, there is no spacing between modules. (Although due to the stochastic nature of the ray tracing, the calculated daily f_T is actually ±0.07%.)

The largest difference between systems is the shading factor f_S, as plotted below. It is much larger for the fixed-tilt system than the tracker, as one would expect by comparing the structural supports in Figure 3. We also find that f_S increases (rather than decreases) as the diffuse fraction increases.

Finally, the electrical mismatch f_M is higher for fixed-tilt systems, largely due to row-to-row shading of direct light when the sun is low in the sky. The contribution to electrical mismatch from rear-side irradiance f_{MR} is very low (at 1.2%) although not as low as for the tracker (~0%).

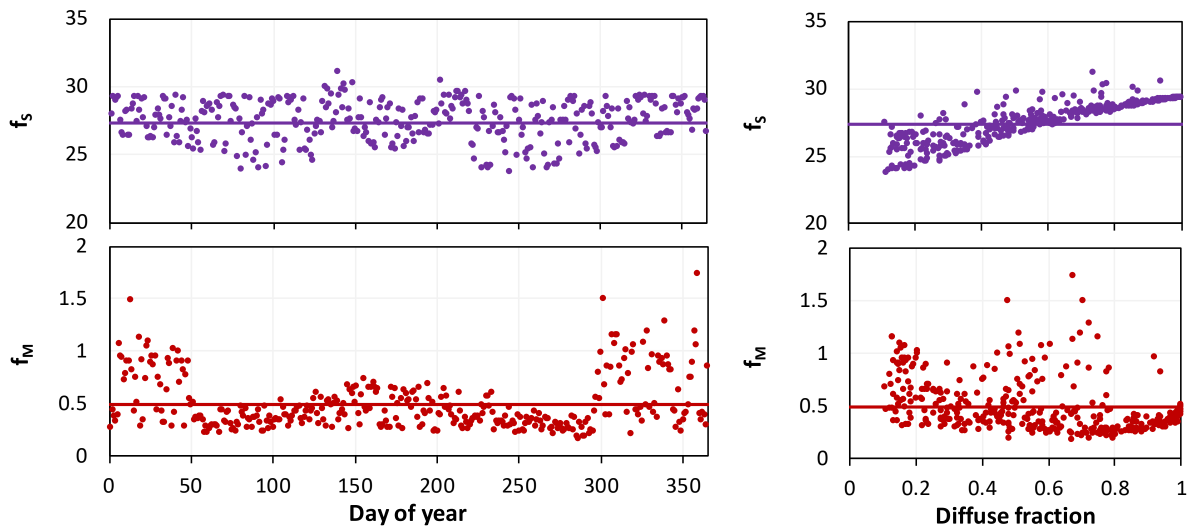


Figure 8: Energy-weighted daily values for PVSyst factors determined for the fixed-tilt example. The factors are plotted against **(left)** day of year and **(right)** diffuse fraction.

6. Derivations

The derivation of the equations in Section 3 is now provided. The purpose of these derivations is to provide a sensible methodology to determine the PVSyst factors. We note, however, that it is impossible to derive inputs that are completely consistent with the PVSyst equations because the effects of transmission, shading, albedo and mismatch are convoluted. We therefore welcome any feedback on whether our approach can be improved.

6.1 Derivation of f_T and f_S

PVSyst calculates the ideal irradiance on the rear of the module Φ_{ideal}. It assumes a constant albedo, no structural shading and no transmission through or between modules.

PVSyst then adjusts the rear irradiance, increasing it to account for transmission and reducing it due to structural shading:

$$\Phi_{adjusted} = \Phi_{ideal} \cdot (1 + f_T) \cdot (1 - f_S), \tag{7}$$

where f_T and f_S are inputs to PVSyst called the transmission and shading factors.

In the case where there is no shading ($f_S = 0$), we can rewrite (7) as

$$1 + f_T = \frac{\Phi_{adjusted}}{\Phi_{ideal}}, \tag{8}$$

and, since module current is proportional to irradiance,[10] (8) is equivalent to

$$1 + f_T = \frac{I_{R3}}{I_{R4}}, \tag{9}$$

where I_{R3} and I_{R4} are the rear current from the third and fourth simulations (see Section 3). Both simulations have the desired albedo and no shading, but I_{R3} includes transmission and therefore corresponds to the adjusted irradiance $\Phi_{adjusted}$, whereas I_{R4} excludes transmission and therefore corresponds to the ideal irradiance Φ_{ideal}.

[10] PVSyst effectively determines the short-circuit current I_{SC} by multiplying the irradiance by I_{SC_STC} / 1000 W/m^2, where I_{SC_STC} is the I_{SC} at standard-testing conditions.

Thus,

$$f_T = \frac{I_{R3} - I_{R4}}{I_{R4}}. \tag{10}$$

Hence, f_T is the relative difference in rear current between simulations with and without transmission between modules.

Likewise, we can rearrange (7) to give

$$1 - f_S = \frac{\Phi_{adjusted}}{\Phi_{ideal} \cdot (1 + f_T)}, \tag{11}$$

or

$$1 - f_S = \frac{I_{R1}}{I_{R4} \cdot (1 + f_T)}, \tag{12}$$

since I_{R1} is the solution to the simulation that accounts for the desired albedo, the transmission, and shading (adjusted for everything). Combining (12) with (10) and rearranging gives

$$f_S = \frac{I_{R3} - I_{R1}}{I_{R3}}. \tag{13}$$

Hence, f_S is the relative difference in rear current of simulations performed with and without structural shading.

6.2 Derivation of f_A

The derivation of f_A is only meaningful when the albedo implemented in SunSolve varies with wavelength. In such case, we must determine an 'effective albedo' to represent all wavelengths, as required for PVSyst.

We therefore derive an equation for the effective albedo f_A. It is defined as the albedo for which solving SunSolve with f_A gives the same rear irradiance (and hence rear current) as when solving with the wavelength-dependent albedo.

For simplicity, we assume that the rear current I_R is proportional to albedo, which is not strictly true due to secondary bounces (see Section 7.2), but which is a good approximation.

Thus, we use the results of Simulations 4, 5 and 6, which all represent the ideal case of having no structural components (f_S = 0) and no transmission (f_T = 0), but where I_{R4} is determined with the desired albedo, I_{R5} is determined with an effective albedo of f_{A5}, and I_{R6} is determined with an effective albedo of f_{A6}.

Since we assume there is a linear relationship between current and albedo, we can write

$$\frac{f_{A4} - f_{A6}}{I_{R4} - I_{R6}} = \frac{f_{A5} - f_{A6}}{I_{R5} - I_{R6}}, \tag{14}$$

where f_{A4} is the effective albedo that yields I_{R4} (i.e., f_{A4} is our desired f_A). Hence, we can rearrange (14) to give

$$f_A = f_{A6} + (I_{R4} - I_{R6})\frac{f_{A5} - f_{A6}}{I_{R5} - I_{R6}}. \tag{15}$$

The effective albedo of most sites lies between 0.1 and 0.35, and so we recommend using f_{A5} = 0.3 and f_{A6} = 0.2. (Section 7.2 discusses this in more detail). This gives the equation for f_A given in Section 3:

$$f_A = 0.2 + (I_{R4} - I_{R6}) \cdot \frac{0.3 - 0.2}{I_{R5} - I_{R6}}. \tag{16}$$

6.3 Derivation of f_M, f_{MF} and f_{MR}

The total cell-to-cell mismatch is determined from Simulation 1, which calculates the output power when mismatch is included P_1, as well as the output power when mismatch is neglected P_{noM1}. The mismatch factor f_M is then defined as

$$f_M = \frac{P_{noM1} - P_1}{P_{noM1}}, \tag{17}$$

which is the relative reduction in power due to mismatch under bifacial illumination.

We define the front mismatch factor f_{MF} in the same way using the results of Simulation 2, which blocks rear illumination:

$$f_{MF} = \frac{P_{noM2} - P_2}{P_{noM2}}. \tag{18}$$

The utility of this definition is debatable, given that in a real module there is no separate output for front and rear illumination, and given that mismatch is not linearly related to the non-uniformity of front and rear irradiance. Nevertheless, if PVSyst users wish to enter a separate value for the rear mismatch factor, we must select some procedure to breaking the total mismatch into constituents.

Our definition of the rear mismatch factor f_{MR} follows a different approach because we must ensure that it is consistent with the equations applied by PVSyst.

In PVSyst (Version 7.4.6 onwards), the power loss due to rear mismatch is calculated as

$$P_{MR} = f_{MR} \cdot f_B \cdot \eta_F \cdot \Phi_R, \tag{19}$$

where η_F is the front-side efficiency of the module under STC conditions and f_B is the bifaciality of the module.

PVSyst also calculates a power loss due to mismatch from other factors, which is applied to both front and rear irradiance and accounts for f_B:

$$P_{M\,other} = f_{M\,other} \cdot \eta_F \cdot (\Phi_F + f_B \cdot \Phi_R). \tag{20}$$

Notice that this equation depends on the total irradiance (front and rear) and accounts for f_B.

So, these losses should sum to give the total mismatch loss, which we determined with SunSolve Simulation 1 and rewrite as

$$P_{M\,total} = f_M \cdot \eta_F \cdot (\Phi_\mathrm{F} + f_B \cdot \Phi_\mathrm{R}) \tag{21}$$

to keep in the same form as the PVSyst losses.

Since the total mismatch loss is the sum of all mismatch losses,

$$P_{M\,total} = P_{MR} + P_{M\,other}. \tag{22}$$

Thus, Equations (19)–(22) can be combined to give

$$f_M \cdot \eta_F \cdot (\Phi_F + f_B \cdot \Phi_R) = f_{M\,other} \cdot \eta_F \cdot (\Phi_\mathrm{F} + f_B \cdot \Phi_R) + f_{MR} \cdot f_B \cdot \eta \cdot \Phi_\mathrm{R}, \qquad (23)$$

and hence,

$$f_{MR} = (f_M - f_{M\,other}) \cdot \left(1 + \frac{\Phi_F}{f_B \cdot \Phi_R}\right). \qquad (24)$$

Finally, we replace the other sources of mismatch with f_{MF} because this is the only other mismatch contained in the SunSolve simulation; as discussed in Section 7.5, the user can increase f_{MF} to include other sources of mismatch if desired. We also replace the ratio of the irradiance with the ratio of the current generation from Simulation 1,[11] and thus,

$$f_{MR} = (f_M - f_{MF}) \cdot \left(1 + \frac{I_{F1}}{I_{R1}}\right), \qquad (25)$$

which is the equation for f_{MR} in Section 3.

Finally, if using PVSyst versions prior to 7.4.6, which used $P_{MR} = f_{MR} \cdot \eta_F \cdot \Phi_R$ instead of (19), the equation for f_{MR} should be

$$f_{MR} = (f_M - f_{MF}) \cdot f_B \cdot \left(1 + \frac{I_{F1}}{I_{R1}}\right). \qquad (26)$$

[11] This assumes that f_B, which is the STC ratio, η_R / η_F, is equal to I_{LF} / I_{LR}. Put otherwise, it assumes that the dominant reason for differences in η_F and η_R is optical rather than electrical (recombination or series resistance).

7. Comments for advanced users

7.1 Skipping simulation steps

Under some scenarios it is possible to skip one or more simulations and arrive at the same results with PVSyst:

- If the desired albedo is constant with wavelength, skip Simulations 5 and 6 and simply set f_A to equal that albedo.
- If there is no spacing between modules, f_T must be zero and you can skip Simulation 3. Load the results from Simulation 4 to represent Simulation 3 in the analysis spreadsheet.
- If you wish to minimise steps, you can skip Simulations 3 even if there is spacing between modules. In this case, set f_T to zero and set f_{AT} from the spreadsheet to be the ground albedo in PVSyst. This combines the albedo and transmission into a single factor (which was the method described in our previous white paper on PVSyst factors).
- If you're content to combine cell-to-cell mismatch due to both front and rear illumination into a single parameter, you can skip Simulation 2. In this case, use f_M to represent both f_{MF} and f_{MR} combined. We describe how to do that in Section 0.

7.2 Albedo factor f_A

Here we explain how multiple reflections from the ground affects the determination of f_A.

Firstly, a recap on f_A: We determine f_A from the results of Simulations 4, 5 and 6. Simulation 4 contains the actual albedo, which might be the albedo of grass, sand or concrete, and which generally depends on wavelength as we saw for green grass in Figure 4. How then do we determine the 'effective albedo', i.e., the albedo factor f_A, of Simulation 4?

The procedure can be understood with Figure 9, which plots results from Example 1.[12] It shows how we take the rear irradiance I_R from Simulations 5 and 6, which had a constant albedo of 30% and 20%, assume a linear relationship, take I_R from Simulation 4 (for which the ground was green grass), and deduce its effective albedo from the linear fit. We find, in this example, that the effective albedo of Simulation 4 was f_A = 25.5%.[13]

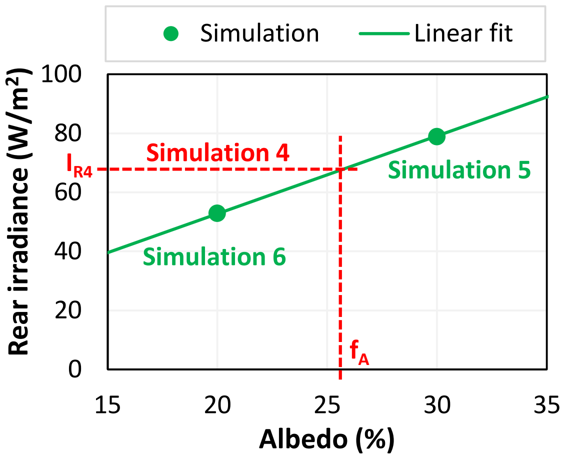


Figure 9: Rear irradiance vs albedo for Example 1.

The albedo selected for Simulations 5 and 6 needn't straddle f_A but, as we'll see, it helps if the points are close to f_A. Our suggested procedure uses 30% and 20% because those values are similar to the albedo at many sites but if, for example, you're simulating snow, you might modify the albedo of Simulations 5 and 6 to, say, 80% and 60%. If you do, be sure to adjust the inputs at cells AM3 and AS3 in the "PVSyst-input-analysis" spreadsheet to ensure that the calculations of f_A account for the albedo applied in Simulations 5 and 6.

---

[12] See Section 0.

[13] Due to the wavelength dependence of the illumination, the actual albedo, and the module response, the effective albedo changes during the day and year. We saw that effect in Figure 6. For this reason, f_A will also vary from site to site—even for the same grass.

In the previous version of our procedure, we'd used 100% and 0% for Simulations 5 and 6. This seemed sensible at the time (those values were easier to describe and gave a higher dynamic range) but they introduced a slight error. That error arose because the relationship between I_R and albedo is not, in fact, linear. Why? Because some rays reflect multiple times from the ground, like the ray shown in Figure 10. Rays that reflect twice from the ground contribute to a parabolic dependence of I_R on albedo, rays that reflect three times contribute to a cubic dependence, and so on. Thus, in reality, the dependence of I_R on albedo is dictated by the properties of the illumination, system and module. Generally, that dependence is slightly superlinear. While we could run three simulations and make a parabolic fit to the data, it is sufficiently accurate to assume a linear fit provided the two points are close to f_A.

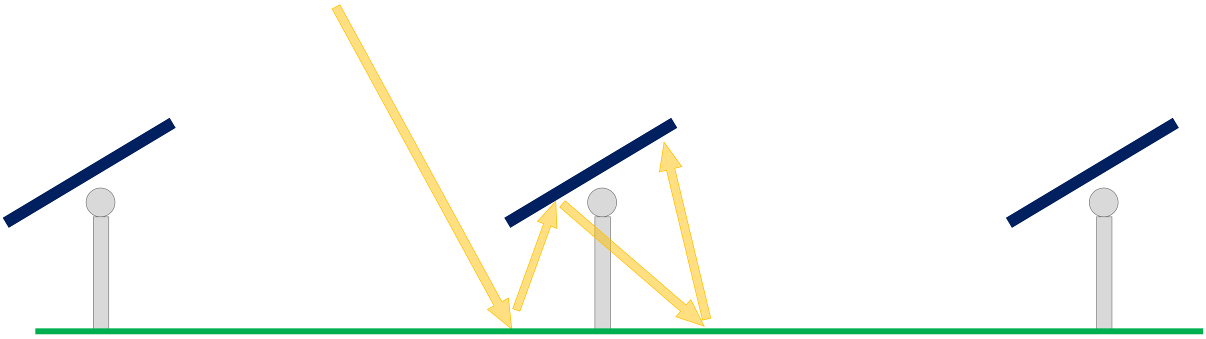


Figure 10: Example of a ray that reflects twice from the ground before being absorbed by a module.

Due to that superlinearity, the calculation of f_A from a linear fit to two datapoints is more accurate when the two datapoints are close to f_A. In this example, if we had used 0% and 100% for Simulations 5 and 6, we would have deduced that f_A was 22.70%. With 20% and 30% we found f_A = 25.51%, and with 24.5% and 25.5% we find f_A to be 25.59%. These results provide an indication of the error introduced by our previous methodology.

There's one final thing to clarify. The goal of the above procedure is to determine f_A in a manner with low (or zero) systematic error caused by multiple bounces. That means that f_A excludes the effect of multiple bounces. Yet while the ray tracing (and reality) includes multiple bounces, PVSyst does not. That's one of the many subtle reasons that the final yield from SunSolve will not be identical to the final yield from PVSyst, even if you insert values for f_A, f_T, f_S and f_{MR} from this procedure.[14]

[14] To quantify the effect of multiple bounces, one could simulate the case when rays are prevented from reflecting more than once from the ground. This is not possible in the current version of SunSolve.

7.3 Shading factor f_S

The determination of f_S appears simple: We ray trace the scene with structural supports (Simulation 1) and without structural supports (Simulation 3) and then calculate the relative reduction in rear irradiance I_R.

This procedure does, however, raise some interesting questions: (i) When solving without structural supports, why do we make the supports transparent rather than excluding them altogether? (ii) How can we quantify the contribution from each structural element? (iii) How do module frames contribute to structural shading? We now address each question in turn.

The use of transparent structures in Simulations 3−6

With Simulation 3 (and subsequent simulations), we want the scene to behave exactly like the original scene, Simulation 1, except that the rays are unaffected by the existence of structures. We can achieve that by making the structural supports transparent.

Why do we make the elements transparent instead of excluding them? There are some structural elements that modify the scene when they are included or excluded. For example, if we omit the torque-tube, the modules no longer rotate around the central axis of the torque tube but around the bottom of the module. Thus, except when they are horizontal, the modules in Simulations 1 and 3 would be at a different height if the torque tube were excluded rather than made transparent.

In Example 1, omitting the structure instead of making it transparent leads to a calculated f_S of 8.4% instead of 8.1%. Granted, it's not a large difference, but there's no need to introduce systematic error if we don't have to.[15]

[15] In fact, in previous versions of this document, we did exclude the structural supports rather than making them transparent, incurring a small systematic error such as this.

The shading factor of individual structural elements

With SunSolve, it's possible to determine the shading factor of each element. We do that by making all elements transparent except one. The shading factors for Example 1 are plotted in Figure 11. (Here, we have included the frame in f_S, which we'll explain below).

We find that, for the dimensions of Example 1, the elements that shade the rear irradiance the most are the torque tube (5.6%) and the rails (4.9%), followed by the frame (2.8%) and the posts (1.0%).

Notice that if we sum of the f_S of the individual elements (14.3%), the result is greater than the f_S when we include all components (10.7%). That's because shading is convoluted; i.e., shading from one structure can fall upon another structure. In fact, light can also reflect from one element onto another—or onto the cells. Thus, the individual contributions from each f_S cannot be uniquely determined but, nevertheless, this approach does provide a means to gauge the impact of each element on rear irradiance.

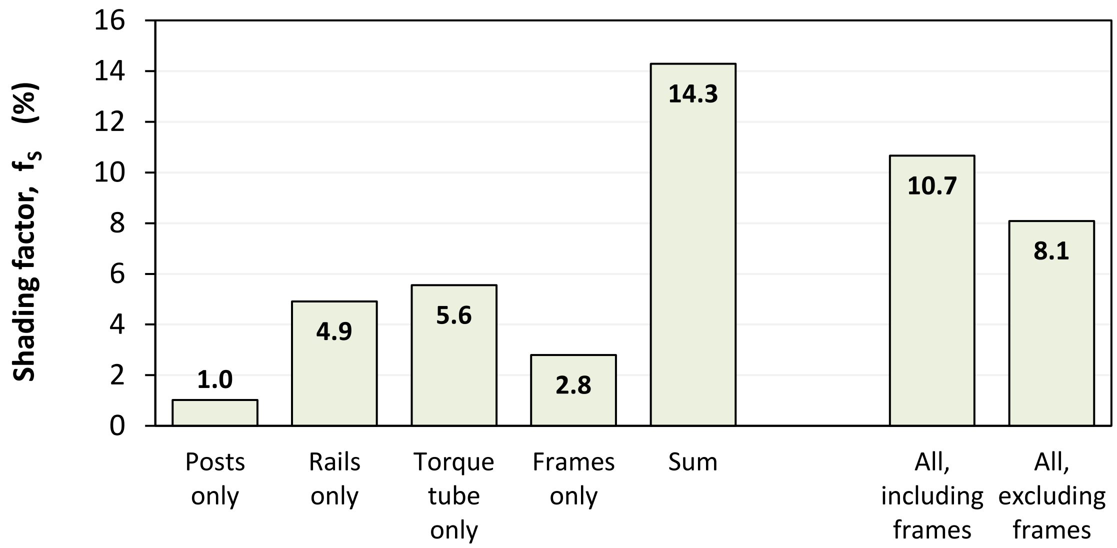


Figure 11: Shading factor calculated for the inclusion of various structural elements in Example 1.

The contribution of module frames to f_S

And now to answer the trickiest of the three questions: Should we account for the frames when determining the shading fraction for PVSyst?

First, consider that the view-factor model within PVSyst excludes frames, so ideally, we want to modify the view-factor calculations to account for shading from the frames. How is that achieved?

First note that some frame shading might already be included in f_B (taken from the PAN file). Normally, a bifacial factor f_B is determined by taking the P_{MP} of a rear STC measurement and dividing it by the by P_{MP} of the front STC measurement. Especially when the flanges (or brackets) of a module frame are large and the cells are close to the edge, the rear STC measurement will incur some shading from the frame, and hence f_B will include some frame shading.[16]

When a module operates within a system, however, we expect additional shading from the frame. That's because light incident to the rear arrives from many angles. Thus, the deeper the frame, or the wider its flanges, the more it shades the cells.

In short, an accurate PVSyst simulation requires us to include in f_S the difference between frame shading in the field and under STC conditions. That's not simple.

In our collaborations with Array Technologies and FTC Solar, and in our previous publications, we have excluded the contribution of the frames in f_S. That's a useful approach because (i) it's simpler, and (ii) information about the frames is often not available.

Nevertheless, we can include the contribution of the frames in f_S by utilising SunSolve-Power as well as SunSolve-Yield. With SunSolve-Power, we create virtual modules that represent the desired module and we use that framed virtual module in Simulations 1 and 2. We then duplicate the module in SunSolve-Power, remove the frame, extend the 'white space' to

[16] Theoretically, an STC measurement is performed under normally incident light but, in practise, the illumination sources tend to be somewhat divergent. This might decrease the contribution of frame shading in f_B (and reduce the contribution of electrical mismatch within f_B).

ensure the module dimensions are unchanged, and use that frameless virtual module in Simulations 3−6.[17] In so doing, the computation of f_S includes shading from the frames.

In the results above for Example 1 (Figure 11), we found that when we neglect the frames, the shading factor was f_S = 8.1%, whereas when account for the frames, f_S = 10.7%. It's evident, therefore, that the frames of the LR5 module have a small but significant impact on shading.[18]

This procedure ensures that all frame shading is included within f_S, but if we want to be more accurate, we must also reduce f_S to account for any contribution of frame shading from f_B, or increase f_B to remove the effect of frame shading. It's simpler to do the latter by determining f_B using SunSolve-Power, and even account for the divergent beam in the STC measurement. (If not, f_S simply provides an upper limit to the contribution of frame shading.)

As you can see, modifying PVSyst input factors to account for frame shading is a difficult and lengthy process. In many cases, the increased accuracy attained by accounting for frames might not be worth it. For now, at least, we do not recommend accounting for frame shading, but contact us if you would like help in this regard.

[17] When using the frameless module in SunSolve-Yield, we also increase the distance from the bottom of the module from the top of the torque tube by the depth of the missing frame. Thus, the distance from the rear glass to the torque tube is the same in Simulations 1 & 2 (with frame) and Simulations 3−6 (without frame).

[18] The shading factor for the frames by themselves was f_S = 2.8%, which is a little more than the difference between the two cases (2.6%). That's because some shade from the posts, torque tube and rails all falls upon the frames, and so, the frames must contribute less than 2.8% to the total shading.

7.4 Transmission factor f_T

PVSyst's view-factor model for the rear irradiance I_R does not include gaps between modules, and so it accounts for them by multiplying the I_R determined with no gap by $(1 + f_T)$.

As illustrated in Figure 12, we can determine a meaningful value of f_T by ray tracing the system (a) with and (b) without gaps between modules and bays.[19] We can then calculate f_T with the equation $f_T = I_{R3}/I_{R4} – 1$. Notice that f_T quantifies the additional rear irradiance that arises from increasing the size of the unit system, not just from the extra light passing between modules.[20]

The reason we don't remove the vertical spacing but block it instead (i.e., no rays pass through the vertical spacing) is that we do not want the modules to move closer to the axis of rotation. It they did move, it might alter the tracking algorithm or the row-to-row shading, or the modules might be further from the ground and receive more bifacial light.

In Simulation 4 there is no lateral spacing or structural supports, so the effective size of the unit system is really just one module wide, as illustrated in Figure 12(c). We point this out because the unit system is now almost identical to the one used in PVSyst to determine the rear-side irradiance, as shown in Figure 12(d).

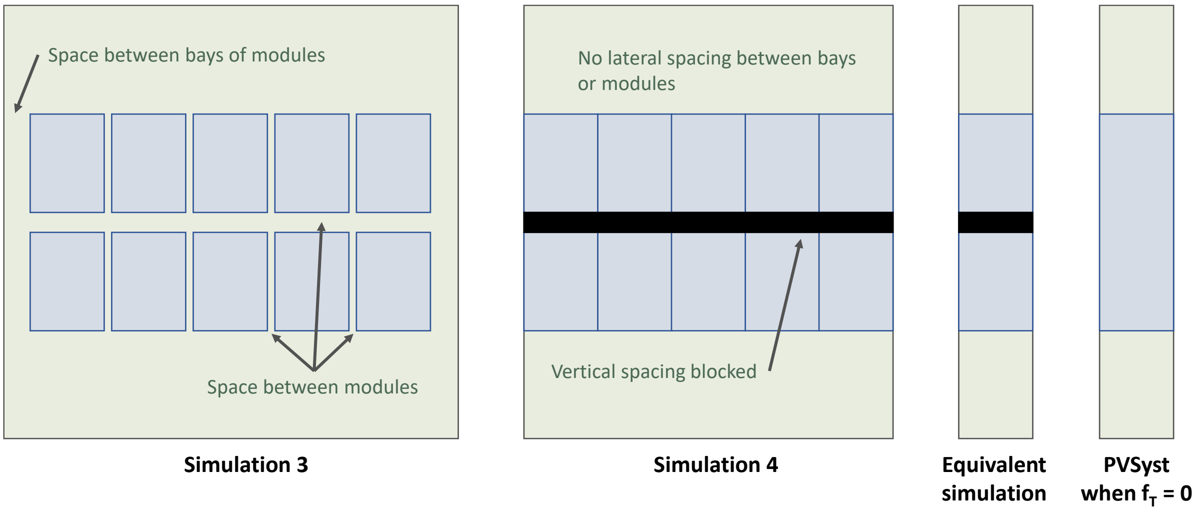


Figure 12: Plan view of the unit system used to simulate Example 1 (2P tracker): **(a)** Simulation 3 has space between bays and modules, **(b)** Simulation 4 has lateral space removed and vertical space blocked out, **(c)** equivalent to Simulation 4 (i.e., will give identical results for I_R), and **(d)** equivalent unit system for the bifacial view-factor model in PVSyst.

---

[19] Both Simulations 2 and 3 should have the desired albedo and no structural shading.

[20] Transmission through modules is trickier to quantify but this is not an issue in modern bifacial modules.

We note that our procedure to determine f_T assumes that the module itself is not transparent. That is, the procedure assumes that no light passes between or through the cells of the module. If the module were semitransparent, our procedure should be modified as follows: First, create a secondary module in SunSolve-Power that mimics the actual module except that it has no transparency. For example, if the actual module is a glass–glass design with transparent EVA and no backsheet, then the secondary module should be identical except for containing a 100% absorbing layer between the cells. Thus, the secondary module prevents any light from passing between the cells. Then, this secondary module should then be used in Simulations 4, 5 and 6 rather than the actual module. The resulting value for f_T will then include transmission both between and through the modules of the system.

7.5 Mismatch factors, f_M, f_{MF} and f_{MR}

Summary

- The cell-to-cell electrical mismatch due to non-uniform irradiance f_M is determined by Simulation 1.
- The contributions from non-uniform irradiance on the front f_{MF} and rear f_{MR} of the module are distinguished by combining Simulations 1 and 2.
- There are multiple inputs in PVSyst that account for electrical mismatch.
  - It is clear where f_{MR} should be inserted in PVSyst.
  - It is not clear where f_{MF} should be inserted because that depends on how the PVSyst simulation is set up. With certain settings, PVSyst calculates some of f_{MF} at each hour of the year, and with other settings it does not. We explain those settings and leave it to the user's discretion how best to incorporate f_{MF}.
- There are additional sources of electrical mismatch not calculated by SunSolve.

Background

The term "electrical mismatch loss" refers to power loss due to non-uniform behaviour of the cells within a module, or the modules within a string.[21]

For example, if all of the cells in a 72-cell module receive the same irradiance except for one cell that receives 50% of that irradiance (e.g., it's shaded by a bird dropping), then the module power is reduced by

- 0.7% due to optical shading (0.5 / 72),

and an additional

- 34.3% due to electrical mismatch (computed by solving the module circuit[22]);

this mismatch loss arises because any cell connected in series to the shaded cell is limited to the same current, preventing it from operating at its ideal maximum-power point.

[21] Electrical mismatch loss is sometimes referred to as "electrical shading" or sometimes just "mismatch loss", and PVSyst refers to it as the "electrical effect".

[22] The exact reduction in module power will depend on how the cells and bypass diodes are strung together.

In this section, we'll first list the many sources of electrical mismatch, then describe how they are incorporated into PVSyst and SunSolve, then remark on what inputs to use PVSyst, and finally, describe how our new approach differs from the old one. Be warned: this explanation contains many details and the conclusions are not particularly satisfactory. There remains a lot to be learned about mismatch in PV systems.

Sources of electrical mismatch

There are many sources of non-uniformity that cause electrical mismatch within a module or a string of modules:

1. Cells or modules receive different illumination due to
    a. row-to-row shading of direct light,
    b. row-to-row shading of diffuse light,
    c. edge effects (e.g., less shading at the end of a row of modules),
    d. clouds,
    e. non-uniform soiling (e.g., bird droppings, more dust at bottom of modules),
    f. near-field shading, like from trees, buildings, telegraph poles,
    g. rear-side non-uniformity arising from[23]
        i. variable irradiance falling on the ground due to gaps between modules,
        ii. shading and reflection from mounting structures (e.g., posts, purlins, rafters, torque tube, module frames),
        iii. spatially non-uniform albedo (e.g., dirt between rows, grass under modules);
2. Cells or modules operate at different temperatures due to
    a. non-uniform irradiance arising from Cause 1,
    b. non-uniform convection loss (e.g., some cells or modules are more exposed to the wind),

[23] In fact, these effects can also affect the front side, but they are usually considered a rear-side effect.

c. non-uniform radiative loss or gain (e.g., some cells 'see' more of the sky and therefore radiate more heat to the sky),

d. greater conduction of heat from cells nearer the module frames and towards the ground;

3. Cells or modules are of a variable quality due to

   a. natural production variation from a cell or module manufacturer,

   b. using different module types within the same string,

   c. variable degradation rates due to their exposure on site (e.g., hot-spot heating, potential-induced degradation, light-induced degradation, differing irradiance and temperatures).

Electrical mismatch in PVSyst — General

PVSyst handles electrical mismatch in three ways:

- Row-to-row shading of direct light (Source 1a) is included as the "electrical effect" in either the "Orientation" section or the "Near shadings" section.
- Rear-side non-uniformity (Source 1g) is included as a bifacial mismatch factor.
- All other sources can be accounted for with two additional mismatch factors in the "Quality – LID – Mismatch" section.

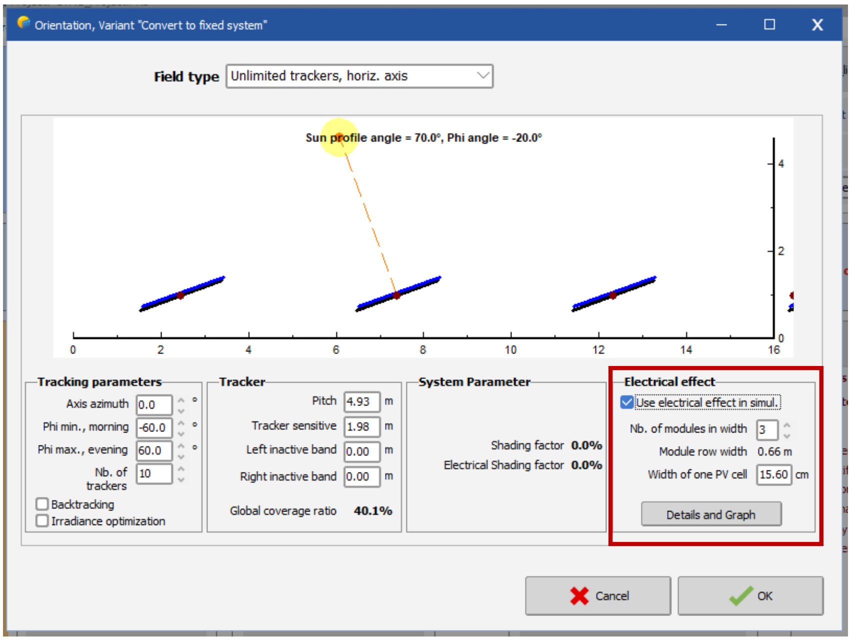
We'll describe each in turn.

Electrical mismatch in PVSyst — row-to-row shading of direct light

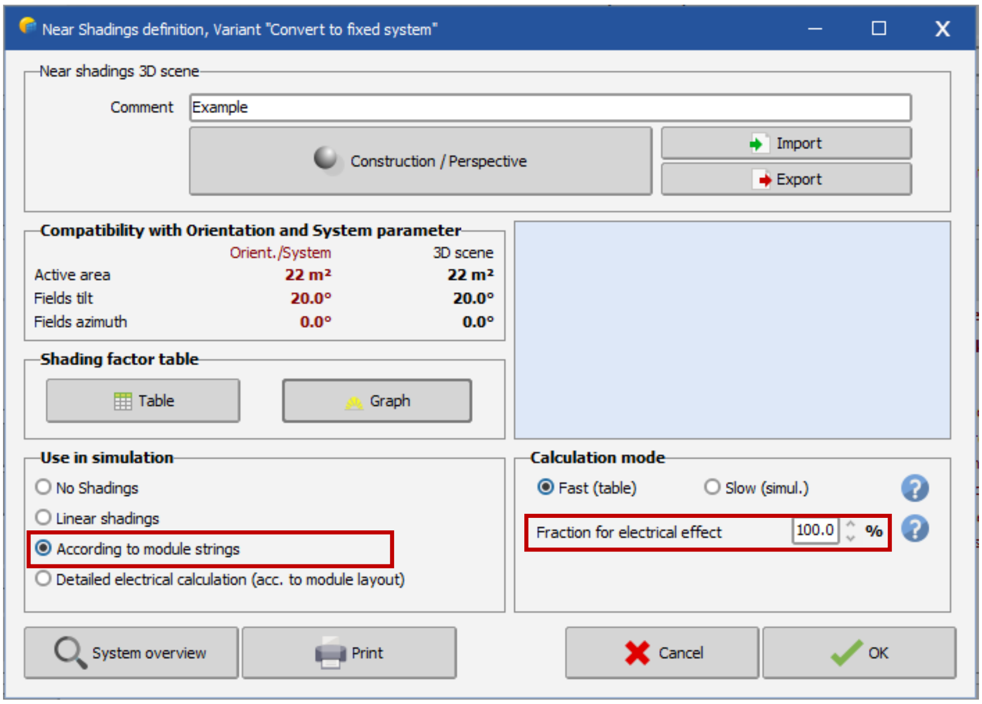
PVSyst can be set up to specifically account for row-to-row shading of direct light, and the subsequent electrical mismatch. PVSyst refer to this as the "electrical effect".

When solving the cases of "unlimited sheds" or "unlimited trackers" without backtracking,[24] you can switch on electrical mismatch due to beam shading by checking "use electrical effect".

[24] If one applies backtracking for an unlimited tracker, PVSyst ignores the "electrical effect" because it only concerns the direct light (neglecting diffuse and ground-reflected light)



Alternatively, when simulating limited-sized systems (i.e., a 3D scene), the user can account for the "electrical effect" by setting the near shading definition to "According to module strings" and assigning a non-zero percentage to the "Fraction for electrical effect".[25]



[25] If one selects "linear shading", it's like having no electrical effect. If one selects "detailed electrical calculation", the electrical effect is also applied but this is only available for small systems and is unlikely to be applicable so we don't discuss it further.

PVSyst calculates the power loss due to row-to-row shading at every time interval as follows: It first determines the module power as if there were no shading or mismatch, P_{mod_ideal}, and then multiplies it by a fraction,[26]

$$P_{mod} = P_{mod\ ideal} \cdot (1 - f_{BS}), \tag{27}$$

where f_{BS} accounts for the losses due to beam shading; i.e., both the shading loss and the resulting mismatch loss. The value of f_{BS} depends on the method used to determine the "electrical effect".

If there is shading but no electrical effect, f_{BS} is the fraction of the modules that is shaded from direct light. PVSyst calls this approach "linear shading" because the power loss depends linearly on shading, which is how a module would behave if there were no mismatch; i.e., if all of its cells operated at their own individual maximum-power point.

If there is shading and an electrical effect, PVSyst determines the shaded fraction of each cell in the module, and sets f_{BS} to equal the highest of all fractions.[27] Then, in the case of a 3D scene (but not unlimited trackers), that shading fraction is multiplied by the "fraction for electrical effect" f_{EEF}.[28] It's not simple to follow!

For example, if a module has at least one fully shaded cell, then

- for unlimited trackers, f_{BS} would equal 1, and the module would produce no power;
- for a 3D scene, f_{BS} would equal 1 multiplied by f_{EEF}, and if f_{EEF} were 0.5, the module power would be $P_{mod} = P_{mod\ ideal} \times 0.5$.

Now, excluding the "electrical effect" (i.e., assuming linear shading) will lead to an overestimate in P_{mod} because mismatch is neglected, whereas including the "electrical effect" with f_{EEF} = 1 will lead to an underestimate in P_{mod} because PVSyst's procedure neglects the contribution from (i) diffuse or rear irradiance, which mitigate beam shading, and (ii) the

[26] You can rearrange this to find the power loss due to beam shading: $\Delta P = P_{mod\_ideal} - P_{mod} = f_{BS} \cdot P_{mod\_ideal}$.

[27] This value depends on the number of cells in the module, their size, and the module layout. E.g., if it's a 2-high system, "Nb of modules in width" should be set to 2.

[28] What makes this section difficult to describe is that both the optical shading and the electrical mismatch are combined in the term $(1 - f_{BS})$.

contribution from bypass diodes and non-shaded strings, which mitigate the effect of non-uniform current in the strings of a module.[29] This is why PVSyst include f_{EEF}; it helps the user modify the mismatch so that its impact is less digital.

Clearly, it is very difficult for the PVSyst user to know the best approach to take to account for an accurate evaluation of row-to-row shading of direct light.

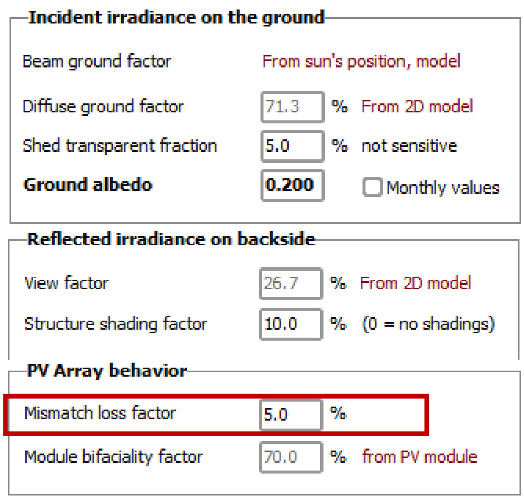
Electrical mismatch in PVSyst — bifacial mismatch factor

PVSyst provides an input factor to account for mismatch arising from non-uniform rear illumination. We call that factor f_{MR}. This is the bifacial factor that many users would like as an input for simulating bifacial modules.

The following equation shows how PVSyst modifies the module power with f_{MR}:

$$P_{mod} = \eta \cdot [\Phi_{\mathrm{F}} + f_B \cdot (1 - f_{MR}) \cdot \Phi_{\mathrm{R}}], \tag{28}$$

where η is the front-side efficiency of the module under STC conditions, Φ_F and Φ_R are the irradiance incident to the rear, and f_B is the bifacial factor. In this equation we assume that Φ_R already accounts for rear shading, transmittance and albedo.



[29] In fact, even if the module had no bypass diodes and all strings were shaded identically, the approach taken by PVSyst would still overestimate the power loss (unless the user set f_{EEF} to some value less than 1**)**.

Electrical mismatch in PVSyst — other sources of mismatch

There are two more places where we can insert mismatch into PVSyst. Both are found in "Detailed losses".

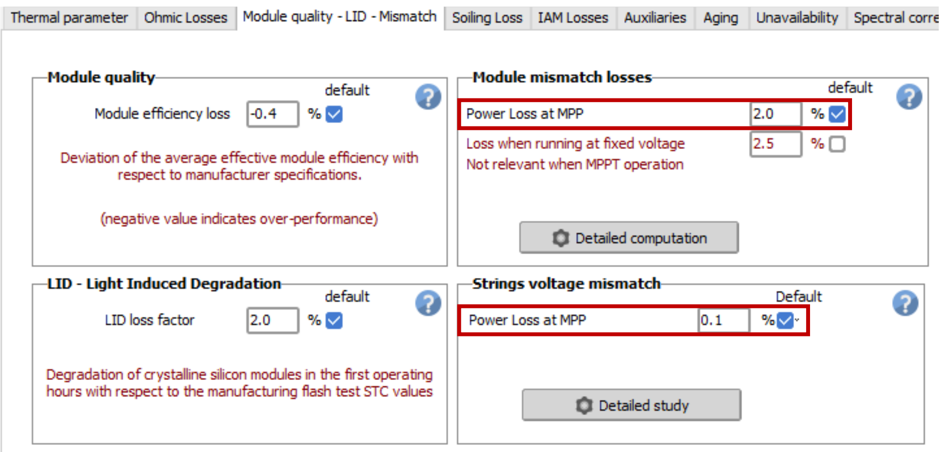
The first is labelled "Power Loss at MPP" in the "Module mismatch losses" section. The PVSyst manual states that this factor accounts for non-uniformity in the performance of the cells and modules.

The second is labelled "Power loss at MPP" in the "Strings voltage mismatch" section. The manual states that this accounts for string-to-string mismatch.

In effect, both factors are applied in the same manner, and we simply refer to them as f_{M1} and f_{M2}. PVSyst incorporates them into the behaviour of the DC string power by

$$P_{DC} = N_{mod} \cdot P_{mod} \cdot (1 - f_{M1}) \cdot (1 - f_{M2}). \tag{29}$$

Thus, they are just linear loss factors. Their default values are 2% and 0.1%.



Electrical mismatch in PVSyst — all combined

If we combine all of these loss factors into a single equation, we get

$$P_{DC} = N_{mod} \cdot \eta \cdot [\Phi_{\mathrm{F}} + f_B \cdot (1 - f_{MR}) \cdot \Phi_{\mathrm{R}}] \cdot (1 - f_{BS}) \cdot (1 - f_{M1}) \cdot (1 - f_{M2}). \tag{30}$$

This highlights a few complications with PVSyst's approach: (i) mismatch due to front and rear irradiance must somehow be deconvolved; (ii) the front mismatch is applied to both the front and rear irradiance; and (iii) the mismatch factors compound. We should also restate another complication: (iv) the term f_{BS} contains both beam shading and the associated mismatch (unless linear shading is selected);

We next attempt to resolve some of these complications. Before we do, we quickly describe how SunSolve accounts for mismatch.

Electrical mismatch in SunSolve

A SunSolve simulation incorporates cell-to-cell mismatch due to reasons 1a−1c and 1g from first principles. That is, it uses ray tracing to determine the irradiance received by each cell (front and rear), then it computes the current generation in each cell, and then it solves the complete electrical circuit of the module, including the cell layout and bypass diodes.

SunSolve therefore avoids the aforementioned complications with PVSyst. Since SunSolve can be used to solve with and without the effects of mismatch, it can determine the mismatch loss f_M due to Sources 1a−1c and 1g within a module. We can also use it to quantify contributions from the rear f_{MR} and front f_{MF} irradiance.

SunSolve does not, however, account for any other source of electrical mismatch.[30] If a user wants to account for them, they need to derate the output power accordingly.

How to set the inputs for PVSyst

As we've seen, there are four mismatch factors used by PVSyst: f_{BS}, f_{MR}, f_{M1} and f_{M2} (remembering that f_{BS} accounts for optical shading as well as electrical mismatch). Together, these factors should account for all sources of electrical mismatch.

[30] At the time of writing, we are referring to SunSolve version 6.11.12. Future version may incorporate more sources of mismatch. Also, the mismatch calculated by SunSolve includes stochastic error if insufficient rays or solar angles are solved.

We can provide instruction determining just one of these inputs: f_{MR}. By the method described in Section 4, one can use SunSolve to determine f_{MR} due to cell-to-cell mismatch (but not module-to-module mismatch) resulting from Sources 1a, 1b, 1c and 1g.

That's useful, but we cannot use SunSolve to determine the other mismatch factors. All we can do is determine f_{MF} due to the same sources and suggest that PVSyst users ensure that their inputs for f_{M1} and f_{M2} sum to a value that is greater than f_{MF} (when the user is not including the "electrical effect").

We now do our best to summarise the PVSyst inputs for four main scenarios:

(i) Unlimited trackers with backtracking, when f_{BS} is necessarily zero at all times (because there is no row-to-row shading of direct light);

(ii) Other scenarios when "electrical effect" is switched off;

(iii) Unlimited sheds and trackers without backtracking, when "electrical effect" is switched on;

(iv) 3D scenes when "electrical effect" is switched on;

where the table below comments on how mismatch can be considered for each scenario.

Table III: Comments on how electrical mismatch is incorporated into PVSyst.
(Source 1a is due to beam shading; Source 1g is due to non-uniform rear irradiance.)

Source[a]	**Input in PVSyst**	**Sym**[b]	**Scenario (i)**	**Scenario (ii)**	**Scenario (iii)**	**Scenario (iv)**
1a	"Electrical effect"	f_{BS}	Irrelevant	Accounts for shading but neglects mismatch at each hour of day	Calculated by PVSyst at each hour of day	Calculated by PVSyst at each hour of day and modified by f_{EEF}.
1g	"Mismatch loss factor" in bifacial inputs section	f_{MR}	f_{MR} can be determined by the procedure in Section 4.			
All others	"Module mismatch" & "string voltage mismatch"	f_{M1} & f_{M2}	$f_{M1} + f_{M2}$ should exceed f_{MF} determined in Section 4, which quantifies the contribution from 1a, 1b and 1c.		$f_{M1} + f_{M2}$ should consider the contribution of f_{MF} determined in Section 4, which quantifies Sources 1a, 1b, 1c. NB: it should exceed the contribution from 1b + 1c, but they are not individually calculated by SunSolve.	

[a] Source of mismatch; [b] Symbol used in this report, not in PVSyst. NB: f_{BS} includes optical shading as well as mismatch.

How this new approach differs from the old approach of Version 4

This new approach is (i) determine f_M, f_{MF} and f_{MR} with SunSolve, (ii) insert f_{MR} as the rear mismatch factor,[31] and (iii) ensure f_{MF} is contained in $f_{M1} + f_{M2}$. How does that differ from the old approach described in previous version of this document?[32]

The old approach was (i) determine f_M with SunSolve,[33] (ii) modify f_M by

$$f_M' = f_M \cdot \frac{\Phi_F + f_B \cdot \Phi_R}{\Phi_R}, \qquad (31)$$

and (iii) insert f_M' as the rear mismatch factor. In that way, mismatch due to front and rear irradiance were combined into a single input that represented rear-side mismatch. We then encouraged users to remove the contribution from front-side mismatch from any of the other PVSyst inputs. This had the advantage of requiring one less simulation and of never leading to negative values.

The old approach had one major disadvantage, however, particularly for fixed systems, for which (i) mismatch is usually much greater due to beam shading of the front side, and (ii) rear irradiance can be very low. It's clear from the above equation that when f_M is large and $\Phi_F \gg \Phi_R$, f_M' will be very large. In fact, the old approach was calculating values for the bifacial input of, say, 1% for single-axis trackers and 70% for fixed systems. This gave the impression that fixed systems had an enormous problem with rear-side non-uniformity, when, in fact, there might be no rear-side non-uniformity at all!

As you'll conclude from this lengthy subsection, there is no straightforward way to determine the four mismatch inputs for PVSyst. This quagmire of detail may or may not have been helpful, but in either case, you're welcome to contact us at support@pvlighthouse.com.au to learn more about how SunSolve can be used to improve the accuracy of system simulation. Hopefully we can also recommend a more exact approach to determine the mismatch factors for your particular system.

---

[31] The PVSyst input called "mismatch loss factor" on the bifacial system definition tab.
[32] Version 4, 25-May-2021.
[33] A reminder that f_M is the electrical mismatch arising from non-uniform irradiance on the front and rear sides.